Learning with Diverse Forms of Imperfect and Indirect Supervision

Benedikt Boecking

February 24, 2023

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania

Thesis Committee:
Artur Dubrawski, *Chair*
Jeff Schneider
Barnabás Póczos
Hoifung Poon

Submitted in fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics.

CMU-RI-TR-23-03

Abstract

Powerful Machine Learning (ML) models trained on large, annotated datasets have driven impressive advances in fields including natural language processing and computer vision. In turn, such developments have led to impactful applications of ML in areas such as healthcare, e-commerce, and predictive maintenance. However, obtaining annotated datasets at the scale required for training high capacity ML models is frequently a bottleneck for promising applications of ML. In this thesis, I study alternative pathways for acquiring domain knowledge and develop methodologies to enable learning from weak supervision, i.e., imperfect and indirect forms of supervision. I cover three forms of weak supervision: pairwise linkage feedback, programmatic weak supervision, and paired multi-modal data. These forms of information are often easy to obtain at scale, and the methods I develop reduce–and in some cases eliminate–the need for pointillistic ground truth annotations.

I begin by studying the utility of pairwise supervision. I introduce a new constrained clustering method which uses small amounts of pairwise constraints to simultaneously learn a kernel and cluster data. The method outperforms related approaches on a large and diverse group of publicly available datasets. Next, I introduce imperfect pairwise supervision to programmatic weak supervision label models. I show empirically that just one source of weak pairwise feedback can lead to significantly improved downstream performance.

I then further the study of programmatic data labeling methods by introducing approaches that model the distribution of inputs in concert with weak labels. I first introduce a framework for joint learning of a label and end model on the basis of observed weak labels, showing improvements over prior work in terms of end model performance on downstream test sets. Next, I introduce a method that fuses generative adversarial networks and programmatic weak supervision label models to the benefit of both, measured by label model performance and data generation quality.

In the last part of this thesis, I tackle a central challenge in programmatic weak supervision: the need for experts to provide labeling rules. First, I introduce an interactive learning framework that aids users in discovering weak supervision sources to capture subject matter experts' knowledge of the application domain in an efficient fashion. I then study the opportunity of dispensing with labeling functions altogether by learning from unstructured natural language descriptions directly. In particular, I study how biomedical text paired with images can be exploited for self-supervised vision–language processing, yielding data-efficient representations and enabling zero-shot classification, without requiring experts to define rules on the text or images.

Together, these works provide novel methodologies and frameworks to encode and use expert domain knowledge more efficiently in ML models, reducing the bottleneck created by the need for manual ground truth annotations.

Acknowledgements

I am very lucky to have been given the opportunity to study at the Robotics Institute at Carnegie Mellon University (CMU), and I have received a great deal of support along the way. I have many people to thank, and I will likely miss many who deserve to be mentioned. My sincere apologies to them. The order in which you will see names appear does not stand in relation to how much the support of each person means to me.

I would like to thank my advisor Artur Dubrawski for his unwavering support before and throughout my PhD. Artur encouraged and enabled me to carve my own path, and for that I am forever grateful. I want to thank my fantastic collaborators Willie Neiswanger, Mononito Goswami, Salva Rühling Cachay, Frederic Sala, Vincent Jeanselme, and Nicholas Roberts. Thank you for being such a pleasure to work with. I would like to thank Kyle Miller and Predrag Punosevac for their help throughout my journey at CMU, and for always making time to provide valuable feedback. Thank you Jeff Schneider and Margeret Hall for giving me the opportunity to visit CMU as a research scholar. I want to thank my wonderful office mates and friends Nick Gisolfi and Sibi Venkatesan for all their support, the laughs, and the many cups of coffee we enjoyed together. Also, thank you Nick for sharing your LaTeXtemplate with me, and thank you to the original creator Manfred Paulini. I also want to thank the humble Alex Ratner for his groundbreaking data programming work and for championing weak supervision.

I want to thank Ozan Oktay for his honest feedback, for supporting me, for challenging me, and for teaching me. I want to thank Hoifung Poon and Tristan Naumann, for being such helpful and kind mentors, and for giving me the opportunity to intern at Microsoft. And I want to thank the rest of the teams at the Biomedical Imaging and Biomedical NLP groups for two wonderful internship experiences and amazing teamwork: Naoto Usuyama, Shruthi Bannur, Daniel C Castro, Stephanie Hyland, Javier Alvarez-Valle, and many more.

Thank you to all of my friends from my time in Pittsburgh, for making it so memorable and enjoyable: Rob Rathke, Zhe Zhang, Gabriel Porro, Valeria Antúnez, Yasmine Kotturi, Elissa Lynch, Dan Delanis, Lauren Delanis, Sina Fazelpour, Jelena Golubović, Micol Marchetti-Bowick, Tim Hyde, Matt Barnes.

Finally, and most importantly, I want to thank my family. Thank you to my parents and my sister, who have shown me unconditional love and support. To my niece Solvejg, who brings me so much joy. And to the love of my life, my wife María De Arteaga, for helping me grow, for loving me, for making me so unbelievably happy.

Funding

The primary funding sources for the work presented in this thesis are an Uber Presidential Fellowship, a Space Technology Research Institutes grant from NASA's Space Technology Research Grants Program, and Defense Advanced Research Projects Agency's awards FA8750-17-2-0130 and HR0011-21-9-0075.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Most recent breakthroughs in ML have been achieved by data-hungry deep learning models standing on the shoulders of giant datasets. In computer vision, large-scale manually annotated datasets such as ImageNET [76] have been core to the development of novel deep learning approaches [151, 241, 117], and deep learning models pretrained on large labeled datasets have driven advances in many related vision problems [138]. In healthcare, recent studies provide convincing evidence that large datasets combined with modern deep learning methods can enable advanced decision support and personalized medicine [149, 210]. For example, deep learning methods fit to hundreds of thousands of annotated training examples have enabled at or above human expert level performance in detecting arrhythmia on Electrocardiogram (ECG) data [115] and in skin cancer image classification [87].

But, a reliance on large amounts of annotated data can hamper the continued proliferation and advancement of ML in new domains and applications. The scalability of the data labeling process, as well as the attainable quality and relevance of the collected labels, have become key limiting factors for many applications of ML. Labels for the concepts of interest do not arise naturally in most applications, and the common process of collecting labels by having annotators manually inspect and annotate individual samples is expensive and time consuming. Furthermore, manual labeling tasks can be quite cumbersome, e.g. labeling anatomically-resolved abnormalities in Computed Tomography (CT) scans requires either performing pixel-level annotations, or sorting abnormalities into a hierarchy of anatomical regions [230, 88]. Moreover, the cost of manual annotation can be exacerbated by factors such as required expert knowledge, data privacy, and constantly shifting problem settings. These issues also plague techniques such as crowdsourcing that aim to scale up the labeling process through low-cost labor[1].

The temporal and monetary cost to labeling data spans across data types and

[1] I strongly believe that ethical issues of crowdsourcing (e.g. highlighted in [105]), in particular the lack of labor laws, need to be mentioned when crowdsourcing is discussed as solution to the labeled data bottleneck.

application domains, impacting organizations from the largest tech companies to healthcare and government institutions. The popular COCO dataset[168] for large-scale object detection, segmentation, and captioning of images took more than 20,000 annotator hours and was annotated over a period of two years. Using primitive tools by current standards, it took a period of 5 years to create an ECG database labeled for abnormalities of cardiac rhythm [191]. And even with modern tools, it took 4 doctors, almost 3 months to annotate $15,000$ short ECG records using the LabelECG tool [80]. Such costs substantially impede the broader adoption of beneficial applications of ML in practice.

The data annotation bottleneck has motivated multiple research directions that aim to improve how we obtain labels to train ML models, or to reduce the amount of labels needed. *Crowdsourcing* systems aim to scale data labeling by distributing the workload to (networked) workers. *Semi-supervised learning* methods make use of unlabeled data by propagating information from labeled samples to unlabeled samples to reduce the need for large labeled datasets. *Transfer learning* applies encoded knowledge from one task to another, aiming to reduce the amount of labels needed in the target application. Related concepts to transfer learning are *zero-shot*, *one-shot*, and *few-shot learning*, which use (learned) prior knowledge to predict new targets while seeing few or no examples of them. *Self-supervised learning* uses pretext tasks on unlabeled data to learn representations that generalize well to downstream tasks, leading to lower labeled data requirements. Finally, *active learning* performs queries to a user to collect labels in an efficient manner, by having an algorithm guide which samples should be labeled. Of late, the active learning community has developed new ideas and approaches to reduce required labeling efforts even further and to scale the labeling process, e.g. by incorporating richer forms of feedback such as information regarding feature importance [82, 217, 235, 213, 64], or learning from comparisons [206, 277, 276]. All of the aforementioned learning paradigms are important and promising steps for reducing the reliance on large labeled datasets. However, none of them fundamentally address the drawbacks of the process of manual data annotation. As in active learning, labeling data is still predominantly pointillistic in nature, meaning that each human query response annotates a single instance (or, as in [206, 277, 276], pairs of instances). The labeling effort is optimized for a human annotator's effort, but the labeling step does not become easier, does not scale, and if a problem definition changes, data often has to be relabeled from scratch.

To use domain knowledge efficiently, ML pipelines should be able to consume and learn from a variety of sources of information. At times, some forms of knowledge such as domain heuristics or constraints may be much cheaper and feasible to obtain than labeled data. We should thus enable subject matter experts to use these sources to train and improve their models. Furthermore, it may frequently be the case that this information is easier to obtain at scale, such as when experts are able to formulate heuristics that automatically assign approximate labels to data, or when meta data can be used to automatically derive such approximations. Collecting annotated data at scale, even if labels are approximate, is promising as research has shown that the scale of data can overcome some noise in the label space [222, 247]. Finally, adequate solutions to the labeled data bottleneck will not just help us train models cheaper and faster, but will also lead to more flexible model development iterations and deployment across the board, even in applications where manual data labeling is currently the norm.

This thesis posits that weak supervision provides alternative pathways for acquiring domain knowledge upon which scalable learning mechanisms can be built to train ML models quickly and efficiently. In the following chapters, I propose novel methodologies for the use and acquisition of a variety of forms of weak supervision signals, and show that the methods lead to improved data exploration, improved modeling of unobserved ground truth, and to drastic reduction of user effort.

Weak supervision and learning from *weak labels* are umbrella terms that refer to learning with partial, indirect, or imprecise signals about an unobserved ground truth variable (see e.g. [295]). The forms of weak supervision studied in the literature are diverse. For example, in object detection and object localization, the term weak supervision has been used to describe settings where binary labels are available to indicate the absence/presence of object instances in an image while their exact locations are unavailable [54]. In programmatic weak supervision, also referred to as data programming, external sources of imperfect labels are aggregated into a pseudolabel to train a model [222]. Prior work has shown that weak supervision provides a promising avenue for reducing the need for humans to hand label large datasets [226, 121, 113, 275, 138, 222, 74, 183, 73, 154], and that weak supervision can reduce manual labeling efforts for a wide range of data types such as time series, images, and unstructured text [102][2][54, 138, 92].

Three forms of weak label information appear in this thesis. The first is *pairwise information about unobserved group membership*. This information is considered a weak label since the known relations about some pairs of points in a dataset reveal samples that should or should not belong to the same group, but the knowledge does not reveal the specific labels. For example, we may be assigning classes to emails and know that two messages x_1 and x_2 are highly likely to be of the same class as they were received from the same sender with the same subject, but this information does not allow us to infer which class that is. In clustering, this setting is referred to as constrained clustering or semi-supervised clustering [259, 16, 147][23]. As explained in Chapter 2, this pairwise linkage information can at times be easier to obtain than labels for individual samples.

The second form of weak label information that this thesis focuses on comes in the form of *variables that provide a direct but imperfect view of the underlying latent variable*. This is to say that such a source of weak labels can be used to infer approximate class labels for all or a subset of one's unlabeled dataset. If one has

[2] References to my work are highlighted in blue.

access to multiple such weak supervision sources, the chief objective is to combine them in an intelligent way in order to obtain a good estimate of the unobserved latent class variable. A prominent framework for learning from this type of weak supervision is Data Programming (DP) [222], which introduced the synthesis of weak supervision sources created by experts for the programmatic creation of labeled data. In Data Programming, experts create multiple so called LFs, each of which imperfectly annotates subsets of data. A factor graph is then defined to model the observed outputs of the LFs and to produce an estimate of the unobserved ground truth. Finally, this label estimate is used to train a classifier–also called end model, or downstream model–using a noise-aware loss function. DP is a fascinating paradigm to study. The use of multiple LFs not only provides a scalable framework for creating large labeled datasets, but it can also be viewed as a vehicle to incorporate high level, conceptual feedback into the data labeling process.

The third weak supervision setting that appears in this thesis concerns *paired multi-modal data with shared latent entities.* For example, we may be interested in predicting clinically relevant findings from radiographic images. In a fully supervised setting, one would have access to image annotations describing the clinically relevant findings in an image, and possibly the image region they correspond to. Instead, in the weakly supervised setting with paired modalities, one has access to radiology images and associated semi-structured text reports written by radiologists. Now, clinically relevant findings have to be learned only on the basis of knowing which report describes which image, without having the findings annotated in either modality. Other common settings where this form of weak supervision for two modalities arises are images with text captions crawled from the world wide web, or audio recordings of speeches and their text transcripts. Note that this problem setting has also been described as multi-modal self-supervised learning and is closely related to other machine learning areas such as multi-view clustering and visual grounding.

1.1 Organizational Structure

Chapter 2, Learning with Pairwise Supervision In **Chapter 2** I begin by studying the utility of indirect supervision through pairs and its application to clustering [23]. The problem setting is known as constrained clustering, and here label information is assumed to be available in the form of so called *must-link* and *cannot-link* pairs for some small amount of data, without knowledge of the ground truth cluster labels. These pairs are known as pairwise constraints. Using such pairwise constraints for data partitioning can improve results considerably since the observed constraints can be used to learn an underlying distance metric and parameters of a clustering algorithm. This is an attractive proposition as pairwise constraints can sometimes be obtained from meta-data. For example, in protein function prediction tasks one can use knowledge about functional links between proteins [86]. My interest in this form of weak supervision and constrained clustering algorithms stemmed from experiences working with subject matter experts in counter human trafficking [83, 196, 129, 24] and illicit online trade, who at times were comfortable expressing relations between data points, but were hesitant to make decisions about absolute group membership. In this work, I study the impact of the common practice of relaxing pairwise constraints in clustering objectives, where the constraints that provide information group membership (*must-link*, *cannot-link*) are relaxed to a continuous space where they inform relative distances (*must be close*, *must be distant*). I introduce a constrained kernel k-means algorithm [23] in which one learns a kernel without relaxing the pairwise constraints, and conduct experiments on over 140 datasets to demonstrate that the proposed approach outperforms related algorithms.

In the second part of **Chapter 2**, I fuse weak supervision in the form of pairwise labels and weak supervision in the form of variables that imperfectly annotate subsets of data. In particular, I focus on *programmatic weak supervision*, also known as data programming [222], in which users define multiple sources of weak supervision–called LFs–to programmatically label a dataset. Here, I introduce **pairwise LFs** and show that *imperfect* pairwise labels can be used to augment learning in label models that traditionally only learn from partial, imperfect labels. The proposed technique fuses both forms of weak supervision and improves the programmatic process of data labeling by increasing the types of weak supervision the model uses to produce estimates of the unobserved ground truth variable. I empirically demonstrate the utility of learning from standard LFs as well as pairwise LFs and show that just one source of weak pairwise feedback can significantly improve downstream test set metrics.

Chapter 3, Label Models for Programmatic Weak Supervision I continue the study of novel programmatic data labeling models, focusing on how strong inductive biases can help to model the data distribution and weak labels simultaneously. First, I present work on end-to-end learning in programmatic weak supervision. Current state-of-the-art data programming [222, 221] proceeds in two steps: a first step in which a label model is learned only on the basis of observed weak supervision votes, and a second step in which an end model is learned on the training examples and associated estimates of the unobserved ground truth obtained via the label model of the first step. I present work on an end-to-end approach for directly learning the end model by maximizing its agreement with probabilistic labels generated by a reparameterized, differentiable label model [36]. Experiments on five benchmark datasets show improved performance over prior work in terms of end model performance on downstream test sets, as well as in terms of improved robustness to dependencies among weak supervision sources.

In the second part of Chapter 3, I study the fusion of a Generative Adversarial Network (GAN) and programmatic weak supervision and propose a Weakly Supervised GAN (WSGAN) [26]. As noted in the previous paragraph, label models for programmatic weak supervision currently only model the outputs of LFs, but not the unlabeled data distribution. Thus, with a focus on image data, I study how estimates of the latent class variables may be improved by directly modeling discrete latent variables in the input data that align well with the signals encoded in the weak supervision sources, and furthermore show that this process leads to improved modeling of $p(x)$, which in turn can he used for data augmentation via synthetic samples and pseudolabels.

Chapter 4, Interactivity and Multi-Modal Learning Obtaining and structuring domain knowledge in forms that can be consumed by weak supervision learning paradigms is not straightforward. In this chapter, I study what can be viewed as two extremes on the spectrum of user involvement in order to efficiently harvest domain knowledge. First, I present work on supporting subject matter experts in finding and defining sources of weak supervision via an interactive method [25]. A practical issue with learning from user-generated LFs in data programming is that LFs are not always straightforward to design, and considerable user effort is needed to develop them. Their creation requires creativity, foresight, and domain expertise from those who hand-craft them, a process which can be tedious and subjective. Thus, I study how to aid users in discovering LFs by introducing an interactive learning framework to systematically capture subject matter experts' knowledge of the application domain in an efficient and effective fashion. I introduce an algorithm that suggests LFs and iteratively queries a user for feedback about the suggestions [25]. Experiments show that this method rapidly uncovers useful LFs that lead to improved training data ground truth estimates and end model performance on held-out test sets.

Finally, I study how unstructured natural language descriptions, such as doctors notes, can be exploited in multi-modal representation learning. Prior work has studied training of image classifiers in the medical domain by exploiting pairs of images and unstructured text to defining rules on the text documents to obtain imperfect labels and then using frameworks such as data programming to estimate the unobserved label and to learn an end model on the paired images [132, 85, 92, 88]. This is a good avenue when weak supervision sources can be defined with high accuracy and good coverage, and when the amount of classes and tasks that are targeted is limited. However, if one can develop methods to learn from the natural language descriptions directly, by exploiting knowledge about multi-modal relationships, the number of concepts that can be learned are not limited by annotation hours, and no expert time needs to be used to define weak supervision sources. Thus, with a focus on radiology images and reports, I study learning from paired image-text data directly [27]. Instead of asking experts to find and define rules on text, the methodology jointly learns image and text representations for zero-shot and few-shot classification, relying solely on the basis of the weak knowledge that stems from knowing which pairs of images and text documents go together.

1.2 Related Work

In this section, I cover related work concerning the three forms of weak supervision that appear in this thesis document. First, I discuss related pairwise weak supervision work, next I cover research in programmatic weak supervision, after which I introduce related work work in the area of paired multi-modal data. I close the chapter by discussing connections of the work presented in this thesis to the popular co-training [29] semi-supervised learning paradigm.

1.2.1 Pairwise Weak Labels

Pairwise Weak Supervision in Clustering

Clustering with weak pairwise labels–also frequently referred to as *pairwise constraints*–is an important knowledge discovery tool, and pairwise labels in this setting enable learning of kernels or distance metrics to improve clustering performance. This type of weak supervision is provided in the form of *must-link* (*ML*) and *cannot-link* (*CL*) constraints, which indicate same or different cluster membership of pairs of samples. The field of research which uses small amounts of such weak label information is often referred to as either *Semi-supervised Clustering* or *Constrained Clustering* [258, 259, 16, 147, 18, 21]. Constrained Clustering algorithms generally belong to constraint-based and/or distance-based approaches, where the former do not include learning of an underlying metric. While not the focus of this document, Constrained Clustering has also been studied under the availability of cluster-level constraints [69], for scenarios where constraints are obtained from different sources [12], and in settings where small sets of cluster label information are available [90, 68, 172].

In constraint-based algorithms, a constraint sensitive assignment of samples to clusters is performed, in order to reduce violations of known constraints with the goal of learning a partitioning function that does well on unobserved constraints. For example, in a constrained k-means algorithm, a constraint-sensitive assignment of samples to clusters may lead to better cluster centers, and therefore to better data partitioning (as in e.g. [208]). The literature often differentiates between hard and soft versions of constraint imposition, where the latter allow for some violations of know constraints while the former does not.

Purely distance-based algorithms such as Mahalanobis Metric Learning for Clustering (MMC) [274] and Information-Theoretic Metric Learning (ITML) [71] separate metric learning from the clustering step. MMC learns a Mahalanobis metric by minimizing the sum of squared distances between similar pairs under the constraint that the sum over dissimilar pairs is kept above some constant. ITML learns a Mahalanobis distance metric that is close to a given initial one, and uses slack variables to keep distances between similar pairs within some margin while maintaining a greater margin between dissimilar pairs. While some authors refer to pairwise constraints as similar/dissimilar points, these pairs are generally assumed to stem from the same/different cluster.

The exclusion of unlabeled data from the metric learning step motivated the introduction of joint metric learning and clustering via pairwise constraints. Seminal work are the Hidden Markov Random Field (HMRF) k-means [18] and Metric Pairwise Constrained k-means (MPC-Kmeans) [21] algorithms which jointly learn a metric and cluster assignment via pairwise constraints. HMRF k-means can be adapted to learn a variety of distortion measures, including Bregman divergences. MPC-Kmeans, which is closely related to HMRF k-means, learns a cluster-specific Mahalanobis distance which allows for clusters to lie in different subspaces. This concept of learning cluster-specific metrics was later also suggested for use in HMRF k-means [17].

Joint clustering and metric learning formulations based on or similar to HMRF k-means iteratively adapt a metric or kernel according to a cluster loss as well as scaled penalties for violating the constraints (e.g. [17, 278]). Adapting the pairwise metric to reduce cluster loss may allow constraint information to be propagated to good initial cluster assignments. But it could also reinforce false cluster assignments. The inclusion of the cluster loss in pairwise metric learning also means that careful measures need to be taken to avoid trivial solutions. For example, [278] aim to avoid degenerate solutions by adding a constraint to the optimization problem such that the sum of distances of all samples to a random point is greater than some constant.

Researchers have also studied kernel learning approaches to constraint clustering, in both parametric and nonparametric ways. [278] derive an adaptive semi-supervised kernel k-means algorithm (Adaptive-SS-Kernel-KMeans) inspired by HMRF k-means. It learns kernel parameters such as the scale of the Gaussian kernel. In [122], the authors propose a nonparametric approach to learn a kernel matrix using pairwise constraints. The optimization problem is set to learn a kernel matrix that is consistent with known constraints while simultaneously being consistent with an assumed known similarity function. [5] introduce a semi-supervised kernel mean shift clustering (SKMS) algorithm. For the kernel learning step, SKMS updates an initial kernel matrix to meet specified target distance values to make pairs of samples with ML/CL constraints similar/dissimilar. Like ITML, SKMS uses slack variables during this step to relax the exact distance requirement.

The use of pairwise constraints to learn improved embeddings for clustering of large datasets with deep neural networks has also been explored. [125] design a loss function to train neural networks using pairwise constraints, and [126] devise a method to perform transfer learning on unknown classes and datasets using pairwise constraints and neural networks. [91] propose to learn an autoencoder and decoder using pairwise constraints to obtain an embedding for non-centroid based clustering by optimizing a representation loss, a reconstruction loss, and the pairwise loss. These deep learning based approaches usually require large datasets to yield reliable models as well as a domain-specific design of an appropriate network architecture that fits a given problem setting and modality.

Pairwise Supervision in Active Learning

Learning with pairwise supervision has also been explored in active learning settings, often with pairwise comparisons/rankings. For example, [206] study how to learn via relative feedback. The authors learn ranking functions for each attribute and subsequently build a generative model over the joint space of ranking outputs. [277] consider an active dual supervision classification problem with algorithms that query oracles for noisy labels and pairwise comparisons. For the latter, the feedback is provided in terms of a pairwise ranking of the likelihood of being positive instead of an absolute label assignment. The algorithm can leverage both types of oracles, direct label assignment and pairwise comparisons. This active learning scheme can be useful in application areas where pairwise comparisons are easier to obtain. The comparison oracle is used to rank data points in order to create sets within which to obtain absolute labels. [276] study active learning in a regression scenario with ordinal (or comparison) information. The authors provide theoretical guarantees and introduce an algorithm for this scenario.

1.2.2 Weak Supervision as Imperfect Labels at Scale

Different from weak labels that inform pairwise relations discussed in the previous section, weak supervision has also been explored in the form of variables that are direct but imperfect observations of the latent ground truth variables [187, 226, 121, 113, 222], e.g. we have samples $x \in \mathbb{R}^d$ with an unobserved class variable $y \in \{-1, 1\}$ and for some or all of the samples we can access a variable $\lambda(x) \in \{-1, 1\}$ which provides an imperfect view of y at better than random accuracy. Prior work as studied the general problem of learning from labels that are imperfect [181, 198, 293], e.g. because the label was corrupted by an adversary. The focus of weak supervision research in this direction often differs in the motivation from adversarial settings, as imperfect labels are collected deliberately because the mechanism of obtaining the labels scales to large amounts of data, as for example in [159, 187]. One prominent example of a paradigm for obtaining noisy training data at scale is *Distant Supervision* [187, 226], which uses existing knowledge bases with known relations to collect training data consisting of noisy examples about these relations. Another popular weak supervision framework that uses multiple such imperfect sources of labels is programmatic weak supervision, also referred to as Data Programming (DP) [222], an approach that intelligently combines multiple user defined heuristics to produce an estimate of the unobserved true class label. In DP, subject matter experts specify multiple so called Labeling Functions (LFs) that imperfectly annotate the data. These LFs are functions that encode domain knowledge, such as domain heuristics or wrappers on external knowledge bases. LFs are assumed to capture partial knowledge about an unobserved ground truth variable at better than random accuracy. As such, DP has strong ties to previous paradigms including distant supervision [187, 226], crowdsourcing [72, 223, 141, 62, 290], and general heuristic and rule-based labeling of data [110, 98]. The core idea behind DP is to model the training data creation as a process via a graphical model, where the true label is a latent variable which generates the observed, noisy labels that the LFs provide. The chief technical challenge here is to learn how to combine the weak sources of labels into a high quality estimate of the latent ground truth. An end model, sometimes also called downstream classifier, is then trained on the estimate of the latent ground truth via a noise-aware loss function. [222] show that this classifier can generalize beyond the label estimate provided by its teacher (the label model), a phenomenon that has also been explored in crowdsourcing [108]. A review of DP and label models can be found in [289, 288]. Existing work shows that expert designed weak supervision sources are possible for a variety of domains and data types such as in medicine [92, 85, 88]. For example, in [92] experts crated labeling functions for Magnetic Resonance Imaging sequence data which included shape features as well as complex semantic objects such as anatomical segmentation masks. [102] shows how diagnostic models of abnormal heartbeats can be trained via human designed heuristics based on electrocardiogram data. In this thesis, I study the introduction of weak pairwise feedback into data programming in Section 2.2 learning the label model and downstream model jointly, in an end-to-end manner[36]. I also study a fusion of Generative Adversarial Networks and programmatic weak supervision[26].

In addition to providing the labeling functions, in the data programming framework users can also induce dependencies between labeling functions, such as that one 'reinforces' another. In this context, [10] propose a structure estimation method to identify the generative model's dependency structure so that the user does not have to specify it. Similarly, [255] introduce a robust PCA-based algorithm for dependency structure estimation. [35] investigated the pitfalls of learning dependency structures and find that errors due to modeling the structure can be substantial, even when when the connections that are induced are correct.

Additional related work has studied the multi-task data programming setting [221], handling of multi-resolution sources [233], addressing latent subsets in the data [252], interactive learning of weak supervision sources [25], LFs with noisy continuous scores [39], weak supervision for neural networks in information retrieval [74, 286, 285], LF generation for image data [63], exploiting small amounts of labels as in semi-supervised learning [42, 185, 186], fast model iteration in data programming via the use of pre-trained embeddings [43], LF generation through affinity functions [63], the evaluation of automated LF creation [227], and label model extensions to structured prediction settings [238].

Finally, as mentioned above, we have to note that aggregating multiple imprecise labels is also a core problem studied in the crowdsourcing literature, as modeling of crowd workers [72, 223, 141, 62, 290]. Common approaches model worker performance and the unknown label jointly [72, 62, 290] using expectation maximization (EM) or similar approaches. Some of the main differences of modeling crowd workers compared to modeling direct weak supervision sources are that errors by crowd workers are usually assumed to be random, and that task assignment to workers is not always fixed but can be optimized for.

The interactive programmatic weak supervision framework that I introduce in Chapter 4 relies on template-like structures of LFs. Prior work has emphasized that LFs defined by experts frequently have a recurring structure in which elements are swapped to change the higher level concept a function corresponds to [254, 253, 11]. As an example, in tasks involving text documents, LFs often follow a repetitive structure in which key terms or phrases and syntactical relationships change, e.g. mentions of specific words [254, 55, 255]. Prior work relies on this observation to create heuristic generators [254], LF templates [11], and domain-specific primitives [253]. In particular, in a semi-supervised data programming setting, [254] propose a system for automatic generation of labeling functions without user interaction, by using a small set of labeled data. The authors motivate their system stating that users frequently perform repetitive steps such as guessing optimal numerical thresholds and developing informative text patterns.

Active Learning and Programmatic Weak Supervision Active strategies for weak supervision sources have largely focused on combinations of data programming with traditional active learning on data points. In [197], a pool of samples is created on which LFs disagree, and active learning strategies are then applied to obtain labels for some of the samples. In [55], samples where LFs abstain or disagree most are selected and presented to users in order to inspire the creation of new LFs. The authors explore two strategies: presenting samples with maximal labeling function disagreement and samples where labeling functions abstain most. The authors find that such strategies outperform random selection of samples. One disadvantage in this setting is that it is unclear how to analyze which sample can inspire the next *best* labeling function as there is generally no explicit connection between a sample shown to a user and to the particular labeling function it may lead to. In [114], natural language explanations provided during text labeling are used to generate heuristics. The proposed system uses a semantic parser to convert explanations into logical forms, which represent labeling functions.

1.2.3 Weak Supervision as Paired Multi-modal Data

Two tasks that are closely related are *weakly supervised cross-modal alignment* and *multi-modal self-supervised representation learning*. In both cases, the learning signal comes from the knowledge of samples across the different modalities go together, e.g. a radiology image accompanied by a text report describing the clinically relevant findings in the image. Weakly supervised cross-modal alignment refers to the task of learning to match entities (and possibly their modifiers) across two or more modalities in this data, without having access to annotations for entities and their correspondence. Multi-modal self-supervised learning refers to the task of jointly learning representations for the modalities that can be used for solving downstream tasks via zero-shot learning or fine-tuning.

Section 4.2 studies a common vision-language case where one has access to pairs of images and textual descriptions, with a particular focus on applications in healthcare. A variety of self-supervised approaches have been proposed towards jointly learning visual and textual representations of paired data without supervision, such as frameworks using contrastive objectives [111, 164, 215], approaches based on joint transformer architectures [160, 163, 179, 245], self-supervised Vision Language Processing (VLP) with word-region alignment and language grounding [50], and text prediction tasks to learn image features [78]. For example, [215] use a contrastive loss over embeddings of text and image pairs to train a model on large data collected from the internet (∼400M pairs) enabling zero-shot transfer of the model to downstream tasks. Some of the proposed approaches utilise a single architecture, usually a transformer, to learn a representation, following encoders for the individual modalities [50, 163, 245]. Another common theme is the use of use cross-modal attention mechanisms to improve the aggregation of image regions in convolutional architectures [2, 65, 111]. [13] investigate how explicit modeling of the temporal structure in the paired data can improve representations and downstream performance of the models in static as well as temporal tasks.

A number of different objectives have been explored for representation learning in VLP, including the prediction of words in image captions [138], predicting phrase n-grams [159], predicting of entire captions [78], *global* contrastive objectives defined on the embeddings of the entire image and text instances [292], and combinations of global and *local* contrastive terms [128, 194], where local means that objectives are defined over text fragments (words or phrases) and image regions.

As mentioned at the beginning of this section, a task that is closely related to instance representation learning is the alignment of entities across modalities. In VLP, this is commonly referred to as *phrase grounding*, but also known as visual grounding, phrase localization, local alignment, or word–region alignment. The goal here is to connect natural language descriptions to local *image regions.* In a supervised learning setting such as in [184, 193], this problem requires expensive manual annotation for region–phrase correspondence. Thus, settings for visual grounding have been explored in which cross-modal pairs are the only form of supervision that is available [89, 111, 175, 264], i.e. the supervision signal is the knowledge of which caption belongs to which image. Much of the general domain prior work on phrase grounding relies on off-the-shelf object-detection networks [50, 65, 111, 264, 284, 294] such as Faster R-CNN [225] which are pretrained on large labeled datasets to extract region candidates from images. This considerably simplifies the problem of matching regions to phrases as the set of possible regions to match can be assumed to be known, a luxury that is often unavailable in domain specific contexts.

1.2.4 Connections to Co-training

Co-training [29] is a well-established semi-supervised learning paradigm that operates in scenarios where two independent feature sets (views) are given. Each view is assumed to have a small amount of labeled data available. In an iterative process, two models are first trained separately within each view. For a set of randomly drawn unlabeled samples, the models then label the most confident ones to create additional training data. The unlabeled set is replenished, and new models are trained in each view with the newly updated labeled set.

Since co-training operates on two views, it resembles some of the scenarios studied in this thesis. In programmatic weak supervision, the use of a set of LFs in addition to a set of features can be seen as having additional views. A first difference is that programmatic weak supervision does not assume access to any labeled data. Furthermore, LFs are assumed to operate on a diverse set of additional information, not necessarily related to the feature set, and each LF can therefore be seen as a different view of the data. Hence, in most applied scenarios of programmatic weak supervision a set of LFs will correspond to multiple views of the data. Finally, as [222] note, dependencies between the LFs/views can be explicitly modeled and learned in programmatic weak supervision, which is not the case in co-training where dependencies have been observed to cause issues [152].

In Section 4.2, I study self-supervised vision-language processing for paired image and text data. The section proposes self-supervised learning methodology to pretrain a joint image-text model on biomedical data. The image and text models, as well as their alignment, are evaluated across a broad range of downstream tasks. The setup of having paired multi-modal data studied in Section 4.2 also resembles the co-training [29] setup. However, many differences separate the two approaches and their assumptions about the problem setting. In contrast to co-training, in Section 4.2 access to labels is not assumed. The joint image-text model is pretrained without any labels at any stage, and the models can be used in zero-shot scenarios at test time. However, some labeled data may be used to fine tune the joint model or the models for individual modalities. Additionally, while the focus of co-training is classification, the focus of the joint image-text model work in Section 4.2 is model-pretraining for a diverse set of downstream tasks, including phrase grounding, segmentation, and classification. Furthermore, the work in Section 4.2 does not assume that the two available views are conditionally independent given a label, which is a core assumption that enables co-training to work [29, 152]. Co-training aims to exploit conditional independence in two feature sets in order to obtain more training data. The independence assumption allows the co-training algorithm to annotate unlabeled examples similarly to drawing labeled data at random [29, 152]. In contrast, the multi-modal self-supervised learning approach of Section 4.2 follows a contrastive learning paradigm and aims to freely learn associations between the feature sets of known true pairs through a normalization term for which negative pairs are sampled at random. Thus, the data distribution for work such as studied in Section 4.2 is required to not contain dominant latent attributes that occur by themselves (e.g. medical imaging exams where images and reports show “no abnormality”) in order avoid excessive false negatives in the random sampling[3]. Therefore, such approaches are further meant to be employed in problem settings with diverse sets of latent attributes (e.g. a large number of co-occurring diseases). Co-training, in comparison, can still perform well even in the presence of a large class-imbalance. Despite these differences, for specific downstream classification tasks in a multi-modal self-supervised learning setting, the co-training paradigm may still present an opportunity to improve models further, by employing co-training during the fine-tuning step.

[3] In practice, if such an issue exists, smart filtering and mining of negatives may mitigate it. For example, a high percentage of text reports in radiology that contain ‘no finding’ can be recognized by using simple regular expressions.

Chapter 2

Learning with Pairwise Supervision

In this chapter, I first study how to learn to partition data when a small amount of pairwise annotations are available, and how it is essential to understand the information that the available pairwise links encode. I then explore the use of *imperfect* pairwise information as a novel feedback mechanism for improving latent label estimates in programmatic weak supervision.

Data annotations concerning group membership of pairs of samples are frequently referred to as pairwise constraints in related work. In particular, a relation about two samples belonging to the same group is referred to as a *must-link* constraint, while a relation stating that two samples belong to different groups is called a *cannot-link* constraint. This pairwise linkage information can be straightforward to obtain in many applications, and is at times easier to obtain than labels for individual samples. There are a variety of reasons that lead to these scenarios. Experts may feel comfortable expressing which samples should belong to the same group, but are hesitant to assign absolute labels.For example, in my personal experience researching counter sex trafficking applications that use escort advertisement data [83, 196, 129, 24], domain experts inspecting the data would frequently express that cases were highly similar, but were hesitant to assign a discrete label that an advertisement was positive for trafficking. In other cases, pairwise constraints arise naturally as linkages between objects, providing an obvious way to encode meta-knowledge. For example, for email spam detection we may not know which messages are spam or not, but we may know which messages share the same sender and similar subjects and are therefore exceedingly likely to share the same label. In the literature, spatial or temporal proximity of samples has been used to induce pairwise linkage constraints, as for example in the analysis of spectral information from planetary observations [260], in video segmentation and speaker identification [14], or for face clustering in videos [270]. Another example is the creation of pairwise constraints from knowledge about functional links between proteins for protein function prediction tasks [86].

For some approaches, such as the work I present in the second part of this chapter, pairwise linkage information may be assumed to be imperfect. In such cases, a scalable approach to creating noisy pairwise feedback is to use locally accurate similarity functions. These functions provide ways to find small numbers of similar samples of the same class with good accuracy. In text classification for example, using cosine similarity on term frequency–inverse document frequency (tf-idf) vectors is known to routinely yield a good approach for finding nearest neighbors of the same class. Such metrics can be used to gather high quality pairwise information with little noise, e.g. by constructing Mutual k-Nearest Neighbors (MKNN) graphs using the user-supplied functions. MKNN approaches have been successfully used in various domains, including in single-cell RNA sequencing [112].

2.1 Constrained Clustering and Multiple Kernel Learning without Pairwise Constraint Relaxation

> This section is based on the work presented in
> Boecking, Benedikt, Vincent Jeanselme, and Artur Dubrawski. "Constrained clustering and multiple kernel learning without pairwise constraint relaxation". In: *Advances in Data Analysis and Classification* (June 2022)

Clustering plays an important role in ML applications and systems such as for anomaly detection, dimensionality reduction, and network analysis [15]. Clustering is also frequently used as a data exploration tool to find patterns that may be validated to have meaning or interpretation by domain experts. However, clustering algorithm are entirely unsupervised and there are few established ways to incorporate inductive biases or domain knowledge into the algorithms to guide the data partitioning towards the latent concepts of interest that a user wants to partition. In many cases, a user may not know apriori what the classes or clusters in a dataset are and how many clusters should be found. Rather, the user wants a system to aid in discovering and modeling groups for them and may be willing to provide some limited amount of information to shape the discovery towards their mental model of what does and does not belong together. An important but sometimes neglected aspect of clustering is the impact of the underlying notion of similarity, e.g. an implicit assumption that the Euclidean distance is a good metric when applying the k-means algorithm. Clustering is an inherently under-specified problem where the notion of a correct grouping depends on its context. Thus, it is often unclear how to choose an appropriate similarity measure for a clustering task. While a user of a clustering algorithm may have an intuitive understanding of which instances should belong to the same clusters, it is generally difficult to map this intuition onto a metric or a feature set that would reflect such intuition well.

To this end, clustering under pairwise constraints is a knowledge discovery tool that enables the learning of appropriate kernels or distance metrics to improve clustering performance. In this part of thesis, I introduce a new constrained clustering algorithm that jointly clusters data and learns a kernel in accordance with the available pairwise constraints. To generalize well, the method is designed to maximize constraint satisfaction without relaxing pairwise constraints to a continuous domain where they inform distances. I show that the proposed method outperforms existing approaches on a large and diverse group of publicly available datasets.

A number of algorithms have been developed that can simultaneously adapt the underlying notion of similarity or distance while clustering the data. The pairwise linkage information is usually available in the form of constraints–one set of *must-link* pairs and one set of *cannot-link* pairs of data instances–and the resulting problem is generally referred to as constrained clustering or semi-supervised clustering [259] [16] [147]. Since true clusters are unknown apriori in many practical scenarios, leveraging such feedback can be intuitive and convenient, for example in assessing inter-patient similarity [262], or in information retrieval [56].

When pairwise linkage constraints are available, one not only desires a grouping of data that minimizes violations of known constraints, but a model that generalizes well to constraints beyond the observed ones. This is evident in existing work on constrained clustering with metric or kernel learning (e.g. [17, 153]), where performance improvement is measured via predicted cluster membership on data for which constraints are not known upfront. To obtain objective functions that are easier to optimize, formulations of constrained clustering with joint metric or kernel learning (e.g. [17, 21, 280]) relax pairwise linkage constraints to a continuous space, where the constraints then inform distances. The goal of learning a better metric or kernel is then formulated as making must-link pairs nearby and cannot-link pairs distant according to the resulting metric, which is a proxy to learning a clustering model that encourages the must-link pairs to belong to the same cluster and cannot-link pairs to different clusters. However, in their original form when constraints were obtained, the constraints only ever informed cluster membership, and not relative distances between samples. The relaxation of the membership constraints to distance constraints is only an approximation to the information the constraints encode. This relaxation can lead to over-specified constraints, e.g. a must-link constraint for samples that naturally lie on opposite ends of a cluster under a sensible metric.

To uncover patterns in data that generalize well to unseen pairwise constraints, I introduce a constrained clustering algorithm that jointly learns a cluster model and kernel by maximizing the number of satisfied training constraints–without having to relax pairwise constraints to a continuous domain. To this end, the known pairwise constraint information is used to: (1) improve cluster initialization (2) learn a kernel by measuring constraint satisfaction when it is used for unconstrained clustering. The proposed method belongs to the category of soft-constrained clustering algorithms, which allow for some amount of violation of known constraints.

The simple motivating idea behind this work is to use constraints to estimate how well a kernel uncovers the structure underlying known constraints when used for clustering. I demonstrate how to learn a kernel from a set of bases by using a kernel k-means algorithm and sparse MKL. The choice of kernel learning for the proposed method is deliberate. The kernel trick leads to the underlying clustering algorithm implicitly operating in some (possibly highly dimensional) feature space, allowing discovery of nonlinear cluster shapes. Furthermore, kernel methods can readily handle a variety of data types such as distributions [212], time series [60], trees [58], or graphs [257], and application specific kernel families have been developed such as for object recognition [232]. Additionally, MKL naturally allows the use of multiple views and transformations of the same data since kernels can be applied to varying views or feature sets. Prior research has shown the benefits of MKL over picking a kernel via cross-validation as well as benefits of sparse MKL formulations [100, 246].

Experiments are conducted on 146 publicly available benchmark datasets [204] and demonstrate that the proposed approach performs better than popular alternatives on a large variety of data. The section also shows empirically that several existing approaches frequently converge to sub-optimal metrics, i.e. by using the proposed method one can find a better solution using the same type of distance metric and the same training data. The results demonstrate that relaxing pairwise constraint labels to distance information in a continuous space can frequently yield sub-optimal pairwise metrics. Further, the experiments demonstrate that the proposed algorithm can scale well to large datasets, which is not the case for many alternative methods. Code for the proposed method is open-sourced to ensure that all results can be readily reproduced[1]. Note that from hereon, the term ‘pairwise metric’ is used as a generic term for distance, similarity, or dissimilarity function.

2.1.1 Methodology

We write vectors $\boldsymbol{x}$ in bold and matrices $\boldsymbol{X}$ in bold capital letters. We are given a dataset $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ of n samples and a number of pairs of samples $\boldsymbol{x}_i, \boldsymbol{x}_j$ known to be in either the same cluster (ML constraint, $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{M}$), or in different clusters (CL constraint, $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{C}$). These pairwise constraints may be weighted with ω_{ij} to reflect uncertainty about the relationship. The goal is to guide the clustering in a way that minimizes constraint violations. Importantly, we want the cluster model to generalize to unseen constraints.

A Multiple Kernel Learning Algorithm

I now introduce a constrained clustering algorithm termed KernelCSC, which learns a linear combination kernel. The overall objective of this algorithm is to find a kernel which leads to an unconstrained clustering that maximizes satisfaction of known pairwise constraints. Pseudo code can be found in Algorithm 1. At each iteration of

[1] Code available at github.com/autonlab/constrained-clustering.

the algorithm, we obtain a candidate kernel $\boldsymbol{K}$ parameterized by a vector $\boldsymbol{\beta}$. Using $\boldsymbol{K}$, we cluster the data via kernel k-means, and evaluate the resulting clusters $\hat{S}$ as a function of satisfied pairwise constraints such that the reward function is:

$$R(\hat{S}) = \frac{1}{|\mathcal{M}| + |\mathcal{C}|} \left(\sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{M}} \omega_{ij} \mathbb{1}[l_i = l_j] + \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{C}} \omega_{ij} \mathbb{1}[l_i \neq l_j] \right). \tag{2.1}$$

where $\hat{S}$ is is the proposed clustering, and l_i is the cluster label assigned to sample i in $\hat{S}$. As common in the related semi-supervised clustering literature, we multiply a pairwise constraint by a weight ω_{ij}, if provided.

Linear combination kernel: The MKL paradigm is used to define the kernel matrix $\boldsymbol{K}$. For now, assume that we have means to obtain a candidate parameter vector $\boldsymbol{\beta}$ defining the kernel during each iteration of Algorithm 1. Let $\mathcal{G} = \{\boldsymbol{G}_i\}_{i=1}^p, \boldsymbol{G}_i \in \mathbb{R}^{n \times n}, \boldsymbol{G}_i \succeq 0 \; \forall i \in \{1, \dots, p\}$ be a set of p Kernel matrices. Given $\boldsymbol{\beta} \in \mathbb{R}_+^p$, we can create a linear combination kernel $\boldsymbol{K} = \sum_{i=1}^p \beta_i \boldsymbol{G}_i$. Note that the analysis and experiments will be constrained to linear combination kernels, but that nonlinear combination kernels can be used as well.

Kernel k-means: Once $\boldsymbol{K}$ is defined, the unconstrained *kernel k-means* clustering step partitions data into k disjoint sets $\mathcal{S} = \{S_1, S_2, \dots, S_k\}$ for the simplified objective:

$$\underset{\{S_c\}_{c=1}^k}{\operatorname{argmin}} \; tr(\boldsymbol{K}) - \sum_{c=1}^{k} \frac{\sum_{x_i, x_j \in S_c} \boldsymbol{K}_{ij}}{|S_c|}, \tag{2.2}$$

where S_c is a set containing all elements assigned to cluster c and $tr(\boldsymbol{K})$ denotes the trace of the Gram matrix $\boldsymbol{K}$. To make good use of known constraints, centroids are initialized using the *farthest first* scheme [153]. This serves to provide better initial cluster assignments and more stable clusters for similar kernels across iterations.

Optimizing $\boldsymbol{\beta}$: Whether the proposed algorithm (pseudocode shown in Algorithm 1) performs well depends on an effective acquisition function, which is responsible for identifying promising values of $\boldsymbol{\beta}$. One can view steps 5-8 of Algorithm 1 as a function f of $\boldsymbol{\beta}$. That is, $f(\boldsymbol{\beta})$ constructs the kernel $\boldsymbol{K}$, performs kernel k-means, and returns the reward of the resulting partition via Eq. (2.1). Since this involves a clustering step, f cannot be differentiated and may be expensive to evaluate. Thus, we require an effective gradient-free optimization procedure to find good candidates for $\boldsymbol{\beta}$. One important empirical observation is that the best $\boldsymbol{\beta}$ is highly likely to be sparse. This is because the base kernels are constructed via heuristics which are not guaranteed to lead to reasonable clustering results themselves, and therefore dense $\boldsymbol{\beta}$ generally lead to bad groupings. We will therefore constrain the search space to sparse candidates $\mathcal{D} = \{\boldsymbol{\beta} : \boldsymbol{\beta} \in [0,1]^p, ||\boldsymbol{\beta}||_0 \leq c\}$.

A naive way of optimizing $\boldsymbol{\beta}$ is to sample uniformly over $\mathcal{D}$ in each iteration of the algorithm. A more sophisticated approach is to use Sequential Model Based Optimization (SMBO) principles such as introduced in [130]. At each iteration, we fit a model g to the history $\mathcal{H} = \{(\boldsymbol{\beta}_1, f(\boldsymbol{\beta}_1)), \dots, (\boldsymbol{\beta}_{t-1}, f(\boldsymbol{\beta}_{t-1}))\}$ of previously explored

Algorithm 1: Constraint Satisfaction Clustering

```
Input  : $\mathcal{G} = \{\boldsymbol{G}_1, \dots, \boldsymbol{G}_p\}$: base kernel matrices; $k$: number of clusters;
         $\mathcal{M}, \mathcal{C}$: must-link and cannot link constraint sets; $\mathcal{W}$: weights of
         pairwise constraints; $a$: acquisition function
Output: $\hat{S}^{best}, \boldsymbol{\beta}^{best}, y^{best}$
$\boldsymbol{C}_M$= $ConnectedComponents(\mathcal{M})$,;
$\mathcal{H} = \varnothing$;
while stopping criterion not met do
    $\boldsymbol{\beta} \leftarrow \text{argmax}_{\boldsymbol{\beta} \in \mathcal{D}}\, a(\beta|\mathcal{H})$ ;
    $\boldsymbol{K} \leftarrow \sum_{i=1}^{p} \boldsymbol{\beta}_i \boldsymbol{G}_i$;
    $S_{init} \leftarrow FarthestFirst(\boldsymbol{C}_M, \mathcal{M}, \mathcal{C}, \boldsymbol{K})$;
    $\hat{S} \leftarrow KernelKMeans(\boldsymbol{K}, S_{init}, k)$;
    $y \leftarrow R(\hat{S})$;
    $\mathcal{H} = \mathcal{H} \cup (\boldsymbol{\beta}, y)$;
    update $y^{best}$, $\boldsymbol{\beta}^{best}$, $\hat{S}^{best}$;
end
```

$\boldsymbol{\beta}$ values. The model g represents the prior belief about the true function f over the domain and is used as an approximation to find promising candidate values. We optimize the Upper Confidence Bound [243] over the domain of sparse vectors to obtain the next candidate:

$$a(\boldsymbol{\beta}|\mathcal{H}) = \underset{\boldsymbol{\beta} \in \mathcal{D}}{\text{argmax}}\, \mu(\boldsymbol{\beta}|\mathcal{H}) + \kappa * \sigma(\boldsymbol{\beta}|\mathcal{H}) \tag{2.3}$$

where we obtain the posterior mean $\mu(\boldsymbol{\beta}|\mathcal{H})$ and standard deviation $\sigma(\boldsymbol{\beta}|\mathcal{H})$ from g and optimize by drawing values uniformly over $\mathcal{D}$ and taking the best sample. That is, we fit a regressor g to $\mathcal{H}$ and then obtain μ and σ from the regressor for samples in $\mathcal{D}$ to find the sample maximizing the acquisition function. SMBO is well suited to applications such as the one presented here, in which optimizing g is less computationally expensive than optimizing f directly. Random Forest (RF) regression or Gaussian Process are common choices for g.

Complexity and Scalability

Assuming a negligible cost to the gradient free optimization which generates $\boldsymbol{\beta}$ candidates, the proposed method requires $\mathcal{O}(pn^2)$ storage and $\mathcal{O}(n^2(dp + t))$ computation for creating kernels and running the algorithm, with n samples in d dimensions, p base kernels, and t optimization iterations. The algorithm can be scaled to larger datasets by approximating the feature map for kernel functions, e.g. by using the Nyström method [97], leading to $\mathcal{O}(qn)$ memory and $\mathcal{O}(n(q^2p + t))$ computation, where q is the rank of the approximation. It has been shown that the use of Nyström approximations for kernel k-means is theoretically sound, practically useful, and scalable to large data sets [265]. Results showing the consistency of such an approximation and feasible runtime on a large dataset are provided in Section 2.1.2. In addition, for large datasets, one may choose to down-sample data for which no constraints are known, to further reduce training complexity.

Optional Constrained Clustering Step using a Fixed Kernel

Once Algorithm 1 has terminated, a user may choose to perform a final constrained clustering with a fixed kernel, rather than an unconstrained clustering. While optional, this can be done to satisfy more training constraints if such behavior is desirable in a particular application. We define the objective function of this *constrained kernel k-means* by adding the following penalty term $g(\boldsymbol{K}, \{S_c\}_{c=1}^k)$ to Eq. (2.2):

$$\begin{aligned} g(\boldsymbol{K}, \{S_c\}_{c=1}^k) = &\sum_{(x_i,x_j)\in\mathcal{C}} \omega_{ij}\mathbb{1}[l_i = l_j]\left(\boldsymbol{K}_{ii} - 2\boldsymbol{K}_{ij} + \boldsymbol{K}_{jj}\right) \\ &+ \sum_{(x_i,x_j)\in\mathcal{M}} \omega_{ij}\mathbb{1}[l_i \neq l_j]\left(D_{max} - \boldsymbol{K}_{ii} + 2\boldsymbol{K}_{ij} - \boldsymbol{K}_{jj}\right), \end{aligned} \tag{2.4}$$

where $D_{max} = \max\left(\{\boldsymbol{K}_{ii} - 2\boldsymbol{K}_{ij} + \boldsymbol{K}_{jj}\}_{i,j=1}^n\right)$ is the largest distance in the feature space. Pairwise constraint violation costs are scaled by distances in feature space to obtain penalties that are of similar magnitude to the distances that are observed between samples and cluster centers. This allows outliers to violate constraints. Centers are again initialized using the *farthest-first* algorithm and clusters are learned via an iterative EM-like algorithm with a greedy approach to handle constraint dependencies as in [18]. Note that, as in related work such as HMRF k-means [18] and MPC-Kmeans [21], this soft constrained formulation of kernel k-means does not guarantee the satisfaction of all training constraints.

2.1.2 Experiments

Datasets and Algorithms

The experiments make use of the Penn Machine Learning Benchmarks database (PMLB) [204] to comprehensively evaluate the proposed approach on a large number of publicly available benchmark problems covering a wide range of applications[2]. The analysis presented in this work is limited to all labeled datasets in PMLB that contain at least 100 samples, resulting in a collection of 146 datasets. With regard to the frameworks available in the vast semi-supervised clustering literature, this chapter shows comparisons to related algorithms considering their scalability and representativeness. Further, to allow for a fair comparison, the analysis is constrained to

[2] Data: `https://github.com/EpistasisLab/penn-ml-benchmarks`

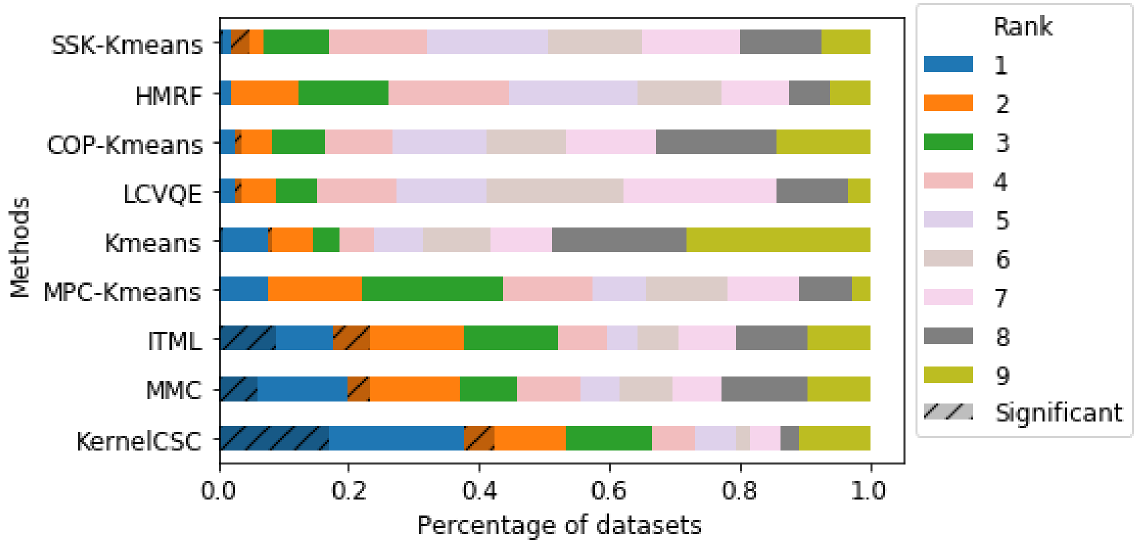


Figure 2.1: Ranks of algorithms on all 146 datasets, based on mean ARI. Shading indicates significant difference at $\alpha = 0.05$ normal confidence intervals. Ties are resolved by assigning the minimum rank.

algorithms where the number of clusters k is assumed to be given. In addition to ***k*-means**, the following algorithms are evaluated:
COP-Kmeans [259]: a hard-constrained k-means algorithm aiming to resolve all constraint violations.
LCVQE [208]: a soft constrained k-means which does not terminate if constraints are violated.
SSK-Kmeans [153]: a constrained graph clustering algorithm; cross-validation is used to choose an RBF kernel to create the input affinity matrix.
HMRF *k*-means [17] and **MPC-Kmeans** [21]: semi-supervised clustering algorithms that also perform joint metric learning.
ITML [71] and **MMC** [274]: metric learning algorithms. For both, LCVQE is used to partition the data with the learned metric.[3]

Implementation Details

The bases used in the proposed KernelCSC algorithm are the Radial Basis Function (RBF), Laplace, Polynomial, Sigmoid, and Linear kernels. Each kernel is computed on the raw data as well as on standardized data. For the parameters of each kernel, the following standard heuristics are adopted to create grids of reasonable values. For the width parameter of the RBF kernel, the median of all pairwise Euclidean

[3] A range of alternatives were evaluated in order to establish difficult baselines. A final LCVQE partitioning provided the best performance compared to other options such as k-means or COP-Kmeans partitioning.

distances is estimated and multiplied by various scaling factors. Similarly, for the Laplacian kernel multiples of the inverse of the approximate median Manahattan distance are used. For Sigmoid and Polynomial kernels, the approximate median of the inner product is used. Polynomial kernels are computed for degrees of 2 and 3. Furthermore, to avoid numerical issues, each kernel matrix in the set of base kernels is scaled by a positive scalar. To optimize $\boldsymbol{\beta}$, a random forest is used as the regressor (the model g), and $\kappa = 1.0$. Note that values in a wide range around $\kappa = 1.0$ worked well, across variety of datasets. For a fair comparison without fine-tuning, the sparsity parameter c is set to a fixed low value of 5 for all datasets.

Experimental Setup and Results

The experiments follow conventions established in related work. Algorithms are applied to the full data, but training constraints are only available between samples in a small train set, while performance is measured only on samples belonging to a test set. Training constraints are sampled uniformly at random from the binary adjacency matrix of points belonging to the training set. All algorithms are trained and evaluated on the exact same sets of constraints and test points. All algorithms are compared across a range of evaluation metrics including *Normalized Mutual Information*, *Adjusted Mutual Information*, *Adjusted Rand Index*, *Fowlkes–Mallows Index*, and *F-score*. In the main part of this thesis, I illustrate test set performance using ARI scores only, but note that the relative performance differences and overall conclusions were consistent when other evaluation metrics were used. Additional Figures using other evaluation metrics are provided in Appendix A.1.

Fixed Number of Training Constraints

The following procedure is repeated 10 times for each dataset: a stratified random split of the data is created based on the true cluster label, designating 25% as a training set. 10% of all possible pairs (up to a maximum number of 5000 pairs) are randomly selected from the training data to obtain known constraints. Constraints are augmented by transitive and entailed constraints. Wherever algorithms consider weights for constraints, unit weights are assigned. For the KernelCSC method, the maximum number of optimization iterations is set to 1000.

The scatter plots in Fig. 2.5 summarize the mean test set ARI across all datasets, showing that the proposed method outperforms other algorithms on a large number of datasets. Fig. 2.1 displays a summary of the ranks that each algorithm achieves on all datasets on the basis of the mean ARI over random trials. The proposed algorithm places first for 37.7% of the datasets, and at least second in 53.4%. For the top 3 ranks, this figure also displays the percentage of datasets for which the difference in mean ARI to all lower ranked algorithms is significant, calculated via normal confidence intervals over the random runs at $\alpha = 0.05$. MMC achieves the second-most top ranks, placing first in 17.1% of the datasets, and at least second

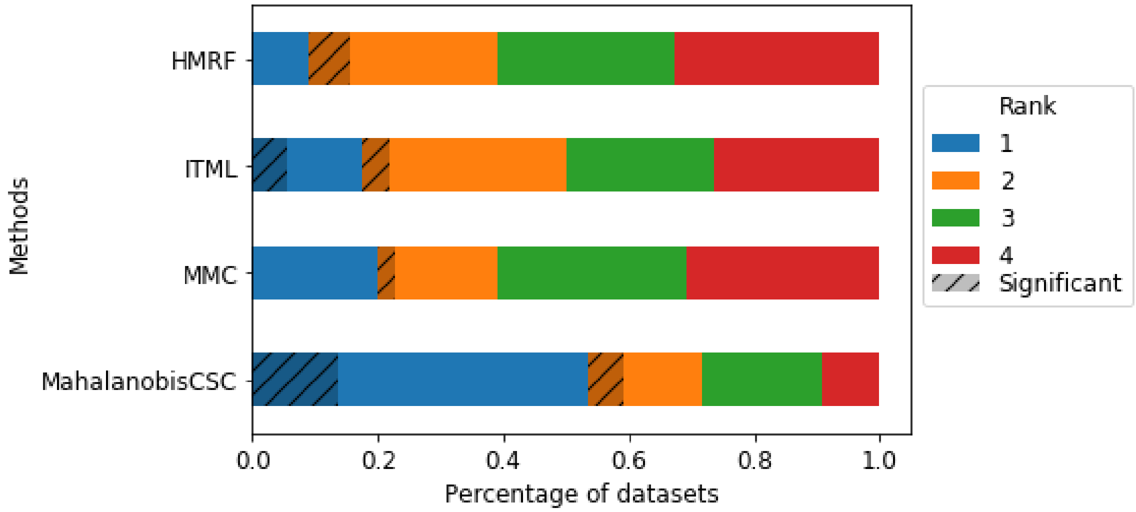


Figure 2.2: The proposed algorithm (CSC) was adapted to learn a diagonal Mahalanobis metric instead of a linear combination kernel. The bars summarize ranks achieved over all 146 datasets using mean ARI with methods using Mahalanobis metric. The results indicate that the performance improvements also observed in the MKL version stem from better generalization of learned pairwise metrics by measuring constraint satisfaction.

in 21.9%, closely followed by ITML, which obtains more significant top results than MMC and more second places. Note that–as far as tested on the datasets considered here–increasing the size of the training data and/or increasing the number of known constraints in the training fold did not affect the results in a way that would change the overall conclusions. When the number of optimization iterations of the proposed KernelCSC algorithm is set to 1000, random parameter optimization works as well as the proposed SMBO strategy.

An alternative to the MKL based version of the algorithm proposed in this work is to learn a Mahalanobis distance in conjunction with k-means. There are several disadvantages to an approach based on learning a Mahalanobis metric leading the the choice of the kernel learning version being the preferred approach, the main concern being that gradient free optimization becomes does not scale well to high dimensional datasets. However, to show that the improved clustering performance does not just stem from non-linearities introduced by using kernels or the particular bases that were chosen, but rather from better generalization of the learned similarity function, I adapt the CSC approach to also learn a Mahalanobis distance. Instead of a linear combination kernel, the algorithm then learns a diagonal projection matrix to transform the data and perform clustering via k-means instead of kernel k-means. The vector which parameterizes this Mahalanobis distance is again learned via SMBO, but without the sparsity restriction used for MKL. Fig. 2.2 provides a relative comparison

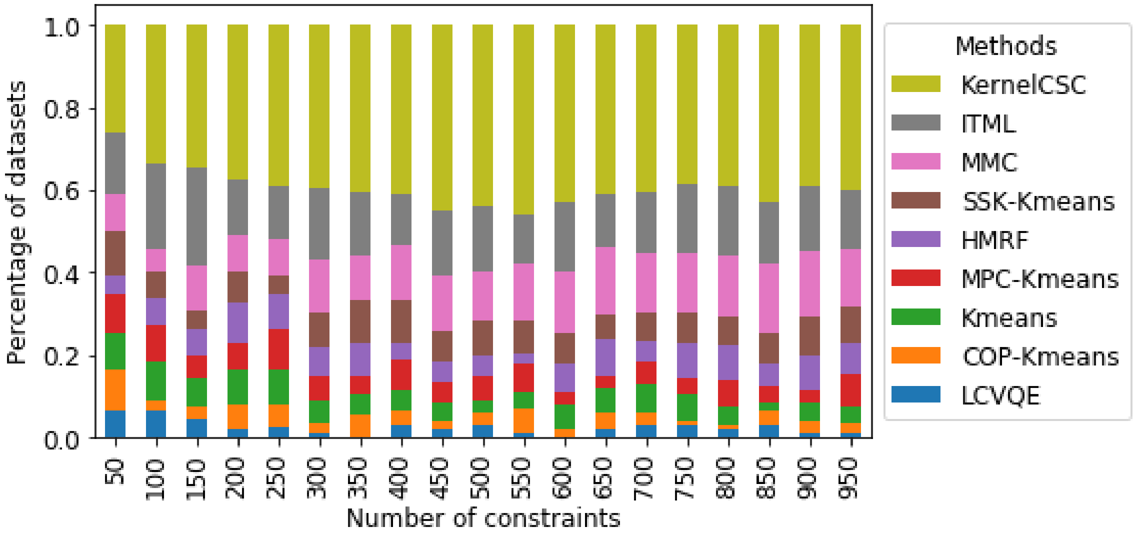


Figure 2.3: Percentage of times over all datasets each algorithm is ranked first on the test set (y-axis), vs. the number of pairwise training constraints (x-axis) used in training. The ranks were established on test-sets using mean ARI over 10 random trials.

to methods from the literature that also learn a Mahalanobis metric, showing that the proposed approach outperforms them on a large number of datasets when learning a Mahalanobis metric. Since the same training data and metrics are used, Fig. 2.2 indicates that related methods frequently converge to sub-optimal Mahalanobis metrics which do not generalize as well to unseen constraints. In the experiments of this work, the proposed methods of learning a kernel (KernelCSC) outperformed the alternative of learning a Mahalanobis metric (MahalanobisCSC) on 63.7% of the datasets.

To provide another baseline, KernelCSC is compared to an alternative approach of choosing one kernel from the set of base kernels via cross-validation. The objective here remains the same (Equation 2.1), but kernel learning is replaced with simply picking one base kernel. The results showed that MKL using a simple linear combination kernel learning approach can indeed boost performance, giving a higher mean ARI on 69.2% of datasets.

Increasing Training Set Size

To study how relative test set performance evolves as more training constraints become available, training constraints are randomly drawn from a train partition in a range from 50 to 1000 pairs, in increments of 50. Again, these experiments are repeated 10 times, and 75% of each dataset is held out for testing. Training constraints are augmented by transitive and entailed constraints. Due to the large number of experiments conducted, the maximum number of optimization iterations of KernelCSC

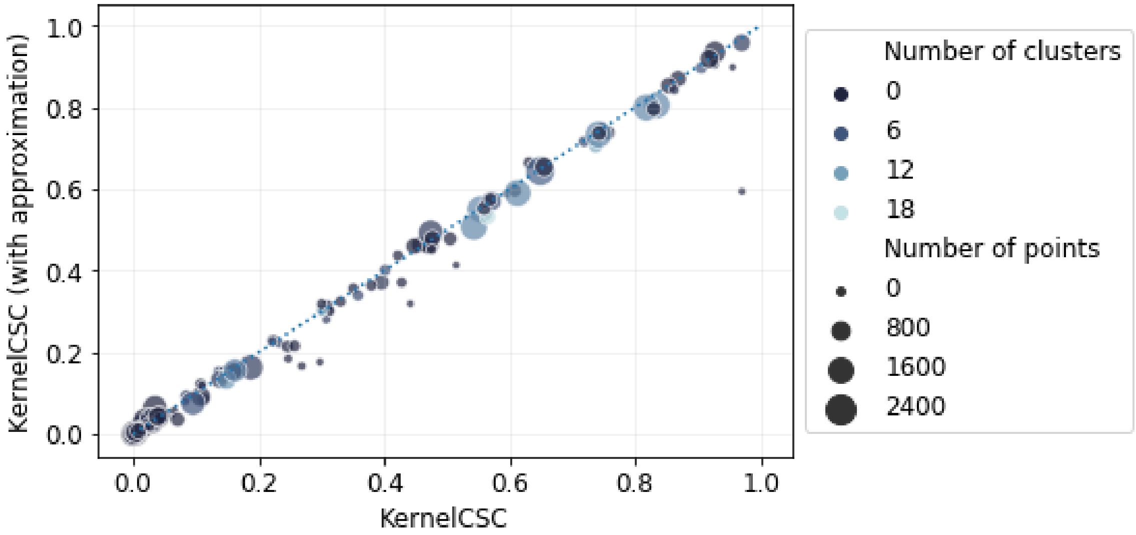


Figure 2.4: A scatter plot comparing the ARI performance of KernelCSC (x-axis) to its scalable implementation using kernel approximations (y-axis), based on test set performance.

is set to 100. Fig. 2.3 summarizes the performance of all algorithms by the percentage of datasets where each places first, showing that KernelCSC outperforms all others, regardless of the number of known constraints. The experiments also reveal that KernelCSC achieves good relative performance even with a smaller number of optimization iterations. Finally, once the metric is learned, the optional soft constraint clustering step does not significantly impact measured performance on the test sets compared to using the learned kernel with an unconstrained kernel k-means to obtain the final partition.

Scalability to Large Datasets

This section shows that the proposed KernelCSC method can scale well to large datasets, and that the required approximations to scale the method do not decrease performance considerably. Fig. 2.4 shows that an implementation of KernelCSC using Nyström approximations of each kernels' feature map produces results very similar to the exact implementation. One of the datasets contained in the PMLB benchmark database is the large *kddcup* dataset, which contains 494020 points with 23 clusters. KernelCSC can cluster this dataset using the approximate KernelCSC without any multi-processing (Intel Xeon Gold 6152 CPU), with 1000 optimization iterations and 5000 known constraints in $\sim$ 396 minutes. 62 base kernels were used, and each of the kernels was approximated with 150 components.

2.1.3 Discussion

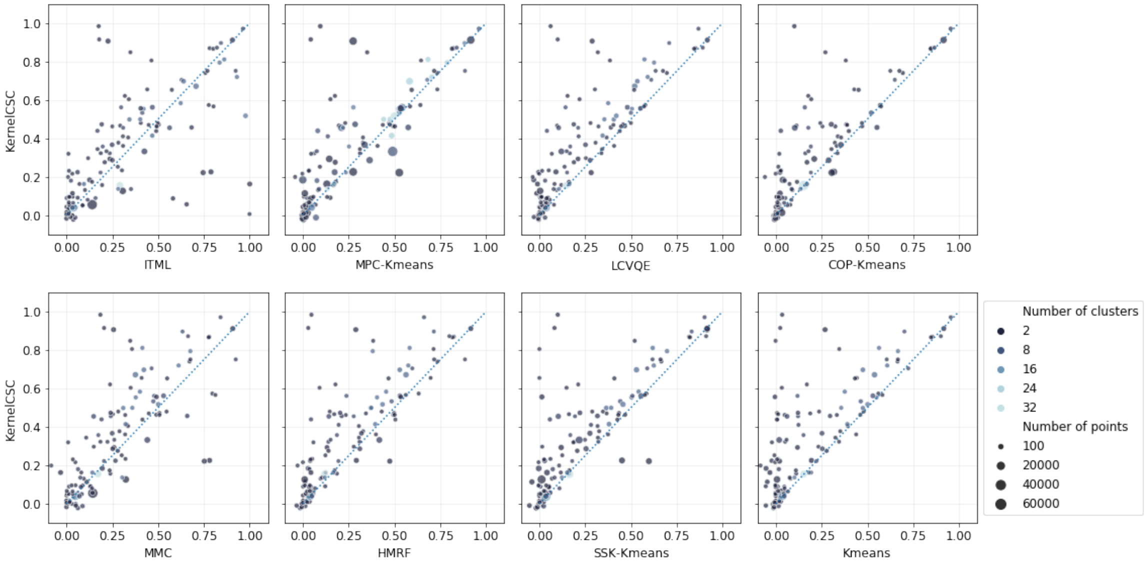


Figure 2.5: Mean test-set performance (ARI) of the proposed method (y-axes) against considered alternatives (x-axes), over all 146 datasets. When the proposed KernelCSC method performs better, points lie above the diagonal. Size and shading of each point indicate the diversity of dataset characteristics.

This section introduced a new algorithm for constrained clustering with kernel or metric learning. It uses discrete pairwise membership constraints to guide learning. I conducted experiments on 146 datasets, demonstrating superior performance of the proposed approach compared to popular alternatives. The results highlighting the importance of generalization to unseen constraints in designing constrained clustering algorithms. When pairwise constraints only indicate same or different cluster membership, a relaxation to an encoding of distances–while convenient for optimization–may lead to numerous constraints being over-specified.

The experiments show that the proposed method typically prevails over popular alternatives on a wide variety of data. An experiment into the evolution of mean test ARI per number of known training constraints shows the superior performance even if a small number of constraints are known, after only a few kernel learning iterations. The proposed approach relies on MKL to learn a kernel and can find good solutions despite relying on a gradient-free optimization. Further, the small number of base kernels which are created automatically for each dataset based on common heuristics appear to work well out of the box for a large variety of datasets.

The proposed approach can learn a kernel without relaxing pairwise constraints, and the experiments suggest that the frequently superior performance is the result of optimizing for a pairwise metric that–when used for clustering–is expected to generalize well to unseen pairwise constraints. In one set of experiments (Fig. 2.2), a diagonal Mahalanobis metric is learned in a similar vein to the proposed kerbel learbning method, by swapping the kernel matrix and kernel k-means with a simple k-means used on a learned projection of the data. This approach is not expected to work well out of the box for data with a large number of features, since the function domain becomes more difficult to optimize with gradient-free algorithms. Nonetheless, on a large number of datasets the approach outperforms related methods that also learn Mahalanobis metrics. This experiment highlights that related approaches frequently converge to sub-optimal metrics when constraints are obtained from the true underlying clusters.

There are several core issues that lead to some methods being outperformed by the proposed approach. First, many related methods are formulated to adapt a pairwise metric to decrease the distance between ML pairs and to increase distance between CL pairs. This relaxation of linkage information to distance information needs to be considered carefully. It is important to observe that the pairwise linkage constraints that guide learning generally do not encode how similar or dissimilar the pairs are but merely inform cluster membership. It is possible that algorithms overfit to pairwise constraint information when they are relaxed to a continuous space, even when slack variables are used. Second, objective functions in constrained clustering with joint metric learning often combine a clustering loss– e.g. cluster variance [18, 21, 278, 153]–and constraint violation cost of relaxed pairwise constraints. One issue is that the cluster variance minimization is performed indiscriminately for all data points in a cluster, including ones that violate constraints, which may reinforce sub-optimal solutions especially during early iterations. Further, this aspect of the objective can be decreased by simply shrinking the distances between all points, which necessitates additional steps to avoid trivial solutions. Finally, due to the cluster assignment of a sample being a function of the cluster loss and penalty terms that depend on known constraints, the cluster assignment of a point with known constraints can be different from the assignment of an equivalent point without known constraints.

The proposed Sequential Model Based Optimization strategy finds good solutions quickly. Experiments also revealed that a random parameter search over the sparse constrained space provides a good baseline, arriving at good solutions within hundreds of iterations. This strategy is especially useful for smaller datasets where the evaluation of the objective function is cheap, while SMBO is well suited to very large datasets where fewer function evaluations are desired.

Advantages of the proposed method are that–due to the use of MKL–it is straightforward to incorporate data from multiple views, and that it naturally extends to problem settings where data is not available in a simple tabular form such as in time series, or distributions. In addition, the proposed method scales gracefully to handling large datasets.

The proposed algorithm has several limitations in its present form. Keeping a number of Gram matrices or kernel approximations in memory may in practice require substantial amount of these resources. Further, its reliance on gradient-free optimization limits the number of bases in MKL or dimensionality of the data for Mahalanobis metric learning that can be handled comfortably. Promising adaptations of this approach include learning nonlinear combination kernels, exploring alternatives to the currently used gradient free optimization, and evaluating utility of the approach in semi-supervised multi-view settings.

2.2 Imperfect Pairwise Feedback for Programmatic Weak Supervision

This section is an extension of the work presented in
Benedikt Boecking and Artur Dubrawski. "Pairwise Feedback for Data Programming". In: *NeurIPS Workshop on Learning with Rich Experience (LIRE)* (2019)

In this part of the thesis, I propose the use of imperfect pairwise feedback–such as same or different class membership–to improve the estimation of true class labels in the programmatic creation of labeled datasets. Recall that DP uses multiple expert defined LFs–functions that imperfectly label subsets of the data–to estimate an unobserved ground truth variable. While DP can take dependencies between LFs into account, it does not operate on any feedback about possible relations between samples. The appeal of incorporating pairwise feedback between samples into DP is that it ties together evidence of LFs across samples. As such, even samples which do not receive many or any direct weak labels from LFs can benefit from information that has been acquired for its associated weak pairs. To this end, I propose methodology for jointly learning a label model for data labeling with noisy pairwise labels in addition to the standard labeling functions (LFs).

2.2.1 Methodology

Problem Setup

As in the standard DP problem setting [222], users provide an unlabeled training set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ with unobserved ground truth $Y \in \{1, \dots, C\}^n$. Users also provide m labeling functions $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\mathbf{x}) \in \{0, 1, ..., C\}^m$, where 0 means that the LF abstained from labeling for any class, i.e. it did not cast a label vote. Let $\Lambda \in \{0, 1, ..., C\}^{n \times m}$ be the corresponding matrix representation of LFs applied to the data.

Now, we will assume that users can write **weak pairwise feedback functions** which output an undirected, sparse graph $\boldsymbol{A}$. In the simplest case, users only write functions that indicate that pairs i, j are likely of the same class ($A_{i,j} = 1$) and abstain otherwise ($A_{i,j} = 0$), resulting in a symmetric matrix $\boldsymbol{A} \in \{0, 1\}^{n \times n}$. In some cases, users may also be able to gather reliable information about pairs i, j which are unlikely to be of the same class ($A_{i,j} = -1$), resulting in a symmetric matrix $\boldsymbol{A} \in \{-1, 0, 1\}^{n \times n}$. Finally, it may be possible to obtain functions outputting values that indicate a strength of belief that some pairs belong to the same or different class. In this work, I constrain the analysis to the use of noisy pairwise feedback that is available as a discrete signal, i.e. a function outputs that two samples are of the same or different class rather than a probability thereof, as this is the most intuitive and reliable form of pairwise feedback that users can provide.

We could model the joint distribution using a factor graph and use $\boldsymbol{A}$ to define factors among pairs of y. Unfortunately, the pairwise dependencies between the unobserved true label make factorization and marginalization very expensive, even for small n. I therefore explore a simple heuristic approach which aims to first summarize the dependencies of each sample i as a function of the observed LF votes λ.

A Heuristic Approach: Neighborhood Evidence

I now introduce a model in which we first aggregate pairwise evidence via each LFs into a variable I term Neighborhood Evidence (NE). Let

$$\bar{\boldsymbol{\lambda}} = (\mathbb{1}\{\boldsymbol{\lambda} = 1\}, \ldots, \mathbb{1}\{\boldsymbol{\lambda} = C\}) \in \{0, 1\}^{m \times C}$$

be the one-hot representation of the LF votes provided by the m LFs for C classes. Let $\bar{\Lambda} \in \{0, 1\}^{n \times m \times C}$ be its tensor representation over all samples. Additionally, users also provide p pairwise labeling functions. For one of these pairwise functions indexed by a, let $A^a \in \{-1, 0, 1\}^{n \times n}$ be its sparse output indicating if samples i and j likely belong to the same ($A^a_{ij} = 1$) or different ($A^a_{ij} = -1$) class. An output of 0 again indicates that the pairwise LF abstained from casting a vote. We will assume that the pairwise relationships encoded by the nonzero entries of each A^a, while imperfect, are better than chance.

First, we define two intermediate variables. Let $l_{j,c} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{\bar{\Lambda}_{i,j,c} \neq 0\}$ denote the propensity of LF j to vote for class c, i.e. the empirical probability that LF j outputs label c. Next, let $q^a_i = \frac{1}{|A^a_{i,\cdot}|} \sum_{k=1}^{n} A^a_{i,k} \bar{\Lambda}_{k,\cdot}$, $q_i \in \mathbb{R}^{m \times C}$, representing the fractional LF votes among sample i's neighbors encoded in A^a. The NE variable is then defined as an indicator variable $B \in \{0, 1\}^{n \times m \times C}$ as follows

$$B^a_{i,j,c} \triangleq \mathbb{1}\{q^a_{i,j,c} \geq l_{j,c}\}.$$

As such, $B_{i,j}$ casts a vote for label c if the fraction of sample i's neighbors' LF j votes for c is greater than the propensity of LF j for c. With this variable, one can use the standard accuracy and propensity factors to define a label model. The LF accuracies are modeled via factor

$$\phi^{Acc}_{i,j}(\Lambda, Y) \triangleq \mathbb{1}\{\Lambda_{ij} = y_i\}$$

and the labeling propensity by factor

$$\phi^{Lab}_{i,j}(\Lambda, Y) \triangleq \mathbb{1}\{\Lambda_{ij} \neq 0\}.$$

For notational simplicity, we will ignore propensity factors in the model below and assume only one source of pairwise labels. Naively, we can define the following labelmodel in which we treat the NE variable as another independent set of labeling functions:

$$p(Y, \Lambda, B|\theta, \gamma) \triangleq Z_{\theta,\gamma}^{-1} \exp(\sum_{i=1}^{n} \sum_{j=1}^{m} (\theta_j \phi_{i,j}^{Acc}(\Lambda, Y) + \gamma_j \phi_{i,j}^{Acc}(B, Y))). \tag{2.5}$$

Now, we know that there should be a direct relationship between the expected accuracy of an LF $\Lambda_{\cdot,j}$ and the corresponding NE variable $B_{\cdot,j}$. We can encode this in the model by defining a prior over the accuracy parameters θ, γ:

$$p(\theta, \gamma) \sim \exp\left(-(\theta - \gamma)^\top \Sigma^{-1} (\theta - \gamma)\right), \tag{2.6}$$

where $\Sigma = diag(\sigma^2, \ldots, \sigma^2)$ for a σ chosen by the user. To fit this model to data, the negative log marginal posterior can be maximized via Gradient Descent and Gibbs sampling[4].

2.2.2 Experiments

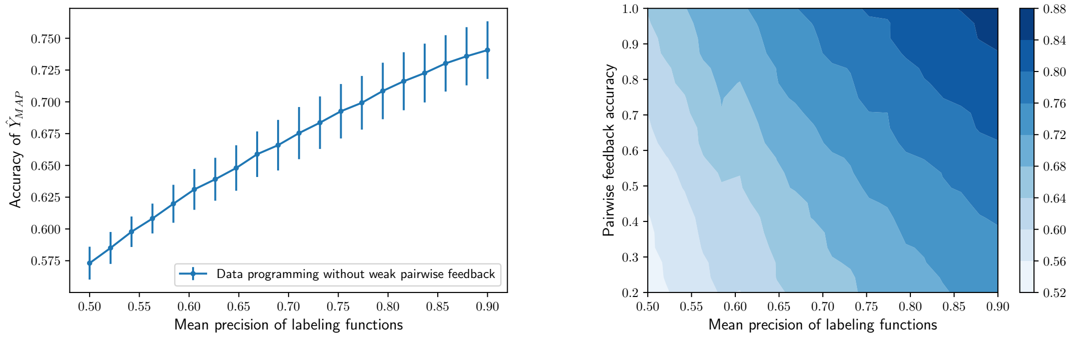
I begin with experiments on synthetic data in order to have full control over the precision and recall of the LFs. I then use three real-world datasets to show improvements in performance on real data and to demonstrate the ease with which same-class pairwise feedback can be generated in practice.

Synthetic data

Same-class feedback I simulate a multi-class classification task with $c = 10$ classes and generate two weak labeling functions per class. For each weak labeling function, false positives (fp) and false negatives (fn) are exchanged with the true label at random to achieve a target recall and precision. Given a target mean recall and precision, the specific recall and precision for one run of an experiment are drawn randomly from a truncated normal distribution. Throughout the experiments, the target recall of each labeling function is fixed at 0.5 and the target precision is incremented from 0.5 to 0.9. To create pairwise feedback, same-class information $A_{i,j} = 1$ is generated by first specifying a target count of pairs and a target accuracy and then randomly create fp and true positive (tp) pairs. To provide a point of reference for the quality of the simulated pairwise feedback one can consider that for a dataset with c classes and even class balance, the default accuracy of a pairwise matrix indicating same-class membership will be $\sim 1/c$.

This process of simulating labeling functions and weak pairwise feedback is repeated ten times, results are presented based on the mean accuracy achieved by

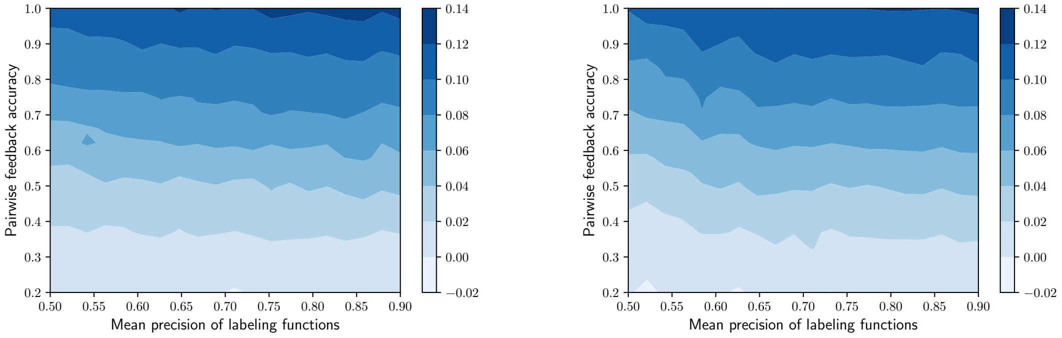
[4] For details, see: Section 20.3.3.1 of [148] or Section 19.5.2 of [195]



(a) Data programming **without** pairwise feedback. Y-axis: accuracy of latent class variable MAP estimate. X-axis: average precision of simulated labeling functions at fixed recall.

(b) The proposed model with uses pairwise feedback (same-class). Accuracy of $\hat{Y}_{MAP}$ indicated by color bar. Y-axis: accuracy of simulated pairwise feedback (5000 pairs). X-axis: average precision of simulated labeling functions at fixed recall.

Figure 2.6: **Accuracy** of MAP estimate for the latent class variable Y on synthetic data using simulated labeling functions and pairwise feedback.



(a) 1000 pairs.

(b) 5000 pairs.

Figure 2.7: **Increase in accuracy** of MAP estimate for the latent class variable Y achieved by data programming **with** pairwise feedback, compared to a model without pairwise feedback, on synthetic data. Results shown for $1k$ and $5k$ pairs. The increase in accuracy is indicated by the contours.

the estimate $\hat{Y}_{MAP}$. Figure 2.6a shows the accuracy of $\hat{Y}_{MAP}$ produced by a basic data programming model with independent labeling functions on the simulated task as the precision of the underlying labeling functions increases. The contour plot in Figure 2.6b shows the accuracy of $\hat{Y}_{MAP}$ when 5000 pairs of *same-class feedback* of varying accuracy are added to the model. Figure 2.7b shows the increase in $\hat{Y}_{MAP}$

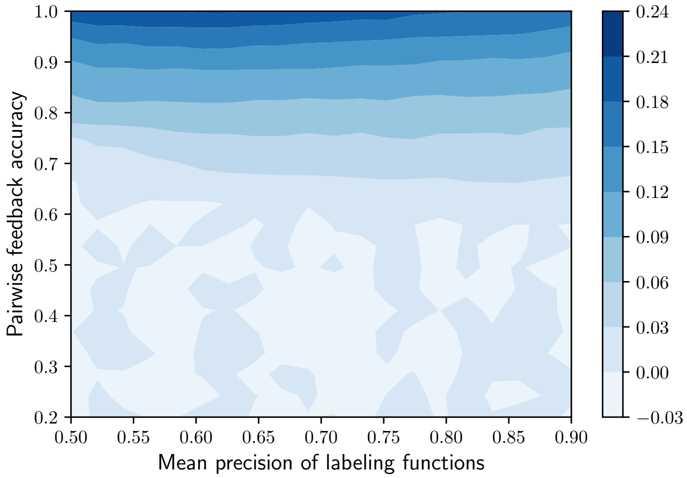


Figure 2.8: **Increase in accuracy** of MAP estimate for the latent class variable Y achieved by data programming **with same-class and different-class pairwise feedback**, compared to a model without pairwise feedback, on synthetic data. Results shown for $10k$ pairs. The increase in accuracy is indicated by the contours.

accuracy gained by using this weak pairwise feedback. Figure 2.7a reveals that even a small number of 1000 pairs can lead to drastic improvements.

Same-class and different-class feedback Using the same approach as described in the previous section, I also simulated the acquisition of different-class feedback $A_{i,j} = -1$ in addition to same-class feedback. To this end, I randomly sampled unique pairs from the data and then corrupted the ground-truth pairs to achieve desired noise levels of pairwise feedback. Results for an experiment with $10,000$ pairs are shown in Figure 2.8. Comparing Figure 2.8 to Figure 2.7, it appears that a higher pairwise accuracy is needed to achieve consistent improvements when pairwise feedback contains both same and different class feedback, as the negative feedback in this setting is naturally less informative.

Benchmark Datasets

Datasets I conduct experiments with three benchmark datasets. I use a subset of the Amazon Review Data[5] [118] to create a binary sentiment classification task, aggregating all categories with more than $100k$ reviews from which $200k$ reviews are sampled and split into $160k$ training points and $40k$ test points. I also use the IMDB Movie Review Sentiment dataset[6] [182] which has $25k$ training samples and $25k$ test

[5] `https://nijianmo.github.io/amazon/index.html`

[6] `https://ai.stanford.edu/~amaas/data/sentiment/`

Dataset	Weak Source Type	Accuracy	Coverage of all possible pairs	#pairs
Amazon	Text MKNN	0.830	$4.572e-06$	58519
IMDB	Text MKNN	0.763	$7.266e-05$	22704
Newsgroups	Text MKNN	0.880	$2.375e-3$	7466
Newsgroups	Emails	0.917	$2.956e-3$	9293

Table 2.1: Weak pairwise source stats for the Amazon, IMBD, and Newsgroups datasets. For MKNN, pairwise LFs are created by computing mutual k-nearest neighbors based on cosine similarity over tf-idf embeddings.

samples and presents another binary sentiment classification task. Finally, I also create a subset[7] of the popular 20 Newsgroups text classification dataset[8] with $c = 5$ classes with roughly even class balance, resulting in 2508 training samples and 1669 test samples from the official test split. For all three datasets, labeling functions are created manually, prior to the experiments, based on user-defined heuristics that look for mentions of specific words.

Weak Pairwise Sources For all three datasets, a weak pairwise matrix is created using an MKNN heuristic computed based on text similarities. A term frequency - inverse document frequency (tf-idf)[9] embedding is created, and the 5 nearest neighbors for all documents are computed using cosine similarity. Using this nearest neighbor graph, pairs of documents i, j are created that are MKNN[10] and assign a label $A^2_{i,j} = 1$.

For the Newsgroups dataset, I also create pairs using meta-data. I use a regular expression to extract the first email address mentioned in each document. Using the extracted email addresses, I create pairs of documents i, j containing the same email address and assign a label $A^1_{i,j} = 1$. The assumption is that the first email address in a document identifies the author, and that documents by a unique author are more likely to belong to the same topic. This results in a pairwise matrix $\boldsymbol{A}^1$ covering about 0.12% of all possible pairs, see Table 2.1. Note that both of these sources of pairwise feed back only create same-class pairs.

Results I compare the NE model to a basic DP model [222], assuming conditionally independent LFs. Table 2.2 shows how including the pairwise weak sources via the NE label model changes the estimate of the ground truth obtained for the training data. On all datasets, the accuracy across the entire dataset, including samples where LFs

[7] topics: alt.atheism, talk.religion.misc, talk.politics.misc, comp.windows.x, sci.space

[8] As available at:
`https://scikit-learn.org/stable/datasets/index.html#newsgroups-dataset`

[9] With a minimum document frequency cutoff at 0.001.

[10] i is among the 5 nearest neighbors of j, and j is among the 5 nearest neighbors of i.

Dataset	Approach	Accuracy (covered)	Accuracy (all samples)	Coverage
Amazon	DP	0.882 ± 0.000	0.634 ± 0.000	0.550
	NE	0.849 ± 0.002	0.646 ± 0.001	0.619
IMDB	DP	0.775 ± 0.001	0.663 ± 0.001	0.518
	NE	0.749 ± 0.001	0.693 ± 0.001	0.757
Newsgroups	DP	0.862 ± 0.002	0.567 ± 0.001	0.582
	NE Mail	0.834 ± 0.003	0.647 ± 0.002	00.750
	NE MKNN	0.759 ± 0.003	0.689 ± 0.002	00.892
	NE both	0.761 ± 0.003	0.701 ± 0.003	00.913

Table 2.2: Label model *transductive* performance on the training dataset, averaged over ten trials. The coverage column the fraction of the dataset where at least one LF votes. (covered) indicates that the metric is only computed on samples with at least one LF vote.

abstain, increases with the use of the NE label model, in large part due to the increase in coverage. When the accuracy of the estimate is only computed based on covered samples (i.e. samples with at least one weak labels from an LF or the NE variable), it slightly decreases compared to the accuracy of the estimate derived with the basic DP label model. Now, the important question is if the increase in coverage translates into improvements in downstream test-set accuracy. As a downstream classifier, I use tf-idf features as input to an Multilayer Perceptron (MLP), and fit the model using Adam with dropout (0.2). Table 2.3 shows the test set accuracy of the downstream classifier. On all three datasets, and across all pairwise sources, the downstream test set accuracy improves with the use of the NE label model.

Finally, I conducted ablations which showed that parameter prior which encourages the LF accuracy parameter θ and the NE parameter γ to remain close was not essential to achieving good results.

2.2.3 Discussion

Noisy pairwise feedback is a valuable resource for programmatic weak supervision as it ties evidence of the sample-wise weak votes together, across different samples of a dataset. The experiments in this section demonstrate that just one imperfect pairwise function can lead to improved downstream performance. The NE variable heuristic that I introduced aggregates the pairwise dependencies between samples as a function of the observed weak labels, allowing learning of the label model to scale to large datasets. The experiments on real data demonstrate that, for common classification tasks, pairwise feedback can be collected with ease, at scale, and with sufficient accuracy.

Dataset	Approach	Accuracy
Amazon	DP	0.718 ± 0.018
	NE	0.750 ± 0.013
IMBD	DP	0.786 ± 0.015
	NE	0.800 ± 0.007
Newsgroups	DP	0.704 ± 0.026
	NE Mail	0.751 ± 0.014
	NE MKNN	0.711 ± 0.016
	NE Both	0.721 ± 0.019

Table 2.3: Downstream test set performance averaged over ten trials.

A limitation of the current formulation of the NE label model is that it is not robust to ‘bad’ pairwise sources with random accuracy or worse. Thus, future work may investigate a model that composes the weight of the NE variable into two components: an overall accuracy of the pairwise weak source a, and the accuracy of the LF that each NE entry aggregates.

Chapter 3

Label Models for Programmatic Weak Supervision

In programmatic weak supervision, the weak sources of labels capture imperfect, partial knowledge about the unobserved ground truth at better than random accuracy, and their votes are combined by a label model to derive an estimate of the unobserved ground truth. This information bottleneck allows us to obtain a good teacher with low complexity (the label model), from which the student (the end model) can–given a sufficiently sized dataset–learn to generalize beyond the knowledge that the weak sources of labels capture. The standard DP [220, 221] paradigm proceeds in two steps: a first where a label model $p_\theta(y, \lambda)$ estimates the unobserved ground-truth y only on the basis of the votes cast by LFs λ, and a second in which an end model $f(x)$ is trained to predict the estimate of the unobserved ground truth $\hat{y} = p_\theta(y|\lambda)$ based on the features x. The quality of the teacher thus depends on how well the label model aggregates the LF votes. In current DP approaches, the teacher's complexity in how votes are aggregated is kept extremely low, as each LF is only associate with one global accuracy parameter. Furthermore, the label model parameters are estimated only on the basis of the LF outputs, ignoring the unlabeled data distribution.

In this Chapter, I will continue the study of novel programmatic weak supervision methods and investigate how carefully designed inductive biases enable us to design approaches that model not just LF vote patterns $\lambda(x)$ but also the distribution of inputs x in concert, in order to obtain an improved estimate of the unobserved ground truth y. First, I will introduce an end-to-end modeling approach for joint learning of a label and end model, showing improved performance over prior work in terms of end model performance on downstream test sets. Next, I study how Generative Adversarial Networks (GANs) enable improved modeling of pseudolabels derived from weak supervision sources, while simultaneously improving data generation of the generative models and enabling downstream data augmentation via weakly labeled synthetic samples. In both sections, I again consider weak supervision in the form of multiple direct but imperfect labels for subsets of data, supplied in the form of LFs that subject matter experts create.

3.1 End-to-end Modeling with Labeling Functions

This Section is based work with my collaborator and mentee Salva Rühling Cachay, presented in
Salva Rühling Cachay, Boecking, Benedikt, and Artur Dubrawski. "End-to-End Weak Supervision". In: *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS)*. 2021

The main task for learning from multiple sources of weak supervision is to recover the sources' accuracies in order to estimate the latent true label, without access to ground truth data. In prior work [222, 221, 93], this is achieved by first learning a generative Probabilistic Graphical Model (PGM) over the weak supervision sources and the latent true label to estimate *probabilistic labels*, which are then used in the second step to train a *downstream model* (also referred to as *end model*) via a noise-aware loss function. The existing PGM-based approaches maximize the likelihood of the observed LF votes under the label model, marginalizing over the latent ground truth. As such, the methods not take the patterns in the features or the predictions of the downstream model into account. Furthermore, while label models only based on LF votes are straightforward, the approaches and the associated theoretical analyses [222, 221, 93] make assumptions that may not hold in practice, such as the availability of a well-specified generative model structure (i.e. that the dependencies and correlations between the weak sources have been correctly specified), that LF errors are randomly distributed across samples, and that the latent label is independent of the features given the weak labels (i.e. only the joint distribution between the sources and labels needs to be modeled). The benefits of jointly optimizing a downstream model and a label model of imperfect labels have been recognized in multiple end-to-end methods that have been proposed for the crowdsourcing problem setting [223, 108, 267, 144, 3, 228, 37] where such methods have outperformed approaches that first model the latent ground truth only based on the crowd worker votes. Related work also exists in natural language processing, where [263] introduced deep probabilistic logic (DPL), a framework that uses virtual evidence as prior belief over latent labels and their inter-dependencies, and learns an end model jointly with the label model.

This section introduces `WeaSEL`, a Weakly Supervised End-to-end Learner approach for training neural networks with multiple sources of weak supervision. `WeaSEL` is based on 1) *reparameterizing previous PGM based posteriors* with a neural encoder network that produces *accuracy scores for each weak supervision source*; and 2) training the encoder and downstream model with *the same target loss, using the other model's predictions as targets*, to maximize the agreement between both models. The proposed method needs no labeled training data, and does not assume sample-independent source accuracies. Experiments show that it is not susceptible to highly correlated LFs. In addition, the proposed approach can learn from multiple probabilistic sources of weak supervision. The contributions are as follows:

- A flexible, end-to-end method for learning classifiers from multiple sources of weak supervision is introduced.
- Experiments demonstrate that the method is naturally robust to adversarial sources as well as highly correlated weak supervision sources.
- An open-source, end-to-end system for arbitrary PyTorch downstream models is released that will allow practitioners to take advantage of the proposed approach[1].
- The method outperforms state-of-the-art latent label modeling approaches on 4 out of 5 benchmark datasets by as much as 6.1 F1 points, and achieves state-of-the-art performance on a crowdsourcing dataset against methods specifically designed for this setting.

3.1.1 Methodology

This section presents a flexible base algorithm called `WeaSEL` for learning from multiple LFs, which can be extended to probabilistic sources and other network architectures (Section 3.1.3). See Algorithm 2 for pseudocode.

[1] `https://github.com/autonlab/weasel`

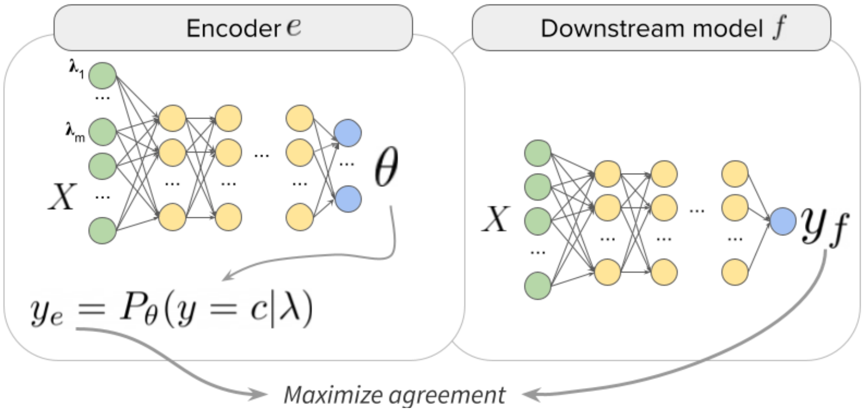


Figure 3.1: For a task with unobserved ground truth labels y, given m sources of weak supervision λ_i and training features X, `WeaSEL` trains a downstream model f by maximizing the agreement of its predictions y_f with probabilistic labels $y_e = P_\theta(y = c|\,\boldsymbol{\lambda})$ generated by a weighted combination of LF votes with sample-dependent accuracy scores θ produced by an encoder network e.

Algorithm 2: `WeaSEL`: The proposed Weakly Supervised End-to-end Learning algorithm for learning from multiple weak supervision sources.

```
Input  : batch size $n$, networks $e$, $f$, inverse temperatures $\tau_1, \tau_2$,
         noise-aware loss function $L$, class balance $P(y)$.
Output: downstream network $f(\cdot)$
for sampled minibatch $\{z^{(k)} = (\mathbf{x}^{(k)}, \boldsymbol{\lambda}^{(k)})\}_{k=1}^{n}$ do
    for all $k \in \{1, \dots, n\}$ do
        # Produce accuracy scores for all weak sources
        $\theta\left(z^{(k)}\right) = \text{softmax}\left(e(z^{(k)})\tau_1\right)$
        # Generate probabilistic labels
        define $\mathbf{s}^{(k)}$ as $\mathbf{s}^{(k)} = \theta(z^{(k)})^T \bar{\boldsymbol{\lambda}}^{(k)}$
        $y_e^{(k)} = P_\theta(y | \boldsymbol{\lambda}^{(k)}) = \text{softmax}\left(\mathbf{s}^{(k)}\tau_2\right) \odot P(y)$
        # Downstream model forward pass
        $y_f^{(k)} = f(\mathbf{x}^{(k)})$
    end
    $\mathcal{L}_f = \frac{1}{n}\sum_{k=1}^{n} L\left(y_f^{(k)}, \texttt{stop-grad}\left(y_e^{(k)}\right)\right)$
    $\mathcal{L}_e = \frac{1}{n}\sum_{k=1}^{n} L\left(y_e^{(k)}, \texttt{stop-grad}\left(y_f^{(k)}\right)\right)$
    update $e$ to minimize $\mathcal{L}_e$, and $f$ to minimize $\mathcal{L}_f$
end
```

Problem Setup

Let $(\mathbf{x}, y) \sim \mathcal{D}$ be the data generating distribution, where the unobserved labels belong to one of C classes: $y \in \mathcal{Y} = \{1, ..., C\}$. As in [222], users provide an unlabeled training set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, and m labeling functions $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\mathbf{x}) \in \{0, 1, ..., C\}^m$, where 0 means that an LF abstained from voting for a class. We write

$$\bar{\boldsymbol{\lambda}} = (\mathbb{1}\{\boldsymbol{\lambda} = 1\}, \dots, \mathbb{1}\{\boldsymbol{\lambda} = C\}) \in \{0, 1\}^{m \times C}$$

for the one-hot representation of the LF votes provided by the m LFs for C classes. The goal is to train a downstream model $f : \mathcal{X} \to \mathcal{Y}$ with a *noise-aware* loss $L(y_f, y_e)$ that uses the downstream model's predictions $y_f = f(\mathbf{x})$ and *probabilistic labels* y_e generated by an encoder model e that has access to LF votes, $\boldsymbol{\lambda}$, and features, $\mathbf{x}$. Note that prior work restricts the probabilistic labels to only being estimated from the LFs.

Posterior Reparameterization

Previous PGM based approaches assume that the joint distribution $p(\boldsymbol{\lambda}, y)$ of the LFs and the latent true label can be modeled as a Markov Random Field (MRF) with optional pairwise dependencies between weak supervision sources [222, 220, 221, 93, 42]. These models are parameterized by a set of LF accuracy and intra-LF correlation parameters, and in some cases by additional parameters to model LF and class label propensity. Note however, that the aforementioned models ignore features $\mathbf{x}$ when modeling the latent labels and therefore disregard that LFs may differ in their accuracy across samples and data slices.

In this work, these assumptions are relaxed, and the latent label is viewed as an *aggregation of the LF votes that is a function of the entire set of LF votes and features, on a sample-by-sample basis.* That is, we model the probability of a particular sample $\mathbf{x}$ having the class label $c \in \mathcal{Y}$ as

$$P_\theta(y = c | \, \boldsymbol{\lambda}) = \text{softmax}\left(\mathbf{s}\right)_c P(y = c), \tag{3.1}$$

$$\mathbf{s} = \theta(\boldsymbol{\lambda}, \mathbf{x})^T \bar{\boldsymbol{\lambda}} \in \mathbb{R}^C . \tag{3.2}$$

where $\theta(\boldsymbol{\lambda}, \mathbf{x}) \in \mathbb{R}^m$ weighs the LF votes on a sample-by-sample basis and the softmax for class c on s is defined as

$$\text{softmax}\left(\mathbf{s}\right)_c = \frac{\exp\left(\theta(\boldsymbol{\lambda}, \mathbf{x})^T \mathbb{1}\{\boldsymbol{\lambda} = c\}\right)}{\sum_{j=1}^{C} \exp\left(\theta(\boldsymbol{\lambda}, \mathbf{x})^T \mathbb{1}\{\boldsymbol{\lambda} = j\}\right)}.$$

While the class balance $P(y)$ is not used in the experiments, it is frequently assumed to be known in prior work [221, 93, 42], and can be estimated from a small validation set, or from unlabeled data as described in [221]. This formulation can be seen as a reparameterization of the posterior of the pairwise Markov Random Field (MRF)s in [220, 221, 93], where θ corresponds to the LF accuracies that in prior work were fixed across the dataset and were solely learned via LF agreement and disagreement signals, ignoring informative features. Appendix B.1 further motivates this formulation and expands upon the aforementioned connection.

Neural Encoder

Based on the setup introduced in the previous section and captured in Eq. (3.1), the goal is to estimate latent labels by means of learning sample-dependent accuracy scores $\theta(\boldsymbol{\lambda}, \mathbf{x})$, which this work proposes to parameterize by a neural encoder e. This network takes as input the features $\mathbf{x}$ and the corresponding LF outputs $\boldsymbol{\lambda}(\mathbf{x})$ for a data point, and outputs unnormalized scores $e(\boldsymbol{\lambda}, \mathbf{x}) \in \mathbb{R}^m$. Specifically, define

$$\theta(\boldsymbol{\lambda}, \mathbf{x}) = \tau_2 \cdot \text{softmax}\left(e(\boldsymbol{\lambda}, \mathbf{x})\tau_1\right), \tag{3.3}$$

where τ_2 is a constant factor that scales the final softmax transformation in relation to the number of LFs m, and is equivalent to an inverse temperature for the output softmax in Eq. (3.1). It is motivated by the fact that most LFs are sparse in practice. When the number of LFs is large, without scaling this leads to small accuracy magnitudes (since, without scaling, the accuracies after the softmax sum up to one)[2]. τ_1 is

[2] In the main experiments of this Section $\tau_2 = \sqrt{m}$.

an inverse temperature hyperparameter that controls the smoothness of the predicted accuracy scores: the lower τ_1, the less emphasis is given to a small number of LFs – as $\tau_1 \rightarrow 0$, the model aggregates according to the equal weighted vote.

Training the Encoder

The key question now is how to train e, i.e. how can we learn an accurate mapping of the sample-by-sample accuracies, given that we do not observe any labels?

First, note that initializing e with random weights will lead to latent label estimates close to an equal weighted vote, which acts as a reasonable baseline for label models in data programming (and crowdsourcing), since in expectation votes of LFs are assumed to better than random guesses. Upon initialization, $P_\theta(y|\,\boldsymbol{\lambda}, \mathbf{x})$ will therefore provide a better than random initial guess for the unobserved true labels y. Further, *in most practical cases, features, latent label, and labeling function aggregations are intrinsically correlated due to the design decisions made by the users defining the features and LFs. Thus, one can jointly optimize e and f by maximizing their agreement with respect to the target downstream loss L in an end-to-end manner.* The natural classification loss is the cross-entropy, but in order to encode the desire to maximize the agreement of the two separate models that predict based on different views of the data, it is adapted in the following form[3]: The loss is symmetrized in order to compute the gradient of both models using the other model's predictions as targets. To that end, it is crucial to detach targets (the second argument of L) from the computation graph (sometimes regerred to as a `stop-grad` operation), i.e. to treat them as though they were ground truth labels. This choice is supported by the synthetic experiments and ablations. This operation has also been shown to be crucial in siamese, non-contrastive, self-supervised learning, both empirically [106, 49] and theoretically [250]. *By minimizing both $L(y_e, y_f)$ and $L(y_f, y_e)$ simultaneously to jointly learn the network parameters for e and the downstream model f, we learn the accuracies of the noisy sources $\boldsymbol{\lambda}$ that best explain the patterns observed in the data, and vice versa the feature-based predictions that are best explained by aggregations of LF voting patterns.*

WeaSEL Design Choices

Note that it is necessary to encode the inductive bias that the unobserved ground truth label y is a (normalized) linear combination of LF votes – weighted by sample- and feature-dependent accuracy scores. Otherwise, if the encoder network directly predicts $P_\theta(y|\,\boldsymbol{\lambda}, \mathbf{x})$ instead of the accuracies $\theta(\boldsymbol{\lambda}, \mathbf{x})$, the pair of networks e, f have no incentive to output the desired latent label, without observed labels, and do not start with a reasonable first guess of y. Of course, this two-player cooperation with strong inductive biases can still lead to degenerate solutions. However, empirically it is

[3] This holds for any asymmetric loss, while for symmetric losses this is not needed.

Table 3.1: The final test F1 performance of various multi-source weak supervision methods over seven runs, using different random seeds, are averaged out ± standard deviation. The top two performance scores are highlighted as **First**, **Second**. Triplet-median [42] is not listed, as it only converged for IMDB with 12 LFs (F1 = 73.0±0.22), and Spouse (F1 = 48.7 ± 1.0). The downstream model is the same for all methods. For Sup. (Val. set), and Majority vote it is trained on the hard labels induced by the labeled validation set and the majority vote of the LFs, respectively. For the rest it is trained on the probabilistic labels estimated by the respective state-of-the-art latent label model. For reference, the *Ground truth* performance of the same downstream model trained on the true training labels (which are unused by all other models, and not available for Spouse) is also reported.

Model	**Spouse** (9 LFs)	**ProfTeacher** (99 LFs)	**IMDB** (136 LFs)	**IMDB** (12 LFs)	**Amazon** (175 LFs)
Ground truth	–	90.65 ± 0.29	86.72 ± 0.40	86.72 ± 0.40	92.93 ± 0.68
Sup. (Val. set)	20.4 ± 0.2	73.34 ± 0.00	68.76 ± 0.00	68.76 ± 0.00	84.18 ± 0.00
Snorkel	48.79 ± 2.69	85.12 ± 0.54	**82.22 ± 0.18**	**74.45 ± 0.58**	80.54 ± 0.41
Triplet	45.88 ± 3.64	74.43 ± 10.59	75.36 ± 1.92	73.15 ± 0.95	75.44 ± 3.21
Triplet-Mean	**49.94 ± 1.47**	82.58 ± 0.32	79.03 ± 0.26	73.18 ± 0.23	79.44 ± 0.68
Majority vote	40.67 ± 2.01	**85.44 ± 0.37**	80.86 ± 0.28	74.13 ± 0.31	**84.20 ± 0.52**
WeaSEL	**51.98 ± 1.60**	**86.98 ± 0.45**	**82.10 ± 0.45**	**77.22 ± 1.02**	**86.60 ± 0.71**

observed that the simple WeaSEL model is 1) competitive and frequently outperforms state-of-the-art PGM-based and crowdsourcing models (see Tables 3.1 and 3.2); and 2) is robust against LF correlations and able to recover the performance of a fully supervised model on a synthetic example, while all related models break in this setting (see Section 3.1.2 and Appendix B.6).

3.1.2 Experiments

Datasets As in related work on label models for weak supervision [222, 221, 93, 42], for simplicity the focus here is on the binary classification case with unobserved ground truth labels $y \in \{-1, 1\}$. See Table 3.3 for details about dataset sizes and the number of LFs used. An experiment on a multi-class, crowdsourcing dataset (see Section 3.1.2) is also reported. The proposed WeaSEL end-to-end system for learning a downstream model from multiple weak supervision sources is evaluated on previously used benchmark datasets in weak supervision work [220, 25, 42]. Specifically, test set performance on the following classification datasets is evaluated:

- *The IMDB movie review* dataset [182] contains movie reviews to be classified into positive and negative sentiment. Two separate experiments are conducted, where in one the same 12 labeling functions are used as in [42], and for the other 136 text-pattern based LFs are created manually. More details on the LFs can be found in Section B.3.

- A subset of the *Amazon review* dataset [118], where the task is to classify product reviews into positive and negative sentiment.

- The *BiasBios biographies* dataset [8] is used to distinguish between binary categories of frequently occurring occupations. The same task of professor vs teacher classification is used as in [25].
- Finally, the highly unbalanced *Spouse* dataset (90% negative class labels) is used, where the task is to identify mentions of spouse relationships among a set of news articles from the Signal Media Dataset [57].

For the Spouse dataset, the same data split and LFs as in [93] are used, while a small subset of the test set is used as a validation set for the other datasets. This is common practice in related work [220, 221, 93, 25] for tuning hyperparameters, and allows for a fair comparison of models.

Benchmarking Weak Supervision Label Models

To evaluate the proposed system, it is benchmarked against state-of-the-art systems that aggregate multiple weak supervision sources for classification problems, without any labeled training data. The proposed approach is compared to the following systems: 1) *Snorkel*, a popular system proposed in [220, 221]; 2) *Triplet*, exploits a closed-form solution for binary classification under certain assumptions [93]; and 3) *Triplet-mean* and *Triplet-median* [42], which are follow-up methods based on *Triplet* with the aim of making the method more robust.

The held-out test set performance of `WeaSEL`'s downstream model f is reported. Note that, in many settings it is often not possible to apply the encoder model to make predictions at test time, since the LFs usually do not cover all data points (e.g. in Spouse only 25.8% of training samples get at least one LF vote), and can be difficult to apply to new samples (e.g. when the LFs are crowdsourced annotations). In contrast, the downstream model is expected to generalize to arbitrary unseen data points.

In the experiments, the proposed `WeaSEL` model performs well, with four out of five top scores, and a lift of 6.1 F1 points over the next best label model-based method in the Amazon dataset. The results are summarized in Table 3.1. Since the proposed model is based on a neural network, the large relative lift in performance on the Amazon review dataset may be due to it being the largest dataset in the experiments. To obtain the comparisons shown in Table 3.1, Snorkel is run over six different label model hyperparameter configurations, and the downstream model is trained on the labels estimated by the label model with the best validation AUC score. The results of the Triplet-median approach are not reported in Table 3.1 since the method only converged for the two tasks with a very small number of labeling functions. Training the downstream model on the hard labels induced by a majority vote leads to competitive performance, better than the triplet methods in four out of five datasets. This baseline is not reported in previous papers (only the raw majority

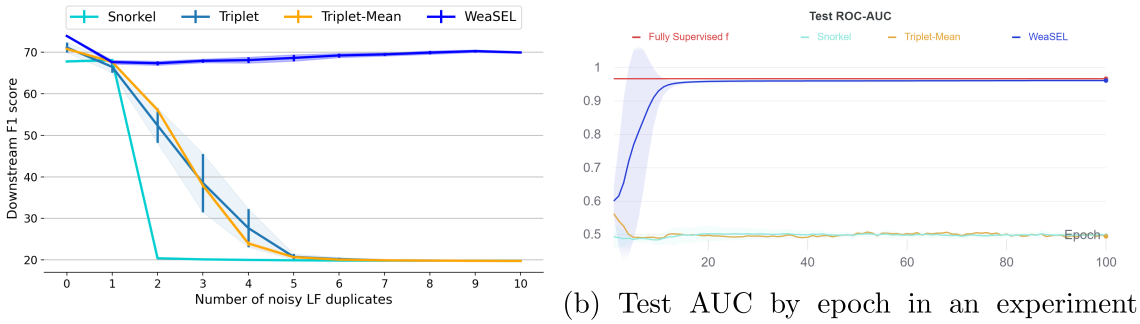
Table 3.2: Test accuracy scores on the crowd-sourced, multi-class LabelMe image classification dataset.

Model	Accuracy
Majority vote	79.23 ± 0.5
MBEM [144]	76.84 ± 0.4
DoctorNet [108]	81.31 ± 0.4
CrowdLayer [228]	82.83 ± 0.4
AggNet [3]	84.35 ± 0.4
MaxMIG [37]	**85.45 ± 1.0**
Snorkel+CE	82.89 ± 0.7
WeaSEL+CE	82.46 ± 0.8
Snorkel+MIG	85.15 ± 0.8
WeaSEL+MIG	**86.36 ± 0.3**

vote is usually reported, without training a classifier). The proposed model, WeaSEL, on the other hand consistently improves over the majority vote baseline (which in Table B.1, in the Appendix, can be seen to lead to similar performance as an untrained encoder network, e, that is left at its random initialization).

Crowdsourced Labels

Data programming and crowdsourcing methods have been rarely compared against each other, even though the problem setup is quite similar. Indeed, end-to-end systems specifically for crowdsourcing have been proposed [223, 144, 228, 37]. These methods follow crowdsourcing-specific assumptions and modeling choices (e.g. independent crowdworkers, a confusion matrix model for each worker, and in general build upon [72]). Still, since crowdworkers can be seen as a specific type of labeling functions, the performance of general WS methods on crowdsourcing datasets is of interest. The proposed WeaSEL method is therefore evaluated on the multi-class LabelMe image classification dataset, which has previously been used in related crowdsourcing work [228, 37]. The results are reported in Table 3.2, and more details on this experiment can be found in Section B.5. Note that the evaluation procedure in [37] reports the best test set performance for all models, while this work follows the standard practice of reporting results obtained by tuning based on a small validation set – as in the main experiments. The proposed model, WeaSEL, is able to outperform Snorkel as well as multiple state-of-the-art methods that were specifically designed for crowdsourcing (including several end-to-end approaches). Interestingly, this is achieved by using the mutual information gain loss (MIG) function introduced in [37], which significantly boosts performance of both Snorkel (the end-model, f,



(a) Test F1 score on robustness experiment as a function of the number of adversarial LFs.

(b) Test AUC by epoch in an experiment where one LF corresponds to the true class label and others are random.

Figure 3.2: `WeaSEL` is significantly more robust against correlated adversarial (left) or random (right) LFs than prior work whose assumptions make them equivalent to a Naive Bayes model. For subfigure (a), a fake adversarial LF is duplicated up to ten times, and the proposed end-to-end system is robust against the adversarial LF, while other systems quickly degrade in performance (over ten random seeds). In (b), one LF is set to be the true labels y^* and a random LF is then duplicated 2, 5, ..., 2000 times. The test AUC performance curve is shown as a function of the epochs, averaged out over the different number of duplicates (and five random seeds). `WeaSEL` consistently recovers the test performance of the supervised end-model f trained directly on the true labels y^*, whose end performance (AUC $= 0.967$) is shown in red.

trained on the MIG loss with respect to soft labels generated by the first Snorkel label model step) and `WeaSEL` compared to its version that uses the cross-entropy (CE) loss. This is evidence that the MIG loss is a great choice for the special case of crowdsourcing, due to its strong assumptions common to crowdsourcing which are much less likely to hold for general LFs. This is reflected in the ablations too, where using the MIG loss consistently leads to worse performance on the multi-source weak supervision datasets.

Robustness to Adversarial LFs and LF correlations

Users will sometimes generate sources they mistakenly think are accurate. This also encompasses the 'Spammer' crowdworker-type studied in the crowdsourcing literature. Therefore, it is desirable to build models that are robust against such sources. The proposed system, which is trained by maximizing the agreement between an aggregation of the sources and the downstream model's predictions, should be able to ignore the adversarial sources. Fig. 3.2a shows that the proposed system does not degrade in its initial performance, even after duplicating an adversarial LF ten times. Prior latent label models, on the other hand, rapidly degrade, given that they often assume the weak label sources to be conditionally independent given the latent label, equivalent to a Naive Bayes generative model. Note that the popular open-source implementation of [220, 221] does not support user-provided LF dependencies modeling, while [93, 42] did not converge in the experiments when modeling dependencies, and as such it was not possible to test the performance when the correlation dependencies between the duplicates are provided (which in practice, of course, are not known).

A synthetic experiment inspired by [37] is also conducted, where one LF is set to the true labels of the ProfTeacher dataset, i.e. $\lambda_1 = y^*$, while another LF simply votes according to a coin flip, i.e. $\lambda_2 \sim P(y)$. This latter LF is then simply duplicated, i.e. $\lambda_3 = \cdots = \lambda_m = \lambda_2$. Under this setting, the proposed `WeaSEL` model is able to consistently *recover the fully supervised performance* of the same downstream model directly trained on the true labels y^*, *even when the random LF is duplicated up to* 2000 *times* ($m = 2001$). The Snorkel and triplet methods, on the other hand, were unable to recover the true label (AUC $\approx$ 0.5). Importantly, the design choices for `WeaSEL` are to a large extent key in order to recover the true labels in a stable manner as in Fig. 3.2b. Various other choices either collapse similarly to the baselines, are not able to fully recover the supervised performance, or lead to unstable test performance curves, see Fig. B.3. More details about the experimental design and an extensive discussion, ablation, and figures based on the synthetic experiment can be found in Section B.6.

Implementation Details

Here, I provide a high-level overview over the used encoder architecture, the LF sets, and the features. More details, especially hyperparameter and architecture details, are provided in Section B.3. All downstream models are trained with the (binary) cross-entropy loss, and the proposed model is trained with a symmetric cross-entropy loss which detaches targets from the computational graph.

Encoder network The encoder network e does not need to follow a specific neural network architecture and a simple MLP is therefore used in the benchmark experiments.

Features for the encoder A big advantage of the proposed model is that it is able to take into account the features $\mathbf{x}$ for generating the sample-by-sample source accuracies. For all datasets, as input to the encoder model the LF votes are therefore concatenated with the same features that are used by the downstream model (for Spouse, smaller embeddings are used than the ones given to the downstream Long Short-term Memory Networks (LSTM)).

Weak supervision sources For the Spouse dataset, and the IMDB variant with 12 LFs, the same LFs are used as in [93, 42] respectively. The remaining three LF sets were selected manually prior to the experiments. These LFs are all pattern- and regex-based heuristics, while the Spouse experiments also contain LFs that are distant supervision sources based on DBPedia.

Table 3.3: Dataset details, where training, validation and test set sizes are N_{train}, N_{val}, N_{test} respectively, and f denotes the downstream model type. The total coverage Cov. of all LFs is also reported, which refers to the percentage of training samples which are labeled by at least one LF. For IMDB, two different sets of labeling functions of sizes 12 and 136 were used.

Dataset	#LFs	N_{train}	Cov. (in %)	N_{val}	N_{test}	f
Spouse	9	$22,254$	25.8	2811	2701	LSTM
BiasBios	99	$12,294$	81.8	250	$12,044$	MLP
IMDB	12	$25k$	88.0	250	$24,750$	MLP
IMDB	136	$25k$	83.1	250	$24,750$	MLP
Amazon	175	$160k$	65.5	500	$39,500$	MLP

Ablation Studies

Here, the strength of the WeaSEL model design decisions is demonstrated via extensive ablations. The ablations are conducted on four datasets (all but the Spouse dataset), for twenty configurations of WeaSEL, and with different encoder architectures, hyperparameters, and loss functions. The tabular results and a more detailed discussion than in the following can be found in Section B.4.

The ablations show that ignoring the features when modeling the sample-dependent accuracies, i.e. $\theta(\boldsymbol{\lambda}, \mathbf{x}) = \theta(\boldsymbol{\lambda})$, usually underperforms by up to 1.2 F1 points. A more drastic drop in performance, up to 4.9 points, occurs when the encoder network is linear, i.e. without hidden layers, as in [37]. It also proves helpful to scale the softmax in Eq. (3.3) by $\sqrt{m}$ via the inverse temperature parameter τ_2. Further, while the MIG loss proved important for WeaSEL to achieve state-of-the-art performance on the crowdsourcing dataset (with a similar lift in performance observable for Snorkel using MIG for downstream model training), this does not hold for the weakly supervised datasets, indicating that the assumptions encoded in the MIG loss are indeed a good choice for crowdsourcing, but not for general weakly supervised settings.

Furthermore, the ablations show that it is important to restrict the LF accuracy predictions to a positive interval (e.g. (0, 1), with the sigmoid function being a good alternative to the softmax). In contrast, using ReLU and tanh underperforms. The sigmoid function encodes the inductive bias that LFs are assumed to be better than random, and furthermore may stabilize learning by influencing the scale of the gradients.

Additionally, the choice of using the symmetric cross-entropy loss with `stop-grad` applied to the targets is crucial for the performance of WeaSEL. Not detaching the targets from the computation graph, or using the standard cross-entropy (without `stop-grad` on the target) leads to significantly worse scores and unstable training dynamics. Losses that already are symmetric (e.g. L1 or Squared Hellinger loss) neither need to be symmetrized nor use `stop-grad`. While the L1 loss consistently underperforms, the Squared Hellinger loss leads to better performance on two of the four datasets.

However, only the symmetric cross-entropy loss with `stop-grad` on the targets is shown to be robust and able to recover the true labels in the synthetic experiment in Section 3.1.2. Thus, to complement the above ablation on real datasets, extensive ablations are run on this synthetic setup in Section B.6. This synthetic ablation provides strong support for the proposed design of `WeaSEL`. Indeed, many choices for `WeaSEL` that perform well enough on the real datasets, such as ignoring features in the encoder, $\tau_2 = 1$, sigmoid parameterized accuracies, and all other losses that were evaluated, lead to significantly worse performance and less robust learning on the synthetic adversarial setups.

3.1.3 Discussion

This section introduced `WeaSEL`, an approach for end-to-end learning of neural network models for classification from multiple LFs that streamlines prior latent variable models. The proposed approach was evaluated on several benchmark datasets where downstream models outperform state-of-the-art data programming approaches in four out of five cases, while remaining highly competitive on the remaining task, and also outperforming several state-of-the-art crowdsourcing methods on a crowdsourcing task. The experiments further demonstrated that the `WeaSEL` approach can be more robust to dependencies between LFs as well as to adversarial labeling scenarios. The proposed method works with discrete and probabilistic LFs and can utilize various neural network designs for probabilistic label generation. It can simplify the process of developing effective machine learning models using weak supervision as the primary source of training signal, and help adoption of this form of learning in a wide range of practical applications.

Practical Aspects and Limitations

On why it works & degenerate solutions Overall, `WeaSEL` avoids trivial overfitting and degenerate solutions by hard-coding the encoder generated labels as a (normalized) linear combination of the m LF outputs, weighted by m sample-dependent accuracy scores. This design choice also ensures that the randomly initialized e will lead the downstream model f that is trained on soft labels generated by the random encoder, to obtain performance similar to when f is trained on majority vote labels. In fact, the random-encoder-`WeaSEL` variant itself often outperforms other baselines, and triplet methods in particular (see Section B.2).

Empirically, degenerate solutions were only observed when training for too many epochs. Early-stopping on a small validation set ensures that a good final solution is returned, and should be done whenever such a set exists or is easy to create. When no validation set is available, choosing the temperature hyperparameter in Eq. (3.3) such that $\tau_1 \leq 1/3$ was observed to avoid degenerating solutions on all datasets. This can be explained by the fact that a lower inverse temperature forces the encoder-predicted label to always depend on multiple LF votes when available, rather than a single one (which happens when the softmax in Eq. (3.3) becomes a `max` as $\tau_1 \to \infty$). This makes it harder for the encoder to overfit to individual LFs. The ablations indicate that this temperature parameter setting comes at a small cost in terms of loss in downstream performance, compared to when using a validation set for early stopping. Thus, when no validation set is available, lowering τ_1 is advised.

Complex downstream models The experiments show that `WeaSEL` achieves competitive or state-of-the-art performance on all datasets that were evaluated, for a given set of LFs. In practice, however, this LF set needs to first be defined by users. This can be done via an iterative process, where the feedback is sourced from the quality of the probabilistic labels generated by the label model. A limitation of the proposed approach is that each such iteration would require training the downstream model, f. When f is slow to train, this may slow down the LF development cycle and lead to unnecessary energy consumption. A practical solution to this can be to a) do the iteration cycle with a less complex downstream model; or b) use the fast to train PGM-based label models to choose a good LF set, and then move to `WeaSEL` in order to achieve better downstream performance.

Extensions

Probabilistic labeling functions The proposed approach can support labeling functions that output continuous scores instead of discrete labels as in [39]. In particular, this includes probabilistic sources that output a distribution over the potential class labels. This can be encoded in the proposed model by changing the one-hot representation of the base model to a continuous representation $\bar{\boldsymbol{\lambda}} \in [0, 1]^{m \times C}$.

Modeling structure explicitly While a simple MLP is used as the encoder e in the benchmark experiments, the formulation is flexible to support arbitrarily complex networks. In particular, the approach can naturally model dependencies among weak sources via edges in a Graph Neural Network (GNN), where each LF is represented by a node that is given the LF outputs as features. Furthermore, while the proposed base model outputs accuracy parameters of the sources, it is straightforward to augment $\bar{\boldsymbol{\lambda}}$ with additional sufficient statistics, e.g. the fixing or priority dependencies presented in [222, 35] which encode that one source fixes (i.e. should be given priority over) the other whenever both vote.

3.2 Generative Modeling Helps Weak Supervision (and Vice Versa)

This Section is based on work presented in:
Boecking, Benedikt, Nicholas Roberts, Willie Neiswanger, Stefano Ermon, Frederic Sala, and Artur Dubrawski. "Generative Modeling Helps Weak Supervision (and Vice Versa)". In: *International Conference on Learning Representations (ICLR)*. 2023

In this Section, I study the fusion of programmatic weak supervision with a Generative Adversarial Network (GAN)[101], and provide theoretical justification motivating this fusion. The methodology proposed in this Section captures discrete latent variables in the data alongside the weak supervision derived label estimate. Alignment of the two allows for better modeling of sample-dependent accuracies of the weak supervision sources, improving the estimate of unobserved labels. It is the first approach to enable data augmentation through weakly supervised synthetic images and pseudolabels.

Generative models enable learning data distributions which can benefit downstream tasks, e.g. via data augmentation or representation learning, in particular when learning latent factors of variation [120, 176, 127]. Intuitively, generative modeling and programmatic weak supervision should complement each other, as each can be thought of as a different approach to extracting structure from unlabeled data. However, to date there is no simple way to combine them.

Fusing generative models with weak supervision holds substantial promise. For example, it could yield large reductions in data acquisition costs for training complex models. Programmatic weak supervision replaces the need for manual annotations by

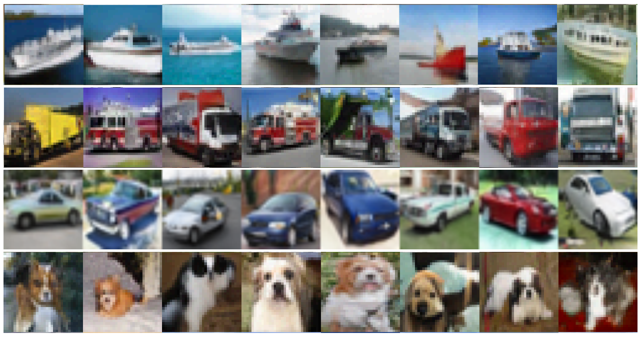


Figure 3.3: Class-conditional image generation by the proposed WSGAN based on a ***weakly supervised*** subset of CIFAR10 containing 30k samples. Here, WSGAN uses a StyleGAN2 base architecture for its networks. The sampled discrete code in each row remains fixed.

applying user designed sources of weak supervision in the form of Labeling Functions (LFs) to unlabeled data, producing weak labels that are combined into a pseudolabel for each sample. This leaves the majority of the acquisition budget to be spent on unlabeled data, and here generative modeling can reduce the number of real-world samples that need to be collected. Similarly, information about the data distribution contained in weak label sources may improve generative models, reducing the need to acquire large volumes of samples to increase generative performance and disentangle discrete structure. Additionally, pseudolabels may allow for class-conditional sample generation without access to ground truth, enabling more targeted data augmentation.

The main technical challenge is to build an *interface* between the core models used in the two approaches. For example, GANs [101], which I focus on in this work, have at least a generator and a discriminator, and frequently additional auxiliary models, such as those that learn to disentangle latent factors of variation [47]. In weak supervision, the *label model* is the main focus, which aggregates the LFs into an estimate of the unobserved ground truth. It is necessary to develop an interface that correctly aligns the structures learned from the unlabeled data by the various components.

This section introduces weakly-supervised GAN (WSGAN), a simple but powerful fusion of weak supervision and GANs visualized in Fig. 3.4. This work also provides a theoretical justification that motivates the fusion of the techniques and the expected gains. The proposed WSGAN approach is related to the unsupervised InfoGAN [47] generative model, and also inspired by encoder-based label models as introduced in the previous Section [36]. These techniques expose structure in the data, and WSGAN ensures alignment between the resulting variables by learning projections between them. The proposed method offers a number of benefits, including:

- **Improved weak supervision:** WSGAN's label model obtains better-quality pseudolabels, yielding consistent improvements in pseudolabel accuracy up to 6% over established programmatic weak supervision techniques such as Snorkel [220].

- **Improved generative modeling:** Weak supervision provides information about unobserved labels which can be used to obtain better disentangled latent variables, thus improving the model's generative performance. Over 6 datasets, WSGAN improves image generation by an average of 5.8 FID points versus InfoGAN. Architecture ablations show that the proposed approach can be integrated into state-of-the-art GAN architectures such as *StyleGAN* [143] (see Fig. 3.3), achieving state-of-the-art image generation quality.

- **Data augmentation via synthetic samples:** WSGAN can generate samples and corresponding label estimates for use in data augmentation (e.g. Fig. 3.6), providing improvements of downstream classifier accuracy of up to 3.9% in the experiments. The trained WSGAN can produce label estimates even for samples, real or fake, that have no weak supervision signal available.

3.2.1 Background

Programmatic Weak Supervision Weak supervision methods using multiple sources of imperfect and partial labels [222, 220, 36], often referred to as *programmatic weak supervision*, seek to replace manual labeling for the construction of large labeled datasets, by using sources of weak labels defined by users. The technical challenge is to combine the source votes into a high-quality pseudolabel via a *label model*. This requires estimating the errors and dependencies between sources and using them to compute a posterior label distribution. Prior work has considered various choices for the label model, most of which only take the weak source outputs into account. Instead, the label model presented in this section produces sample dependent accuracy estimates for the weak sources based on the features of the data, similar to the work introduced in the previous Section [36].

The main focus of this section is on applications of weak supervision to image data. On images, imperfect labels are often obtained from domain specific primitives and rules [254, 92], rules defined on top of annotations by surrogate models [254, 46, 123], and rules defined on meta-data [162, 48, 133, 77, 25] or a second paired modality such as text [138, 266, 132, 230, 85, 88]. Much of this work on images is motivated by the availability of data sources that contain natural language descriptions or other metadata for images. Such sources have been used in computer vision [162, 48, 133, 77, 138], medicine [266, 132, 230, 27], and video analysis [93].

Generative Models and Disentangled Representations Generative models are used to model and sample from complex distributions. Among the most popular such models are generative adversarial networks (GANs) [101]. GANs consist of a generator and discriminator model that play a minimax game against each other. In this work, we are particularly interested in prior work that aims to learn disentangled representations [47, 169] that can align with class variables of interest. [47] introduce InfoGAN, which learns interpretable latent codes. This is achieved by maximizing the mutual information between a fixed small subset of the GAN's input variables and the generated observations. [94] present a unified formulation for class and content disentanglement as well as a new approach for class-supervised content disentanglement. [200] study semi-supervised high-resolution disentanglement learning for the state-of-the-art StyleGAN architecture. A potential downside to modeling latent factors in generative models is a decrease in image quality of generated samples that has been noted when disentanglement terms are added [33, 145]. This section builds on the hypothesis that connecting discrete latent variables modeled by a GAN to the label model should yield benefits for both weak supervision and generative modeling.

Prior work has studied how to integrate additional information into GAN training,

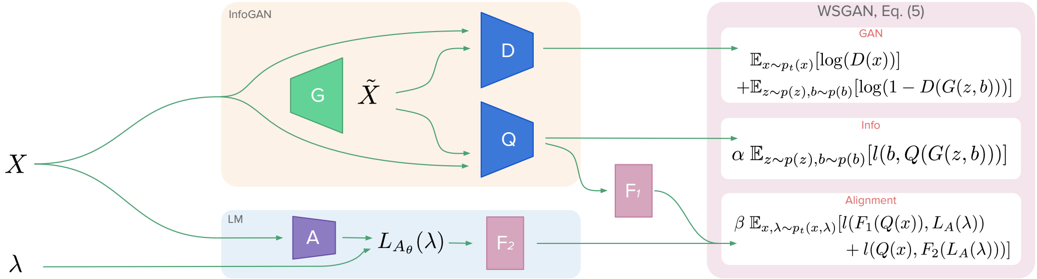

InfoGAN
G
$\tilde{X}$
D
Q
X
λ
LM
A
$L_{A_\theta}(\lambda)$
F₁
F₂
WSGAN, Eq. (5)
GAN
$\mathbb{E}_{x\sim p_t(x)}[\log(D(x))]$
$+\mathbb{E}_{z\sim p(z),b\sim p(b)}[\log(1-D(G(z,b)))]$
Info
$\alpha\,\mathbb{E}_{z\sim p(z),b\sim p(b)}[l(b,Q(G(z,b)))]$
Alignment
$\beta\,\mathbb{E}_{x,\lambda\sim p_t(x,\lambda)}[l(F_1(Q(x)),L_A(\lambda))$
$+\,l(Q(x),F_2(L_A(\lambda)))]$

Figure 3.4: The proposed WSGAN architecture models discrete latent variables in X via a network Q, while the generator G learns to fool the discriminator D with generated images $\tilde{X}$. A label model L uses weights estimated by A to produce pseudolabels based on weak supervision votes λ. The WSGAN model aligns this pseudolabel with the discrete structure learned by Q.

in particular ground truth class labels [188, 234, 202, 203, 30, 249, 190, 180], also considering noisy scenarios [139]. However, in the programmatic weak supervision setting, having multiple noisy sources of imperfect labels that include abstains present large hurdles to similar conditional modeling. Some prior work uses other weak formats of supervision to aid specific aspects of generative modeling. For example, [41] propose learning disentangled representation using user-provided ground-truth pairs. Yet, prior work does not fuse programmatic weak supervision frameworks and generative models, and so are limited to one-off techniques to solely improve generative models.

Using GANs for Data Augmentation An exciting application of GANs is to generate additional samples for supervised model training. The challenge is to produce sufficiently high-quality samples. For example, [1] use a conditional GAN to generate synthetic images of tomato plant leaves for a disease detection task. GANs for data augmentation are also popular in medical imaging [279, 192, 127]. For example, [127] use an InfoGAN-like model to learn cell-level representations in histopathology, [192] augment radiology data, and [207] generate synthetic epileptic brain activities. Data augmentation is also a potential application of the proposed WSGAN model; in contrast to prior work, it seeks to use the weak supervision to produce improved-quality samples and pseudolabels for downstream training.

3.2.2 Methodology

I will first introduce the proposed weakly-supervised GAN (WSGAN) model, visualized in Fig. 3.4, and then provide theoretical justification for the model fusion. Here, we work with n unlabeled samples $X \in \mathcal{X} \subseteq \mathbb{R}^d$ drawn from a distribution $\mathcal{D}_X$. We want to achieve two goals with the samples X. First, in generative modeling, we approximate $\mathcal{D}_X$ with a model that can be used to produce high-fidelity synthetic samples. Second, in supervised learning, we wish to use X to predict labels $Y \in \{1, 2, \ldots, C\}$, where (X, Y) is drawn from a distribution whose marginal distribution is $\mathcal{D}_X$. However, in the weak supervision setting, we do not observe Y. Instead, we observe m *labeling functions* (LFs) $\Lambda \in \{0, \ldots, C\}^{n \times m}$ that provide imperfect estimates of Y for a subset of the samples. These LFs vote on a sample x_i to produce an estimate of the label $\lambda_j(x_i) \in \{1, \ldots, C\}$ or abstain (i.e. no vote) with 0. The goal is to combine the m LF estimates into a pseudolabel $\hat{Y}$ that can be used to train a supervised model [222]. While weak supervision and generative modeling function over a number of domains, in this work I focus on images.

Proposed Method

To improve generative performance and the weak supervision-based pseudolabels, I propose a model that consists of a number of components. Because the component models should benefit each other, the architecture aims for the following characteristics: (I) A generative model component that learns discrete latent factors of variation from data and exposes these externally, (II) a weak supervision label model component that makes predictions of the unobserved label by aggregating the weak supervision votes, using sample-dependent weights, (III) a set of *interface* models that connect the components. The design choices are made to satisfy those goals.

GAN Architecture We will write G for the generator; its goal is to learn a mapping to the image space based on input consisting of samples z from a noise distribution $p_Z(z)$ along with a set of latent factors of variation $b \sim p(b)$, following the ideas introduced in InfoGAN [47]. Because a classification setting is targeted, the analysis is restricted to discrete b. The output of G are samples x; these are consumed by a discriminative model D, which estimates the probability that a sample came from the training distribution rather than G. Furthermore, an auxiliary model Q is defined which learns to map from a sample x to the discrete latent code b. Let us denote the standard GAN objective by V(D,G), and the InfoGAN objective IV(D,G,Q) [47]:

$$\min_G \max_D \; V(D, G) = \mathbb{E}_{x \sim \mathcal{D}_X}\left[\log(D(x))\right] + \mathbb{E}_{z \sim p(z), b \sim p(b)}\left[\log(1 - D(G(z, b)))\right], \quad (3.4)$$

$$\min_{G,Q} \max_D \; IV(D, G, Q) = V(D, G) + \; \alpha \, \mathbb{E}_{z \sim p(z), b \sim p(b)}\left[l(b, Q(G(z, b)))\right], \quad (3.5)$$

where l is an appropriate loss function, such as cross entropy, and α is a trade-off parameter. Equation 3.5 aims to maximize the mutual information between generated images and b, while G continues to fool the discriminator D, leading to the discovery of latent factors of variation.

Weak Supervision Label Model The purpose of the label model is to encode relationships between the LFs λ and the unobserved label y, enabling us to produce an informed estimate of y based on the LF outputs. In prior work, the model is often a factor graph [222, 221, 93, 288] with potentials $\phi_j(\lambda_j(x), y)$ and $\phi_{j,k}(\lambda_j(x), \lambda_k(x))$ capturing the degree of agreement between a λ_j and y or correlations between λ_j and λ_k. Let us define the accuracy potentials $\phi_j(\lambda_j, y) \triangleq \mathbb{1}\{\lambda_j = y\}$ as in related work. Each potential ϕ_j is associated with an accuracy parameter θ_j. Once estimates of θ_j are obtained, one can predict y from the LFs λ via

$$L_\theta(\lambda)_k = \frac{\exp(\sum_{j=1}^m \theta_j \phi_j(\lambda_j(x), k))}{\sum_{\tilde{y} \in \mathcal{Y}} \exp(\sum_{j=1}^m \theta_j \phi_j(\lambda_j(x), \tilde{y}))} \ , \quad \forall\, k \in \{1, \ldots, C\}.$$

This is a softmax over the weighted votes of all LFs, which derives from the factor graph introduced in [222]. Note that related work only models the LF outputs to learn θ, ignoring any additional information in the features x. However, the structure in the input data x is crucial to the fusion. For this reason, a modified label model predictor is defined, in the spirit of the previous section of this thesis [36]. It generates *local* accuracy parameters (sample-dependent parameters encoding how accurate each λ_i is estimated to be) via an accuracy parameter encoder $A(x) : \mathbb{R}^d \to \mathbb{R}^m_+$. This variant is given by:

$$L_{A_\theta}(\lambda)_k = \frac{\exp(\sum_{j=1}^m A(x)_j \phi_j(\lambda_j(x), k))}{\sum_{\tilde{y} \in \mathcal{Y}} \exp(\sum_{j=1}^m A(x)_j \phi_j(\lambda_j(x), \tilde{y}))} \ , \quad \forall\, k \in \{1, \ldots, C\}, \tag{3.6}$$

a softmax over the LF votes by class, weighted by the accuracy encoder output. Note that, while $A(x)$ allows for finer-grained adjustments of the label estimate $\hat{Y}$, the estimate is still anchored in the votes of LFs which represent strong domain knowledge and are assumed to be better than random.

Learning the Label Model The technical challenge of weak supervision is to learn the parameters of the label model (such as θ_j above) without observing y. Existing approaches find parameters under a label model that (i) best explain the LF votes while (ii) satisfying conditionally independent relationships [222, 221, 93]. The features x are ignored; it is assumed that all information about y is present in the LF outputs. Instead, the proposed approach promotes cooperation between the models by ensuring that *the best label model is the one which agrees with the discrete structure that the GAN can learn, and vice versa.* The intuition for this key notion is that, as each of the generative and label models learn useful information from data, this information can—if aligned correctly—be shared to help teach the other model. To this end, note that in InfoGAN, the model Q is only applied to generated samples as the sampled variable b can be observed for generated images, but not for real images. Nonetheless, Q can be applied to the real-world samples to obtain a prediction of the latent b. Crucially, in the weak supervision setting one also observes the LF outputs, enabling us derive a label estimate for each real image $L_{A_\theta}(\lambda) = \hat{Y}$, which is used to guide Q on real data, and vice versa.

Interface Models and Overall WSGAN Objective Here, I introduce the following *interface* models to map between the estimates of b and y. Let $F_1 : [0,1]^C \rightarrow [0,1]^C$ and $F_2 : [0,1]^C \rightarrow [0,1]^C$. An effective choice for F_1 and F_2 are linear models with a softmax activation function. To achieve agreement between the latent structure discovered by the GAN's auxiliary model Q as well as by the label model L_{A_θ} via the LFs, the following overall objective is introduced, ensuring that a mapping exists between the latent structures on the real images in the training data:

$$\begin{aligned} \min_{G,Q,A,F_1,F_2} \max_{D} \; & IV(D,G,Q) \qquad (3.7) \\ & + \beta \, \mathbb{E}_{x,\lambda\sim\mathcal{D}_{X,\Lambda}}[l(F_1(Q(x)), L_A(\lambda)) + l(Q(x), F_2(L_A(\lambda)))], \end{aligned}$$

where β is a trade-off parameter, and l again denotes an appropriate loss function such as the cross entropy. In the implementation, as common in related GAN work, D, Q and A share convolutional layers and distinct prediction heads are defined for each. For L_A, the obtained features are detached from the computation graph before passing it to a small MLP followed by a sigmoid activation function. Thus, the WSGAN method only adds a small number of additional parameters compared to a basic GAN or InfoGAN.

Improving Alignment Importantly, initializing the label model L_A such that it produces equal weights for all LFs results in a strong baseline estimate of $\hat{Y}$, as users build LFs to be better than random. Initializing L_{A_θ} in this way, it can act as a teacher in the beginning and guide Q towards the discrete structure encoded in the LFs. Experiments revealed that adding a decaying penalty term that encourages equal label model weights in early epochs–while not necessary to achieve good performance–almost always improves latent label estimates. Let $i \geq 0$ denote the current epoch. I propose to add the following linearly decaying penalty term for an encoder A that uses a sigmoid activation function: $C/(i \times \gamma + 1)||A(x) - \vec{1} \times 0.5||_2^2$, where γ is a decay parameter. In the experiments $\gamma = 1.5$.

Augmenting the Weak Supervision Pipeline with Synthetic Data Given a WSGAN model trained according to Eq. (3.7), images can be generated via G to obtain unlabeled synthetic samples $\tilde{x}$. To obtain pseudolabels for these images we have at least one and sometimes two options. When LFs can be applied to synthetic images, we can obtain their votes $\lambda(\tilde{x}) = \tilde{\lambda}$ and apply the WSGAN label model $L_A(\tilde{\lambda})$. However, in many practical applications of weak supervision, some LFs are not applied to images directly, but rather to metadata or an auxiliary modality such as text (cf. Section 3.2.1). With WSGAN, one can obtain pseudolabels via $\hat{y} = F_1(Q(\tilde{x}))$ for samples that have no LF votes, using the trained WSGAN components Q and F_1, in essence transferring knowledge from Q to the end model. Note that the quality of these synthetic pseudolabels hinges on the performance of Q, which can conceivably improve with the supply of weakly supervised as well as entirely unlabeled data.

Theoretical Justification

This section provides theoretical results that suggest that there is a provable benefit to combining weak supervision and generative modeling. In particular, two theoretical claims are provided, justifying why *weak supervision should help generative modeling (and vice versa)*: (1) generative models help weak supervision via a generalization bound on downstream classification and (2) weak supervision improves a multiplicative approximation bound on the loss for a conditional GAN trained using the unobserved true labels—namely, we extend the theoretical setup and noisy channel model of the Robust Conditional GAN (RCGAN) [249]. Formal statements and proofs of these claims can be found in Section C.6.

Claim (1) Assume that we have n_1 unlabeled real examples where the label model fails to produce labels, i.e. all LFs abstain on these n_1 points. This is a typical issue in weak supervision, as sources often only vote on a small proportion of points. We then sample enough synthetic examples from the generative model such that we obtain n_2 synthetic examples for which the label model *does* produce labels; this enables training of a downstream classifier on synthetic examples alone with the following generalization bound:

$$\sup_{f \in \mathcal{F}} |\hat{\mathbb{R}}_{\hat{\mathcal{D}}}(f) - \mathbb{R}_{\mathcal{D}}(f)| \leq 2\mathfrak{R} + \sqrt{\frac{\log(1/\delta)}{2n_2}} + B_\ell G^{\frac{1}{2}} + B_\ell \sqrt{2} \exp(-m\alpha^2),$$

where $\mathfrak{R}$ is the Rademacher complexity of the function class. The first two terms are standard. The third term is the penalty due to generative model usage; any generative model estimation result for total variation distance can be plugged in. For example, for estimating a mixture of Gaussians, $G = (4c_G k d^2/n_1)^{1/2}$ which depends on the number of mixture components k and dimension d. The last term is the penalty from weak supervision with m LFs whose accuracy is α better than chance; this implies that generated samples can help weak supervision generalize when true samples cannot.

Claim (2) Noisy labels from majority vote improve the multiplicative bound on the RCGAN loss given in Theorem 2 of [249]. Let P and Q be two distributions over $\mathcal{X} \times \{0, 1\}$ and let $\widetilde{P}_{\text{MV}}$ and $\widetilde{Q}_{\text{MV}}$ be the corresponding distributions with noisy labels generated by majority vote over m LFs. Let $d_{\mathcal{F}}(\widetilde{P}_{\text{MV}}, \widetilde{Q}_{\text{MV}})$ be the RCGAN loss with noisy labels generated by majority vote and let ϵ_λ be the mean error of each of the m LFs. Using majority vote with $m \geq 0.5 \log(1/\epsilon_\lambda)/\left(\frac{1}{2} - \epsilon_\lambda\right)^2$ LFs, we obtain an

Table 3.4: Datasets and labeling function (LF) characteristics used to evaluate the proposed WSGAN. Acc denotes accuracy, and Coverage denotes the proportion of samples where the LF does not abstain.

Dataset	#Classes	#LFs	#Samples	Mean LF Acc	Min LF Acc	Max LF Acc	Mean Coverage	LF Type
AwA2 -A	10	29	6726	0.504	0.053	0.850	0.104	Attribute heuristics
AwA2 -B	10	32	6726	0.548	0.116	0.783	0.131	Attribute heuristics
DomainNet	10	4	6369	0.493	0.416	0.684	1.000	Domain transfer
MNIST	10	29	30000	0.791	0.564	0.931	0.047	SSL, finetuning
FashionMNIST	10	23	30000	0.773	0.542	0.949	0.047	SSL, finetuning
GTSRB	43	100	22640	0.837	0.609	0.949	0.007	SSL, finetuning
CIFAR10-A	10	20	30000	0.773	0.624	0.896	0.061	Synthetic
CIFAR10-B	10	20	30000	0.736	0.531	0.912	0.042	SSL, finetuning

exponentially tighter multiplicative bound on the noiseless RCGAN loss:

$$\begin{aligned} d_{\mathcal{F}}(\widetilde{P}_{\mathrm{MV}}, \widetilde{Q}_{\mathrm{MV}}) \leq d_{\mathcal{F}}(P, Q) &\leq \left(1 - 2\exp\left(-2m\left(\frac{1}{2} - \epsilon_\lambda\right)^2\right)\right)^{-1} d_{\mathcal{F}}(\widetilde{P}_{\mathrm{MV}}, \widetilde{Q}_{\mathrm{MV}}) \\ &\leq (1 - 2\epsilon_\lambda)^{-1} d_{\mathcal{F}}(\widetilde{P}_{\mathrm{MV}}, \widetilde{Q}_{\mathrm{MV}}). \end{aligned}$$

This means that weak supervision can help an RCGAN more-accurately learn the true joint distribution, even when the true labels are unobserved. The full analysis is provided in Section C.6.

3.2.3 Experiments

Experiments on multiple image datasets show that the proposed WSGAN approach is able to take advantage of the discrete latent structure it discovers in the images, leading to better performance compared to label models of prior work. The results also indicate that weak supervision as used by WSGAN improves image generation performance, as well as the quality of the auxiliary model which learns disentangled discrete structure. In the spirit of democratizing AI, the aim of this work is to keep the complexity of the experiments manageable, to ensure accessible reproducibility. Therefore, the main experiments are conducted with a simple DCGAN base architecture. As an ablation, I also adapt the state-of-the art StyleGAN2-ADA [142] to WSGAN, showing that the proposed method can be integrated with other GAN architectures to achieve state-of-the-art image generation and label model performance. Additional experiments, baselines (e.g. comparing against naive combinations of weak supervision and generative modeling), and metrics are provided in Appendix C.3, C.4, and C.5. Code for the WSGAN model is made available at `https://github.com/benbo/WSGAN-paper`.

Setup

Datasets Table 3.4 shows key characteristics of the datasets used in the experiments, including information about the different LF sets. Experiments are conducted with the 3-channel color image datasets Animals with Attributes 2 (AwA2) [185], DomainNet [209], the German Traffic Sign Recognition Benchmark (GTSRB) [244], and CIFAR10 [150] as well as with the 1-channel gray-scale MNIST [156] and FashionMNIST [273] datasets. A variety of types of weak supervision sources are used for these datasets (see Appendix C.2 for more dataset details). The LF types covered in the present experiments are:

- *Domain transfer*: classifiers are trained on images in source domains (e.g. paintings), and the trained classifiers are then applied to images in a target domain (e.g. real images) to obtain weak labels. This LF type is used in the DomainNet experiments, following [185].

- *Attribute heuristics*: are used in the AwA2 experiments. Attribute classifiers are trained on some seen classes of animals. Given these weak attribute predictions, the known attribute relations and a small amount of validation data are used to train shallow decision trees to produce weak labels for a set of unseen classes of animals.

- *SSL-based*: using image features learned on ImageNET with SimCLR [45], shallow multilayer perceptron classifiers are fine-tuned on small sets of held-out data to produce weak labels for the datasets.

- *Synthetic*: these simulated LFs, used in some of the CIFAR10 experiments, are unipolar LFs based on the corrupted true class label. To this end, random errors are introduced to the class label to achieve a sampled target accuracy and propensity.

Models Two accuracy parameter versions of the proposed WSGAN model are studied: (1) *WSGAN-Encoder*, which uses an *accuracy parameter encoder* $A(x)$, that takes in the image x associated with a sample and outputs an accuracy weight vector to the label model. (2) *WSGAN-Vector*, a baseline which learns a *parameter vector* used to weigh LF votes in place of the encoder A.

For the main experiments, the base architecture of G, D follows a simple DCGAN [216]. All networks are trained from scratch and the same hyperparameter settings are used in all experiments. For the architecture ablation, StyleGAN2-ADA [142] is adapted to create StyleWSGAN. See Appendix C.1 for implementation details and parameter settings.

WSGAN is compared to the following **label model** approaches: (I) Snorkel [222, 220]: a probabilistic graphical model that estimates LF parameters by maximizing the marginal likelihood using observed LFs. (II) Dawid-Skene [72]: a model motivated by the crowdsourcing setting. The model, fit using expectation maximization, assumes that error statistics of sources are the same across classes and that errors are equiprobable independent of the true class. (III) Snorkel MeTaL [221]: a Markov

Table 3.5: Average posterior accuracy of various label models on training samples with at least one LF vote. The best result are highlighted in **blue** and the second best result in **bold**.

Dataset	MV	DawidSkene	MeTaL	FS	Snorkel	WSGAN-Vector	WSGAN-Encoder
AwA2 - A	0.631	0.607	0.632	0.615	0.641	**0.647**	**0.681**
AwA2 - B	0.623	0.548	0.582	0.602	0.605	**0.645**	**0.699**
DomainNet	0.614	**0.658**	0.487	0.635	0.499	**0.661**	0.643
MNIST	0.775	0.729	0.766	0.773	0.766	**0.782**	**0.813**
FashionMNIST	**0.735**	0.717	0.730	0.734	0.729	**0.737**	**0.744**
GTSRB	**0.816**	0.619	**0.815**	0.671	**0.814**	**0.815**	**0.823**
CIFAR10-A	0.827	**0.850**	0.806	0.800	0.807	**0.850**	**0.874**
CIFAR10-B	0.716	0.677	0.708	0.708	0.707	**0.725**	**0.731**

random field (MRF) model similar to Snorkel which uses a technique to complete the inverse covariance matrix of the MRF during model fitting, and also allows for modeling multi-task weak supervision. (IV) FlyingSquid (FS) [93]: based on a label model similar to Snorkel, FS provides a closed form solution by augmenting it to set up a binary Ising model, enabling scalable model fitting. (V) Majority Vote (MV): A standard scheme that uses the most popular LF output as the estimate of the true label.

Evaluation Metrics As common in related work, label model performance is compared based on the pseudolabel accuracy the models achieve on the training data, since programmatic weak supervision operates in a transductive setting. Weighted F1 and mean Average Precision are provided in Appendix C.5. To compare the quality of generated color images, the Fréchet Inception Distance (FID) is used, which has been shown to be consistent with human judgments and to be more robust than related measures [119], and which is used to measure performance current state-of-the-art GAN approaches [142]. To show the improvement in alignment of the auxiliary model Q's predictions of the discrete latent code b with the latent labels y, the Adjusted Rand Index (ARI) between the two during training is tracked.

Results

Results comparing label model and image generation performance are discussed first, before the use of WSGAN for augmentation of the downstream classifier with synthetic samples are presented. Each experiment is repeated at least three times and averaged results are reported in the tables.

Label Model Performance Table 3.5 shows a comparison of label model performance based on the accuracy of the posterior on the training data, without the use

Table 3.6: Color image generation quality measured by average Fréchet Inception Distance (FID), using DCGAN base architectures. The best scores for each dataset are highlighted in **blue**.

Dataset	InfoGAN	WSGAN-V	WSGAN-E
AwA2 - A	41.62	36.74	**34.71**
AwA2 - B	41.62	36.79	**34.52**
DomainNet	53.98	50.16	**44.35**
GTSRB	**69.67**	75.27	73.96
CIFAR10-A	28.93	25.70	**22.71**
CIFAR10-B	33.50	26.17	**24.41**

of any labeled data or validation sets. WSGAN-encoder largely outperforms alternative label models, while the simpler WSGAN-vector model performs competitively as well. These results hold according to additional evaluation metrics provided in Appendix C.3 . Results with standard deviations over 5 random runs are provided in the Appendix in Table C.6, indicating that many differences are significant.

Discrete Latent Code Comparison Figure 3.5 shows the evolving ARI between the ground truth and the auxiliary model Q's prediction of the latent code on real data during model training. The figures show a large improvement in Q's ability to uncover the unobserved class label structure when comparing WSGAN to InfoGAN, which is expected as WSGAN can take advantage of the weak signals encoded in LFs, while InfoGAN is completely unsupervised.

Image Generation Performance Table 3.6 compares FID of generated color images, suggesting that WSGAN models do take advantage of the weak supervision

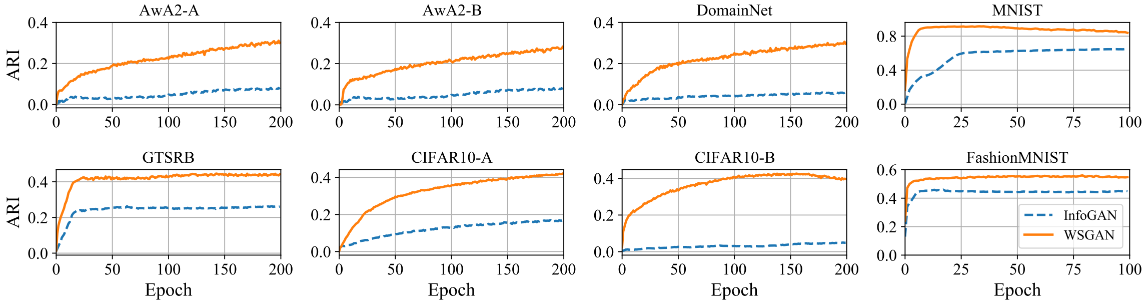


Figure 3.5: ARI between the unobserved class label y and the discrete code prediction by the auxiliary model $Q(x)$ on real image x, during training. Weak supervision allows WSGAN to better uncover the latent class structure compared to an unsupervised InfoGAN.

Table 3.7: Increase in test accuracy when augmenting downstream classifier training with 1,000 synthetic images and corresponding pseudolabels (PLs). Synthetic PLs are obtained via $F_1(Q(\tilde{x}))$, LF PLs via $L_{A(\tilde{x})}(\lambda(\tilde{x}))$.

Dataset	Synthetic PLs	LF PLs
AwA2 - A	0.88%	0.79%
AwA2 - B	2.40%	3.90%
DomainNet	2.31%	1.50%
MNIST	1.60%	1.71%
FashionMNIST	0.29%	0.34%
GTSRB	0.40%	0.02%
CIFAR10-A	0.04%	-
CIFAR10-B	0.30%	-

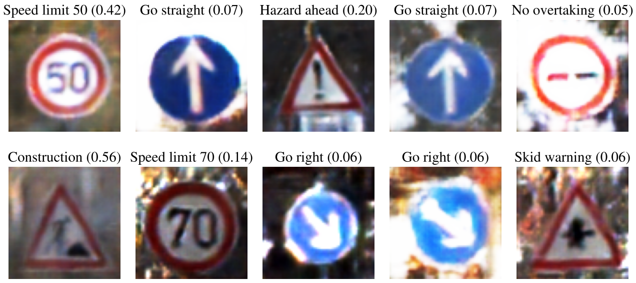


Figure 3.6: Images and pseudolabels generated by the proposed WSGAN (with a simple DCGAN architecture) on the weakly supervised GTSRB dataset. WSGAN can estimate labels even for images where no weak supervision sources provide information (see end of Section 3.2.2).

signal to improve Q, thereby improving the quality of generated images compared to an InfoGAN using the same base DCGAN architecture. WSGAN has a lower FID only on the GTSRB dataset, likely due to GTSRB's class imbalance and the dataset difficulty (43 classes, ¡23k samples).

Synthetic Data Augmentation The change in test accuracy for a ResNet-18 [117] end model is recorded when 1,000 synthetic WSGAN-encoder images $\tilde{x}$ are added to augment each dataset. While the increases are modest, the process is beneficial and does not require additional human labeling or data collection efforts. Adding larger amounts of synthetic samples did not lead to further increases, possibly due to the limited image quality achieved by the basic DCGAN design explored in this section.

Synthetic Images with Labeling Function Votes The last column in Table 3.7 displays test accuracy increases by applying LFs λ to synthetic images $\tilde{x}$. Pseudolabels are obtained via $L_{A(\tilde{x})}(\lambda(\tilde{x}))$. A modest average increase of 1.38% is observed.

Synthetic Images with Synthetic Pseudolabels Pseudolabels can also be created with $F_1(Q(\tilde{x}))$, e.g. when LFs cannot be applied to synthetic images. With this, the second column of Table 3.7 shows an average increase in test accuracy of 1%, and up to 2.4%. Larger increases in accuracy by adding more generated images are not observed. Figure 3.6 shows a small number of generated images along with synthetic pseudolabel estimates. While $F_1(Q(x))$ could conceivably be used as a downstream classifier, the choices of network architecture are constrained as it shares convolutional layers with D.

Synthetic Data Quality Checks In addition to visually inspecting some generated samples and checking if conditionally generated samples reflect the target labels, the class balance in the pseudolabels of synthetic images should be checked before adding them to a downstream training set, as mode collapse in a trained GAN can potentially be diagnosed this way.

Network Ablation — StyleWSGAN

Here, StyleWSGAN is applied to weakly supervised LSUN scene categories [283], and to the CIFAR10-B dataset. The results demonstrate WSGAN's complementarity with other GAN architectures, and that the approach scales to higher resolution images. Please see Section C.1 for implementation details and hyperparameter settings that were used for the StyleGAN experiments. Dataset statistics for the two additional datasets that were created for this ablation are shown in Table 3.8.

LSUN scene categories To test the proposed WSGAN on higher resolution images with a StyleGAN base architecture, I create a balanced subset of the LSUN scene categories dataset [283]. The dataset contains 10 classes (i.e. 10 different scene categories) and images are center-cropped and resized to 256 by 256 pixels. An fixed number of images is sampled from each of the 10 classes for a final dataset size of 1,212,270 images. As weak supervision sources, I create 30 SSL-based LFs by training classifiers on small amounts of held-out data using image features learned via self-supervised learning, as described in Section C.2.1.

StyleWSGAN achieves an average FID of 7.54 on this weakly supervised LSUN scene category dataset. Some generated samples are visualized in Fig. 3.7. An unconditional StyleGAN2-ADA achieves an FID of 10.3 with the settings for 256 by 256 images set in [142], and an FID of 8.41 when these settings are changed to avoid path length regularization and style mixing. Note that unconditional StyleGAN results on LSUN images with lower FID scores reported in related work are generally obtained

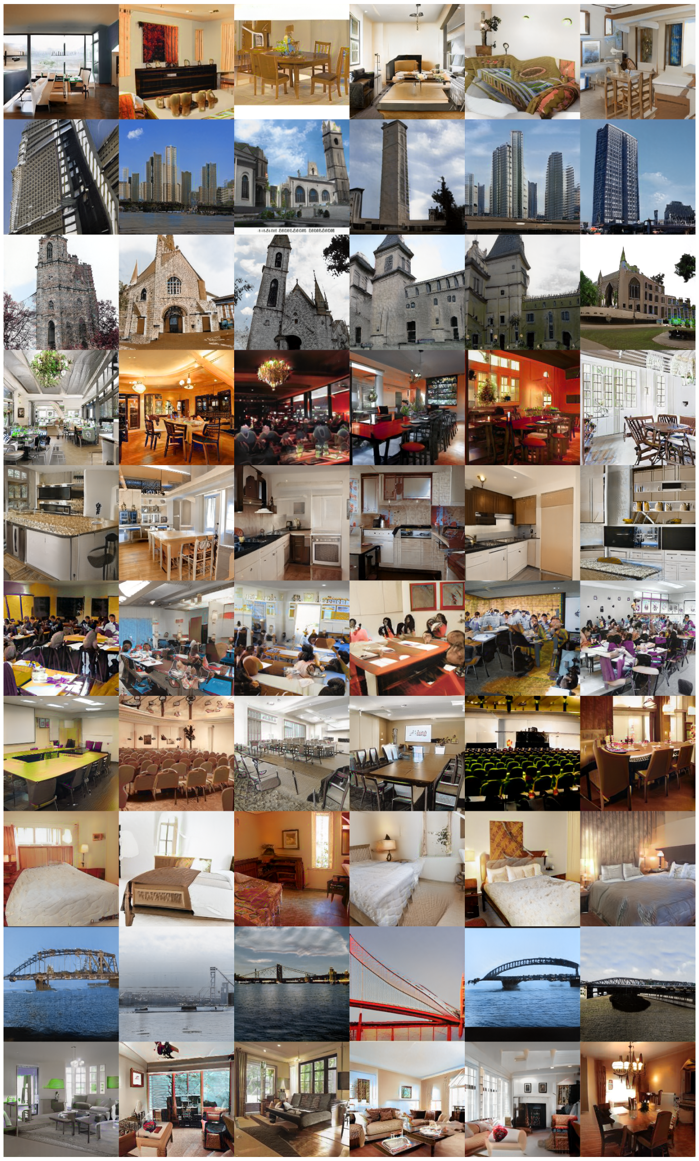


Figure 3.7: Synthetic images learned by StyleWSGAN on a weakly supervised subset of the LSUN scene category dataset.

by training on a single LSUN scene or object category, rather than on multiple categories simultaneously, as in the experiments of this section, which results in a more challenging setup.

CIFAR10 First, Figure 3.8 shows synthetic images by StyleWSGAN on the weakly supervised CIFAR10-B dataset, which uses SSL-based LFs. These LFs are quite noisy, with a mean LF accuracy of 0.736, which is reflected in the noisy class-conditional samples that can be inspected in Figure 3.8. On this dataset, StyleWSGAN achieves

Table 3.8: Additional datasets used to evaluate StyleWSGAN. Acc denotes accuracy, while Coverage denotes the number of samples where an LF does not abstain.

Dataset	#Classes	#LFs	#Samples	Mean LF Acc	Min LF Acc	Max LF Acc	Mean Coverage	LF Type
CIFAR10 - low noise LFs	10	20	48,000	0.888	0.816	0.949	0.102	Synthetic
LSUN scene categories	10	30	1,212,270	0.736	0.624	0.873	0.098	SSL-based

a mean FID of 3.79 (generated images are shown in Fig. 3.8), while also attaining a high label model accuracy of 0.736 (compare this accuracy with the label model results shown in Table 3.5). The unsupervised StyleGAN2-ADA, with the optimal, tuned settings identified in [142], achieves an average FID of 3.85 on this subset. An unsupervised StyleInfoGAN that I created arrived at a mean FID of 4.13.

An additional weakly supervised CIFAR10 with lower noise LFs is created to check if such a setting can lead to results that are better than the state-of-the-art (SOTA) unsupervised image generation quality on the full CIFAR10 dataset reported in [142]. For this experiment, I create LFs by randomly introducing errors and abstains to the ground-truth vector. For each of these LFs, I set a minimum accuracy of 0.8 and a maximum accuracy of 0.95 and create 20 LFs. This dataset contains 48000 samples, has a mean LF accuracy of 0.888, and a mean coverage of 0.102 (meaning that an LF on average abstains on $\sim$ 89.8% of the dataset). For this dataset, StyleWSGAN achieves an FID of 2.84, which is better than the SOTA unsupervised result reported in [142] of 2.92 FID on the full 50k CIFAR10 samples, but shy of the performance of the conditional StyleGAN [142] which uses projection discrimination and has access to all ground-truth labels and achieves and FID of 2.42.

3.2.4 Discussion

This section studied the question of how to build an interface between two powerful techniques that operate in the absence of labeled data: generative modeling and programmatic weak supervision. The proposed fusion of the two, a weakly supervised GAN (WSGAN), defines an interface that aligns structures discovered in its constituent models. This leads to three improvements: first, better quality pseudolabels compared to weak supervision alone, boosting downstream performance. Second, improvement in the quality of the generative model samples. Third, enabling data augmentation using such samples, further improving downstream model performance without additional burden on users.

Standard failure cases of GANs such as mode collapse still apply to the proposed approach. However, the experiments conducted here did not indicate that WSGAN is more susceptible to such failures than the approaches it was compared to. The proposed approach leads to several exciting directions for future work, for example the use of modalities other than images, exploiting for instance generative models for graphs and time series. Further, motivated by the performance of WSGAN, the underlying notion of interfaces between models to a variety of other pairs of models

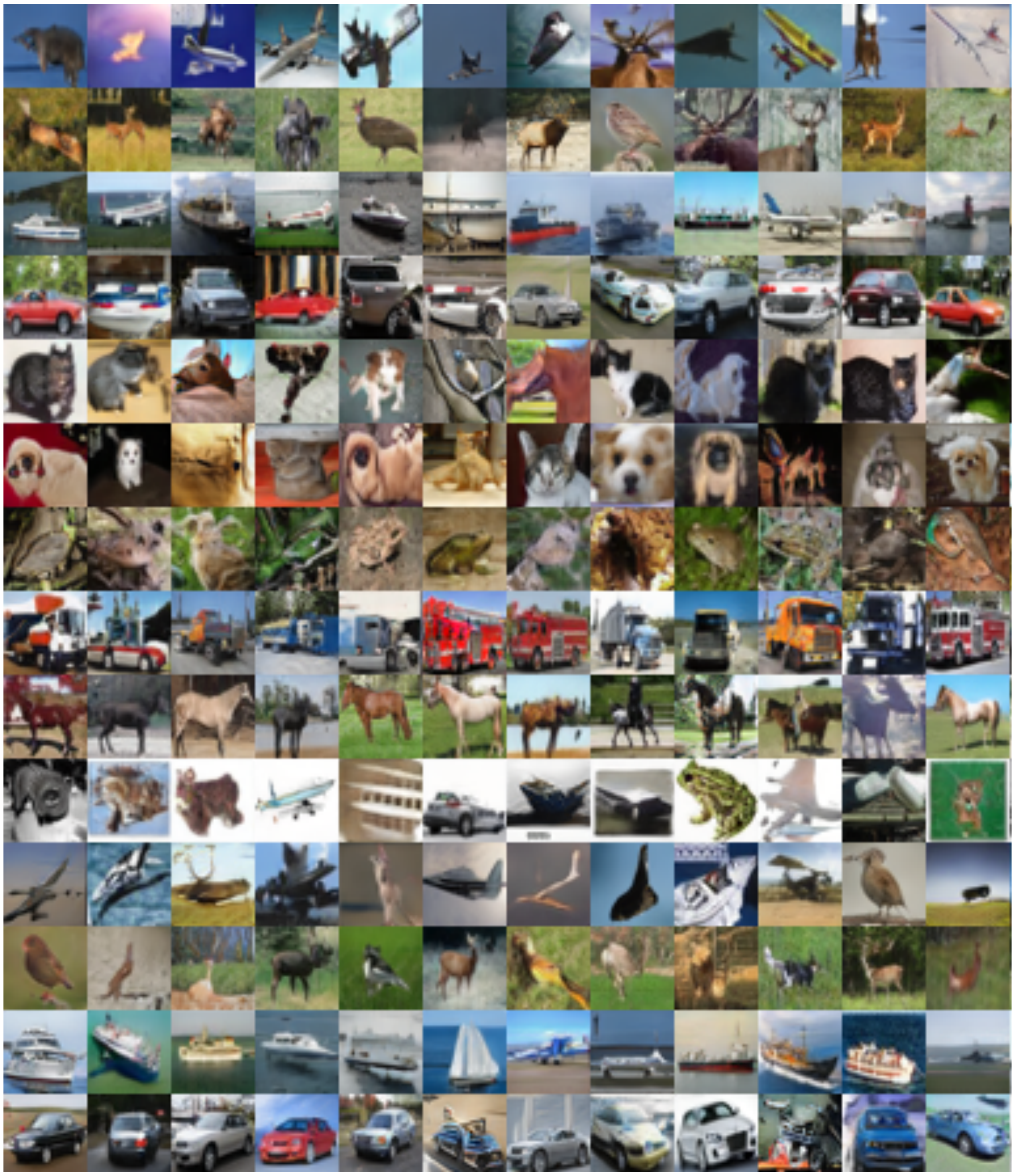


Figure 3.8: Synthetic images learned on the CIFAR10-B subset by StyleWSGAN, which a version of the proposed WSGAN built on StyleGAN2-ADA rather than a simple DCGAN as in the main experiments of this work.

is desirable. Limitations of the proposed approach include common GAN restrictions such as the types of data that can be modeled as well as difficulties in finding the right parameter settings to enable stable training, and furthermore known difficulties of acquiring weak supervision sources of sufficient quality for image data.

Chapter 4

Interactivity and Multi-Modal Learning

Obtaining and structuring domain knowledge in forms that can be consumed by weak supervision learning paradigms is not always straightforward, and popular frameworks such as DP still require considerable user effort at this–frequently obscure–stage of the application pipeline. In this Chapter, I study two extremes on the spectrum of user involvement in order to efficiently harvest domain knowledge. First, I present an interactive method for aiding users in discovering useful labeling functions. The goal of this interactive learning framework is to systematically capture subject matter experts' knowledge of an application domain in an efficient and effective fashion, by letting the experts adjudicate Labeling Function (LF) candidates. As a contrast to this user involvement, in the last Section I study a weak supervision setting in which unstructured natural language descriptions accompanying image data are used to train good encoders for downstream tasks, without users defining rules on top of the text.

4.1 Interactive Weak Supervision

> This Section is based on the work presented in:
> Boecking, Benedikt, Willie Neiswanger, Eric Xing, and Artur Dubrawski. "Interactive Weak Supervision: Learning Useful Heuristics for Data Labeling". In: *International Conference on Learning Representations (ICLR)*. 2021

In this part of the thesis, I develop the first framework for interactive weak supervision in which a method proposes heuristics and learns from user feedback given on each proposed heuristic. The experiments demonstrate that only a small number of feedback iterations are needed to train models that achieve highly competitive test set performance without access to ground truth training labels. I conduct user studies,

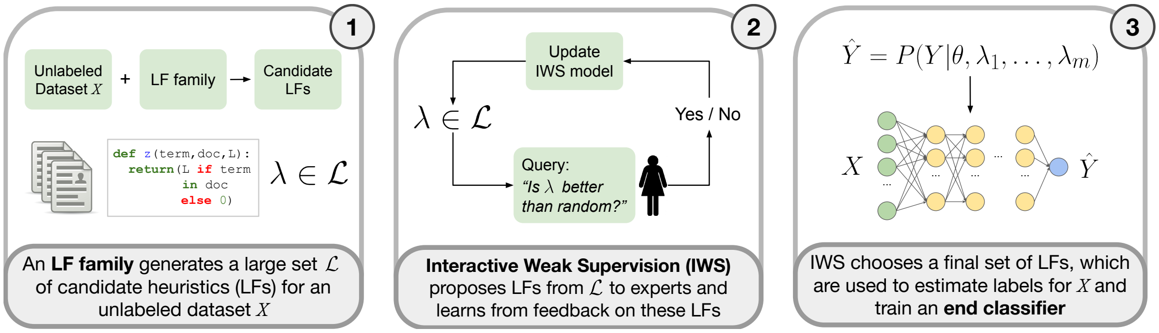

1
Unlabeled Dataset X
+
LF family
Candidate LFs
def z(term,doc,L):
return(L if term
in doc
else 0)
$\lambda \in \mathcal{L}$
An **LF family** generates a large set $\mathcal{L}$ of candidate heuristics (LFs) for an unlabeled dataset X
2
Update IWS model
$\lambda \in \mathcal{L}$
Yes / No
Query: "Is λ better than random?"
Interactive Weak Supervision (IWS) proposes LFs from $\mathcal{L}$ to experts and learns from feedback on these LFs
3
$\hat{Y} = P(Y|\theta, \lambda_1, \ldots, \lambda_m)$
X
$\hat{Y}$
IWS chooses a final set of LFs, which are used to estimate labels for X and train an **end classifier**

Figure 4.1: Interactive Weak Supervision (IWS) helps experts discover good labeling functions (LFs).

which show that users are able to effectively provide feedback on heuristics and that test set results track the performance of simulated oracles.

In data programming, each LF is an imperfect but reasonably accurate heuristic, such as a pre-trained classifier or keyword lookup. For example, for the popular *20 newsgroups* dataset, an LF to identify the class '*sci.space*' may look for the token '*launch*' in documents and would be correct about 70% of the time. While data programming can be very effective when done right, experts may spend a significant amount of time designing the weak supervision sources [254] and must often inspect samples at random to generate ideas [55]. In the *20 newsgroups* example, we may randomly see a document mentioning '*Salman Rushdie*' and realize that the name of a famous atheist could be a good heuristic to identify posts in '*alt.atheism*'. While such a heuristic seems obvious after the fact, we have to chance upon the right documents to generate these ideas. In practice, coming up with effective LFs becomes difficult after the first few. Substantial foresight [218] is required to create a new function that applies to a non-negligible subset of given data, is novel, and adds predictive value.

I propose a new approach termed *Interactive Weak Supervision (IWS)* for training supervised ML models with weak supervision through an interactive process, supporting domain experts in fast discovery of good LFs. The method queries users in an active fashion for feedback about candidate LFs, from which a model learns to identify LFs likely to have a good accuracy. This enables IWS to recommend LFs with desired accuracy and coverage trade-offs. Upon completion, the approach produces a final set of LFs. This set is used to create an estimate of the latent class label via an unsupervised label model and train a final, weakly supervised end classifier using a noise aware loss function on the estimated labels as in [222]. The approach relies on the observation that many applications allow for heuristics of varying quality to be generated at scale (similar to [254]), and that experts can provide good judgment by identifying some LFs that have reasonable accuracy. The full pipeline of the proposed IWS approach[1], is illustrated in Fig. 4.1. The contributions are:

[1] Code is available at `https://github.com/benbo/interactive-weak-supervision`

1. The first interactive method for weak supervision in which queries to be annotated are not data points but labeling functions. This approach automates the discovery of useful data labeling heuristics.

2. Experiments with real users on three classification tasks, using both text and image datasets. Results support the modeling assumptions, demonstrate competitive test set performance of the downstream end classifier, and show that users can provide accurate feedback on automatically generated LFs.

3. IWS shows superior performance compared to standard active learning, i.e. it achieves better test set performance with a smaller number of queries to users. In text experiments with real users, IWS achieves a mean test set AUC after 200 LF annotations that requires at least three times as many active learning iterations annotating data points. In addition, the average user response time for LF queries was shorter than for the active learning queries on data points.

4.1.1 Methodology

I propose an interactive weak supervision (IWS) approach to assist experts in finding good labeling functions (LFs) for training a classifier on datasets without ground truth labels. I will first describe the general problem setting of learning to classify without ground truth samples by modeling multiple weak supervision sources, as well as the concept of LF families. I then dive into the details of the proposed IWS approach. For brevity, I limit the scope of the end classifier to binary classification, but the presented background and ideas do extend to the multi-class settings.

Preliminaries

Learning with Multiple Weak Supervision Sources Assume each data point $x \in \mathcal{X}$ has a latent class label $y^* \in \mathcal{Y} = \{-1, 1\}$. Given n unlabeled, i.i.d. datapoints $X = \{x_i\}_{i=1}^n$, the goal is to train an end classifier $f : \mathcal{X} \to \mathcal{Y}$ such that $f(x) = y^*$. In data programming [222, 220], a user provides m LFs $\{\lambda_j\}_{j=1}^m$, where $\lambda_j : \mathcal{X} \to \mathcal{Y} \cup \{0\}$. An LF λ_j noisily labels the data with $\lambda_j(x) \in \mathcal{Y}$ or abstains with $\lambda_j(x) = 0$. The corresponding LF output matrix is $\Lambda \in \{-1, 0, 1\}^{n \times m}$, where $\Lambda_{i,j} = \lambda_j(x_i)$. In this work, it is assumed that each LF λ_j has the same accuracy for each class, $\alpha_j = P(\lambda_j(x) = y^* | \lambda_j(x) \neq 0)$, where accuracy is defined on items where j does not abstain. Further, we denote by $l_j = P(\lambda_j(x) \neq 0)$ the LF propensity (sometimes called LF coverage), i.e. the frequency at which LF j does not abstain.

In data programming, an unsupervised label model $p_\theta(Y, \Lambda)$ produces probabilistic estimates of the latent class labels $Y^* = \{y_i^*\}_{i=1}^n$ using the observed LF outputs Λ by modeling the LF accuracies, propensities, and possibly their dependencies. A number of label model approaches exist in the crowd-sourcing [72, 290] and the weak supervision literature [220]. In this paper, a factor graph is used to obtain probabilistic labels, as proposed in [222, 220]. The factor graph models the LF accuracies via factor $\phi_{i,j}^{Acc}(\Lambda, Y) \triangleq \mathbb{1}\{\Lambda_{ij} = y_i\}$ and labeling propensity by factor $\phi_{i,j}^{Lab}(\Lambda, Y) \triangleq \mathbb{1}\{\Lambda_{ij} \neq 0\}$, and for simplicity assumes LFs are independent conditional on Y. The label model is defined as

$$p_\theta(Y, \Lambda) \triangleq Z_\theta^{-1} \exp\left(\sum_{i=1}^{n} \theta^\top \phi_i(\Lambda_i, y_i)\right), \tag{4.1}$$

where Z_θ is a normalizing constant and $\phi_i(\Lambda_i, y_i)$ defined to be the concatenation of the factors for all LFs $j = 1, \ldots, m$ for sample i. θ is learned by minimizing the negative log marginal likelihood given the observed Λ. Finally, following [222] an end classifier f is trained using probabilistic labels $p_\theta(Y|\Lambda)$.

Labeling Function Families We will define LF families as sets of LFs that are interpretable by experts, described by functions $z_\phi : \mathcal{X} \mapsto \{-1, 0, 1\}$, for parameters $\phi \in \Phi$. An example are shallow decision trees z_ϕ parameterized by variables and splitting rules ϕ [254], or a function z_ϕ defining a regular expression for two words where ϕ parameterizes the word choices from a vocabulary and the target label. Given such an LF family, one can generate a large set of p candidate heuristics $\mathcal{L} = \{\lambda_j(x) = z_{\phi_j}(x)\}_{j=1}^{p}$, where $\phi_j \in \Phi$, e.g. by sampling from Φ and pruning low coverage candidates. These families often arise naturally in the form of LFs with repetitive structure that experts write from scratch, where template variables—such as keywords—can be sampled from the unlabeled data to create candidates. For text, one can find n-grams within a document frequency range to generate key term lookups, fill placeholders in regular expressions, or generate shallow decision trees [222, 254, 255]. For time series, one can create a large set of LFs based on motifs [167] or graphs of temporal constraints [109]. For images, one can create a library of pre-trained object detectors as in [46], or in some applications combine primitives of geometric properties of the images [254].

An LF family has to be chosen with domain expert input. Compared to standard data programming, the burden of creating LFs from scratch is shifted to choosing an appropriate LF family and then judging recommended candidates. I argue that domain experts often have the foresight to choose an LF family such that a sufficiently sized subset of LFs is predictive of the latent class label. Such LF families may not exist for all data types and classification tasks. But when they exist they offer the opportunity to quickly build large, labeled datasets. Once created, it is reasonable to expect that the same LF generation procedure can be reused for similar classification tasks without additional effort (e.g. a single LF family procedure is used for all text datasets in the experiments in the following Section).

Interactive Weak Supervision

Instead of having users provide m good weak supervision sources up front, this work aims to assist users in discovering them. Successful applications of data programming have established that human experts are able to construct accurate LFs from scratch. This work leverages the assumption that human experts can also judge these properties when presented with pre-generated LFs of the same form.

Suppose again that we have an unlabeled dataset $X = \{x_i\}_{i=1}^n$, and that the goal is to train an **end classifier** f without access to labels $Y^* = \{y_i^*\}_{i=1}^n$. Assume also that we defined a large pool of p candidate LFs $\mathcal{L} = \{\lambda_j(x)\}_{j=1}^p$ from an LF family (following Sec. 4.1.1), of varying accuracy and coverage. In IWS, the goal is to identify an optimal subset of LFs $\mathcal{L}^* \subset \mathcal{L}$ to pass to the **label model** in Eq. (4.1). Below, I will quantify how $\mathcal{L}^*$ depends on certain properties of LFs. While one can observe some of these properties—such as coverage, agreement, and conflicts—an important property that cannot be observed is the accuracy of each LF.

The goal will thus be to infer quantities related to the latent accuracies $\alpha_j \in [0, 1]$ of LFs $\lambda_j \in \mathcal{L}$, given a small amount expert feedback. To do this, I define an **expert-feedback model**, which can be used to infer LF accuracies given a set of user feedback. To efficiently train this model, the IWS procedure sequentially chooses an LF $\lambda_j \in \mathcal{L}$ and shows a description of λ_j to an expert, who provides binary feedback about λ_j. This work follows ideas from active learning for sequential decision making under uncertainty, in which a probabilistic model guides data collection to efficiently infer quantities of interest within T iterations. After a sequence of feedback iterations, the expert-feedback model is used to provide an estimate $\hat{\mathcal{L}} \subset \mathcal{L}$ of the optimal subset $\mathcal{L}^*$. The label model then uses $\hat{\mathcal{L}}$ to produce a probabilistic estimate of Y^*, which is used to train the end classifier f. The full IWS procedure is illustrated in Fig. 4.1 and described in detail below.

Expert-Feedback Model We will first define a generative model of human expert feedback about LFs, given the latent LF accuracies. This model will form the basis for an online procedure that selects a sequence of LFs to show to human experts. We will task experts to classify LFs as either useful or not useful $u_j \in \{0, 1\}$, corresponding to their *belief that LF λ_j is predictive of Y^* at better than random accuracy for the samples where λ_j does not abstain.* Note that prior data programming work [222, 221, 85, 231] assumes and demonstrates that experts are able to use their domain knowledge to make this judgment when creating LFs from scratch. The generative process for this feedback and the latent LF accuracies is modeled as, for $j = 1, \dots, t$:

$$u_j \sim \text{Bernoulli}(v_j), \;\; v_j = h_\omega(\lambda_j), \;\; \omega \sim \text{Prior}(\cdot) \tag{4.2}$$

where v_j can be viewed as the average probability that a human will label a given LF λ_j as $u_j = 1$, and $h_\omega(\lambda_j)$ is a parameterized function (such as a neural network), mapping each LF λ_j to v_j. Finally, to model the connection between accuracy α_j

and v_j, we assume that $v_j = g(\alpha_j)$, where $g : [0,1] \to [0,1]$ is a monotonic increasing function mapping unknown LF accuracy α_j to v_j.

After t queries of user feedback on LFs, we have produced a query dataset $Q_t = \{(\lambda_j, u_j)\}_{j=1}^{t}$. Given Q_t, unknown quantities in the above model–which are used to choose the next LF λ_j to query–are inferred by constructing an acquisition function $\varphi_t : \mathcal{L} \to \mathbb{R}$ and optimizing it over $\lambda \in \mathcal{L}$.

Acquisition Strategy and Final Set of LFs To derive an online procedure for the user queries about LFs, we need to define the properties of the ideal subset of generated LFs $\mathcal{L}^* \subset \mathcal{L}$ which we want to select. Prior data programming work of [222, 221, 220] with label models as in Eq. (4.1) does not provide an explicit analysis of ideal metrics of LF sets and their trade-offs to help define this set. This work provides the following theorem, which will motivate the definition for $\mathcal{L}^*$.

Theorem 4.1.1. *Assume a binary classification setting, m independent labeling functions with accuracy $\alpha_j \in [0,1]$ and labeling propensity $l_j \in [0,1]$. For a label model as in Eq. (4.1) with given label model parameters $\hat{\theta} \in \mathbb{R}^{2m}$, and for any $i \in \{1, \dots, n\}$,*

$$P(\hat{y}_i = y_i^*) \geq 1 - \exp\left(- \frac{(\sum_{j=1}^{m} \hat{\theta}_j^{(1)} (2\alpha_j - 1) l_j)^2}{2||\hat{\theta}^{(1)}||^2} \right)$$

where $\hat{\theta}^{(1)}$ are the m weights of ϕ^{Acc}, and $\hat{y}_i \in \{-1, 1\}$ is the label model estimate for y_i^*.

Proof. The proof is given in Appendix D.1. □

This theorem indicates that one can rank LFs according to $(2\alpha_j - 1)\hat{l}_j$ where $\alpha_j, \hat{l}_j$ are the unknown accuracy and observed coverage of LF j, respectively. Additional analysis is provided in Appendix D.1. The analysis further suggests the importance of obtaining LFs with an accuracy gap above chance. Intuitively, we do not want to add excessive noise by including LFs too close to random. Below, let us assume that the final set of LFs is sufficient to accurately learn label model parameters $\hat{\theta}$, and leave analysis of the influence of additional LF properties on learning $\hat{\theta}$ to future work.

To define the ideal final subset of LFs, three scenarios are distinguished: (A) there are no restrictions on the size of the final set and any LF can be included, (B) the final set is limited in size (e.g. due to computational considerations) but any LF can be included, (C) only LFs inspected and validated by experts may be included, e.g. due to security or legal considerations.

For each of these scenarios, at each step t we will maximize an acquisition function over the set of candidate LFs, i.e. compute $\lambda_t = \text{argmax}_{\lambda \in \mathcal{L} \setminus Q_{t-1}} \varphi_t(\lambda)$. We then query a human expert to obtain (λ_t, u_t) and update the query dataset $Q_t = Q_{t-1} \cup \{(\lambda_t, u_t)\}$. After a sequence of T queries we return an estimate of $\mathcal{L}^*$, denoted by $\hat{\mathcal{L}}$. The corresponding LF output matrix Λ comprised of all $\lambda_j \in \hat{\mathcal{L}}$, is then used to produce an estimate $\widehat{Y}$ of the true class labels via the label model $P_\theta(Y|\Lambda)$. Finally, a noise-aware discriminative end classifier f is trained on $(X, \widehat{Y})$.

Scenario (A): Unbounded LF Set In the absence of restrictions on the final set of LFs, the analysis in Appendix D.1 indicates that the ideal subset of LFs $\mathcal{L}^*$ includes all those with accuracy greater than a gap above chance, i.e. $\alpha_j > r > 0.5$. Thus, let us define the optimal subset in this scenario as

$$\mathcal{L}^* = \{\lambda_j \in \mathcal{L} \, : \, \alpha_j > r\} \, . \tag{4.3}$$

This is a variation of the task of active Level Set Estimation (LSE), where the goal is to identify all elements in a superlevel set of $\mathcal{L}$ [287, 103, 32]. Thus, at each step t we will use the straddle acquisition function [32] for LSE, defined for a candidate $\lambda_j \in \mathcal{L} \backslash Q_{t-1}$ to score LFs highest that are unknown and near the boundary threshold r:

$$\varphi_t^{\text{LSE}}(\lambda_j) = 1.96 \, \sigma_j(Q_{t-1}) - |\mu_j(Q_{t-1}) - r| \tag{4.4}$$

where

$$\sigma_j(Q_{t-1}) = \sqrt{\text{Var}[p(\alpha_j|Q_{t-1})]}$$

is the standard deviation and

$$\mu_j(Q_{t-1}) = \mathbb{E}[p(\alpha_j|Q_{t-1})]$$

the mean of the posterior LF accuracy. The end of Section 4.1.1 describes how to perform approximate inference of $p(\alpha_j|Q_{t-1})$ via an ensemble model. After a sequence of T queries IWS returns the following estimate of $\mathcal{L}^*$:

$$\hat{\mathcal{L}} = \{\lambda_j \in \mathcal{L} \, : \, \mu_j(Q_T) > r\} \, . \tag{4.5}$$

Let us denote the algorithm for scenario (A) by **IWS-LSE-a**. See Algorithm 3 for pseudocode describing this full IWS-LSE-a procedure. In the experiments following this section $r = 0.7$, though an ablation study shows that IWS-LSE works well for a range of thresholds $r > 0.5$ (Appendix 4.1.2, Figure 4.6). Note that the LSE acquisition function aims to reduce uncertainty around r, and therefore tends to explore LFs that cover on parts of Y that the model is still uncertain about.

Scenario (B): Bounded LF Set If the final set is restricted in size to m LFs, e.g. due to computational considerations when learning the label model in Eq. (4.1), one needs to take the trade-off of LF accuracy and LF coverage into account. Let $\hat{l}_j$ be the observed empirical coverage of LF λ_j. We want to identify LFs with accuracy above

r and rank them according to their accuracy-coverage trade-off, thus the analysis in the appendix suggests the optimal subset is

$$\mathcal{L}^* = \operatorname*{argmax}_{\mathcal{D}\subseteq\mathcal{L},|\mathcal{D}|=m} \sum_{\lambda_j\in\mathcal{D}} \left(\mathbb{1}_{\{\alpha_j>r\}}(2*\alpha_j-1)*\hat{l}_j\right). \tag{4.6}$$

Since the LF accuracy-coverage trade-off only comes into effect if $\alpha_j > r$, this yields the same acquisition function φ_t^{LSE} in Eq. (4.4), and the final set is then selected as

$$\hat{\mathcal{L}} = \{\lambda_j \in \mathcal{D} \,:\, \operatorname*{argmax}_{\mathcal{D}\subseteq\mathcal{L},|\mathcal{D}|=m} \sum_{\lambda_j\in\mathcal{D}} (\mathbb{1}_{\{\mu_j(Q_T)>r\}}(2*\mu_j(Q_T)-1)*\hat{l}_j)\},$$

which corresponds to a simple thresholding and sorting operation. We will denote the algorithm for scenario (B) by **IWS-LSE-ac**.

Scenario (C): Validated LF Set Finally, in some application scenarios, only LFs inspected and validated by experts should be used to estimate Y^*, e.g. due to security or legal considerations. An LF j is validated if it is shown to an expert who then responds with $u_j = 1$. This leads to an active search problem [96] where the aim is to identify a maximum number of validated LFs (i.e. $u = 1$) in $\mathcal{L}$ given a budget of T user queries, i.e. to compute

$$\mathcal{L}^*_{\text{AS}} = \operatorname*{argmax}_{\mathcal{D}\subset\mathcal{L},|\mathcal{D}|=T} \sum_{\lambda_j\in\mathcal{D}} u_j, \qquad \hat{\mathcal{L}} = \{\lambda_j \in Q_T \,:\, u_j = 1\}. \tag{4.7}$$

As in [96, 135], we will use a one-step look ahead active search acquisition function defined for a candidate $\lambda_j \in \mathcal{L}\backslash Q_{t-1}$ to be the posterior probability that the usefulness label u_j is positive, i.e. $\varphi_t^{\text{AS}}(\lambda_j) = \mu_j(Q_{t-1})$. The algorithm for scenario (C) is denoted by **IWS-AS**.

Algorithm 3: Interactive Weak Supervision (IWS-LSE-a).

```
Input: $\mathcal{L}$: set of LFs, $T$: max iterations.
$Q_0 \leftarrow \varnothing$
for $t = 1, 2, \ldots, T$ do
    $\lambda_t \leftarrow \operatorname{argmax}_{\lambda\in\mathcal{L}\backslash Q_{t-1}} \varphi_t(\lambda)$        ▷ Eq. (4.4)
    $u_t \leftarrow \mathit{ExpertQuery}(\lambda_t)$
    $Q_t \leftarrow Q_{t-1} \cup \{(\lambda_t, u_t)\}$
end
$\hat{\mathcal{L}} \leftarrow \{\lambda_j \in \mathcal{L} \,:\, \mathbb{E}[p(\alpha_j|Q_T)] > r\}$        ▷ Eq. (4.5)
```

Approximate Inference Details I will now describe how to use the expert-feedback model in Eq. (4.2) to infer $p(\alpha_j|Q_t)$, a quantity used in the acquisition functions and final set estimates. Recall that we defined a generative model of human feedback u_j on query LF λ_j with latent variables v_j and ω. We assumed a connection between v_j and the latent LF accuracy α_j via a monotonic increasing function $\alpha_j = g(v_j)$. Similar to existing work on high dimensional uncertainty estimation [19, 53], we can use an ensemble $\{\tilde{h}_{\omega^{(i)}}\}_{i=1}^{s}$ of s neural networks $\tilde{h}_\omega$ with parameters ω to predict u_j given input λ_j. To perform this prediction, we need a feature representation $\tau(\lambda_j)$ for LFs that is general and works for any data type and task. To create these features, we will use the LF output over the unlabeled dataset $\tau_0(\lambda_j) = (\lambda_j(x_1), \ldots, \lambda_j(x_n))$. We will then project $\tau_0(\lambda_j)$ to d' dimensions using PCA for a final feature representation $\tau(\lambda_j)$, which is given as input to each $\tilde{h}_\omega$. The neural network ensemble can now learn functions $\tilde{h} : \mathbb{R}^{d'} \to [0, 1]$, which map from LF features $\tau(\lambda_j)$ to $v_j = p(u_j = 1|Q_t)$. This yields an ensemble of estimates for v_j, and through g^{-1}, of α_j. These are treated as approximate samples from $p(\alpha_j|Q_t)$, and used to form sample-estimates used in the acquisition functions.

4.1.2 Experiments

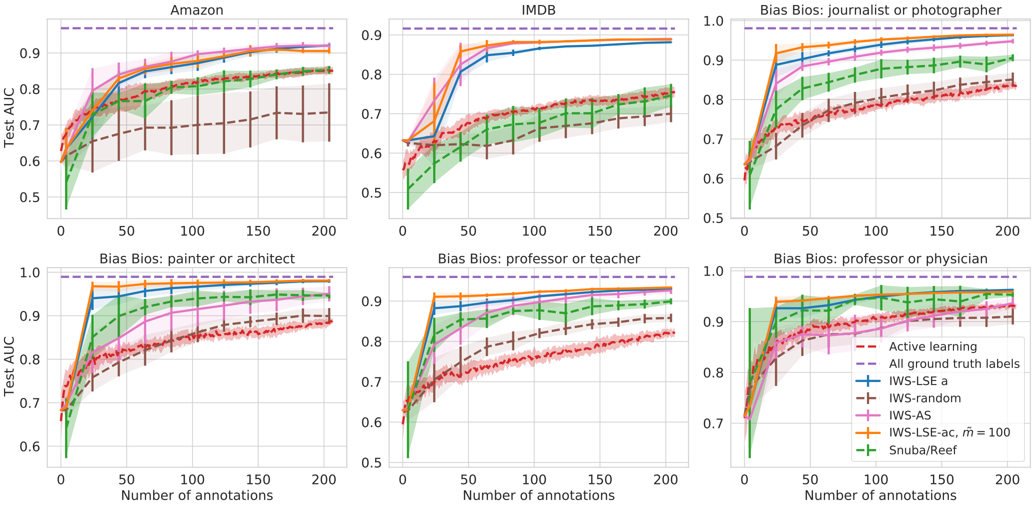


Figure 4.2: Mean test set AUC vs. number of iterations for end classifiers trained on probabilistic labels. IWS-LSE and IWS-AS are compared to active learning, Snuba, training on all labels, and IWS with a random acquisition function. Note that, while one iteration on this corresponds to one expert label, a comparison of effort needed to answer each type of query (label for sample vs label for LF) will vary by application.

The experiments in this Section show that heuristics obtained via a small number of iterations of IWS can be used to train a downstream end classifier f with highly competitive test set performance. I first present results obtained with a simulated

IWS oracle instead of human users. Oracle experiments allow us to answer how the method would perform if users had perfect knowledge about LF accuracies. I then show results from a user study on text data in which the query feedback is given by humans. In Section 4.1.2 I provide results of a user study on images, using image based LFs.

Datasets Six binary text classification tasks are created on the basis of publicly available datasets[2]. The tasks are chosen such that most English speakers can provide sensible expert feedback on LFs, for ease of reproducibility. I use a subset of the Amazon Review Data [118] for sentiment classification, aggregating all categories with more than $100k$ reviews from which $200k$ reviews are sampled and split into $160k$ training points and $40k$ test points. The IMDB Movie Review Sentiment dataset [182] is also used. It has $25k$ training samples and $25k$ test samples. In addition, I use the Bias in Bios [8] dataset from which I create binary classification tasks to distinguish difficult pairs among frequently occurring occupations. Specifically, I create the following subsets with equally sized train and test sets: journalist or photographer ($n = 32\,258$), professor or teacher ($n = 24\,588$), painter or architect ($n = 12\,236$), professor or physician ($n = 54\,476$).

For the cross-modal tasks of text captions and images as well as the pure image task the COCO dataset [168] is used. The official validation set ($n = 4952$) is used as the test set. This set of test images is only used to compute evaluation metrics, and is never accessed at any other point in the pipeline.

Cross-modal classification: As in [254], using the COCO dataset [168] LFs are generated over captions, while classification is performed on the associated images. The two binary tasks are to identify a 'person' in an image, and to identify 'sports' in an image.

Image classification: For image classification tasks with image LFs, I use the COCO dataset and create two binary classification tasks to identify 'sports' in an image and 'vehicle' in an image. For these image-only experiments, nearest-neighbor based LFs are created using feature representatinos of the images.

Approaches All approaches train the same downstream end classifier f on the same inputs X. Results are provided for *IWS-LSE-a* (unbounded LF set), *IWS-LSE-ac* (bounded LF set), and *IWS-AS* (validated LF set). For *IWS-LSE-ac*, the size of the final set of LFs at each iteration t is bound by $m = \sum_{i=1}^{t-1} u_i + \tilde{m}$, i.e. the number of LFs so far annotated as $u = 1$ plus a constant $\tilde{m}$. The test set performance of IWS is compared to a set of alternatives including (1) annotation of samples via *active learning* (uncertainty sampling) by a noiseless oracle, (2) the *Snuba* system [254], and (3) using *all ground truth training labels.* Additionally, the performance of IWS with a random acquisition function (*IWS-random*) is evaluated.

[2] Amazon: `https://nijianmo.github.io/amazon/index.html`, IMDB: `https://ai.stanford.edu/~amaas/data/sentiment/`, BiasBios: `http://aka.ms/biasbios`

In the figures, annotations on the x-axis correspond to labeled samples for Snuba and active learning, and to labeled LFs for IWS. This head to head comparison of user effort is naturally application dependent. For a comparison of effort in these specific experiments, I provide a timing study descibing the time required to carry out LF labeling versus labeling of samples, see Table 4.1.

LF Families For text tasks, prior work such as [220] and [255] demonstrates that word and phrase LFs can provide good weak supervision sources. To generate LFs, I define a uni-gram vocabulary over all documents and discard high and low frequency terms. I then exhaustively generate LFs from an LF family z_ϕ which outputs a target label if a uni-gram appears in a document, where ϕ specifies the uni-gram and target label. I also evaluated combinations of higher-order n-grams, but did not observe a significant change in performance. For COCO images, it is difficult to obtain strong domain primitives to create weak supervision sources, even for data programming from scratch. To generate LFs with high coverage, I first create small, unique clusters of up to k_1 mutual nearest neighbors (MkNN)[3]. For each member of a cluster, I find the k_2 nearest neighbors, and keep ones shared by at least one other cluster member. Finally, each extended cluster defines an LF, which assigns the same label to each member of the extended cluster. The MkNN symmetry produces good initial clusters of varying size, while the second kNN step produces LFs with large and varying coverage. User experiments in Appendix 4.1.2 show that real users can judge the latent LF usefulness quickly by visually inspecting the consistency of the initial cluster and a small selection of the cluster nearest neighbors.

Implementation Details The **probabilistic ensemble** in IWS, which is used in all acquisition functions to learn $p(u_j = 1|Q_t)$, is a bagging ensemble of $s = 50$ multilayer perceptrons with two hidden layers of size 10, RELU activations, sigmoid output and logarithmic loss. To create features for the p candidate LFs in $\mathcal{L}$, I use singular value decomposition (SVD) to project from n to $d' = 150$. Thus, at iteration t, given a query dataset $Q_{t-1} = \{(\lambda_j, u_j)\}_{j=1}^{t-1}$, the ensemble is trained on pairs $\{(\tau(\lambda_j), u_j)\}_{j=1}^{t-1}$ where $\tau(\lambda_j)$ are the SVD features and u_j the binary expert responses. The output of the ensemble on LFs not in the query dataset is used to compute $\sigma_j(Q_{t-1}) = \sqrt{\mathrm{Var}[g^{-1}(p(u_j = 1|Q_{t-1}))]}$ and $\mu_j(Q_{t-1}) = \mathbb{E}[g^{-1}(p(u_j = 1|Q_{t-1}))]$. While g, which maps α_j to v_j, could be fine-tuned from data, I set g as the identity function in the experiments, which works well empirically. Finally, to allow human experts to express some level of confidence about their decision on u_j, I also collect corresponding uncertainty weights $b_j \in \{1, 0.5\}$, and I multiply the contribution to the loss of each u_j by the respective weight b_j. Users can also skip queries if they are unsure, indicated in black in Fig. 4.3. These ‘unsure’ responses are still counted as an iteration/query in the plots.

[3] Image A is a k_1 nearest neighbor of image B, and image B is also a k_1 nearest neighbor of image A.

The downstream **end classifier** f is a multilayer perceptron with two hidden layers of size 20 and RELU activations, sigmoid output and logarithmic loss. Each model in the ensemble as well as f are optimized using Adam [146]. For the text datasets, I fit the end models f to low dimensional projections of a large bag-of-words matrix via truncated Singular Value Decomposition (SVD), fixing the embedding size to $d = 300$. I repeat each experiment ten times. I assume that the class balance is known when fitting the label model, as common in related work. When class balance is unknown, [221] discuss an unsupervised approach to estimate it. For the COCO image experiments, I use the second-to-last layer of a ResNet-18 [117] pretrained on ImageNet to obtain image features. These image features are used as the embedding to train the end classifier for all approaches which I compare. The embeddings are also used to create the nearest-neighbor based image LFs.

The first 8 iterations of IWS are initialized with queries of four LFs known to have accuracy between 0.7 and 0.75 drawn at random and four randomly drawn LFs with arbitrary accuracy. Subsequently, IWS chooses the next LFs to query. Active learning is initialized with the same number of known samples.

Oracle experiments

The simulated oracle labels an LF as useful if it has an accuracy of at least 0.7. Measured by test-set AUC of final classifier f, IWS-LSE outperforms other approaches significantly on five out of six text datasets, and matches the best performance also attained by Snuba on one dataset, see Fig. 4.2. IWS-AS performs similarly well on four text datasets, and competitively on the other two. Both IWS approaches outperform active learning by a wide margin on all text datasets. IWS also quickly approaches the performance achieved by an end model trained on the full ground truth training labels. Ablation results for IWS-LSE varying the final set size as well as thresholds r are provided in Appendix 4.1.2. For the COCO image tasks, LFs were created using image captions as in [254] (Fig. 4.4, first and second plot), as well as on images directly via nearest neighbors (Fig. 4.4, third and fourth plot). IWS also performs competitively on these image tasks and quickly approaches the performance achieved using all training ground truth.

User experiments on text

Experiments of IWS-AS are conducted with real users on the Amazon and IMDB review sentiment classification tasks. The results demonstrate that users judge high accuracy functions as useful and make few mistakes. In the experiments, users are shown a descrip-

Table 4.1: A comparison of the median (mean) user response time for responding to queries about labeling functions (LFs) vs samples.

Dataset	Annotate LF	Annotate sample
Amazon	4.2s (8.3s)	7.9s (10.3s)
IMDB	3.2s (6.0s)	19.s (24.3s)

tion of the heuristic (the key term

pattern) and the intended label. Users can also view four snippets of random documents where the LF applied, but are instructed to only consider the examples if necessary. See Appendix D.2 for a screenshot of the query interface and details regarding the user prompts. The top of Fig. 4.3 shows that mean test set performance of IWS-AS using LFs obtained from human feedback closely tracks the simulated oracle performance after about 100 iterations. Fig. 4.3 further shows the queried LFs and corresponding user responses by their true accuracy vs. their non-abstain votes. To match the mean test AUC of IWS-AS obtained after 200 iterations on the Amazon dataset, active learning (uncertainty sampling) requires about 600 iterations. For the IMDB dataset, to achieve the same mean test AUC of IWS-AS obtained after 200 iterations, active learning requires more than 1000 iterations. For both datasets, the average response time to each query was fast. A manual labeling exercise of samples for the IMDB and Amazon datasets (Table 4.1) is also conducted with real users. Assuming the original ratings are true, the users incorrectly classified ∼9% of IMDB reviews while taking significantly longer compared to the response times to LF queries. For the Amazon dataset, users mislabeled ∼2% of samples and were also slower at labeling samples than LFs. The user-study experiments involved nine persons with a computer science background. Neither the true accuracy of each heuristic nor the end model train or test set results were revealed to the users at any stage of the experiment. Section 4.1.2 provides results for a similar user study on the COCO sports task with image LFs. These results are consistent with those for text, showing that users are able to distinguish accurate vs. inaccurate image LFs well, and that the full IWS procedure with real users achieves similar performance as the one using a simulated oracle.

User Experiments on Images with Image Labeling Functions

I also carried out a user study on the COCO Sports image classification task described above, using a family of mutual nearest neighbor image labeling functions. In line with the experiments on text data, Figure 4.5 shows that users were able to judge the accuracy of LFs consistently and well, and that the performance of IWS closely tracks the simulated oracle performance after about 100 iterations.

Again, users were quite quick at responding to LF queries, and judging LFs to be predictive of the latent class variable appeared to be an intuitive task. The average user response time to these image LF queries was 8.8 seconds, while the response time for annotating individual images was around 4.1 seconds on average. To assess an LF, a human user was shown the LFs MkNN image cluster of up to 20 images (the mean size was 7.9 images), and 15 random images contained in the extended cluster, sorted according to their mean distance to the MkNN image cluster. For this nearest neighbor-based family of LFs, the parameter k_1 was set to 20, and k_2 to 1500—but performance was robust to changes in these parameters. While the results show that

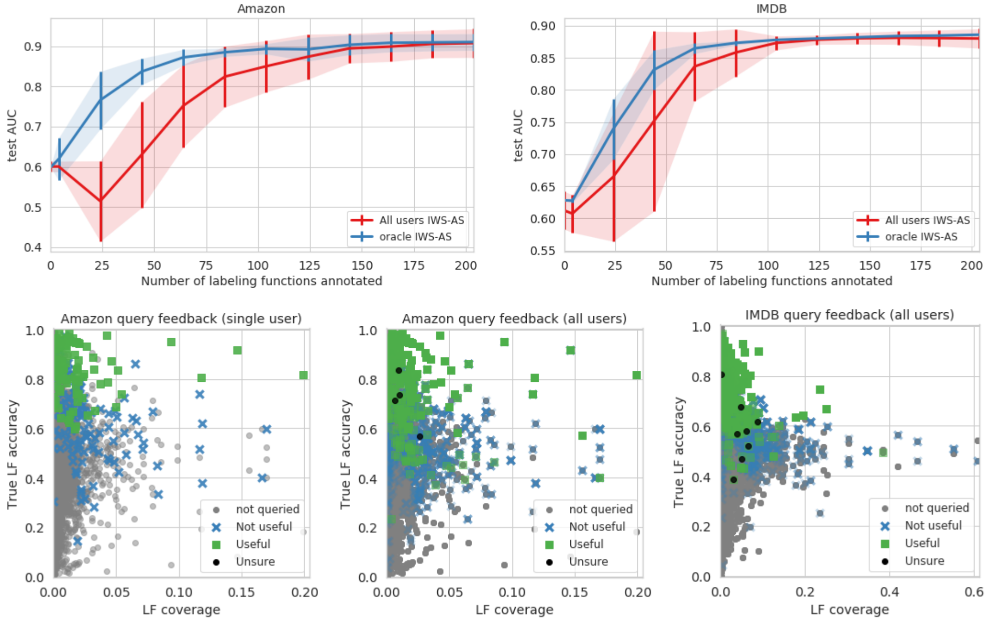


Figure 4.3: **Human user study, text data.** *Top:* Test AUC of end classifiers trained on soft labels obtained via IWS-AS. Test set performance of humans closely tracks performance using a simulated oracle after ∼100 iterations. *Bottom:* scatter plots of human responses to queries showing the true LF accuracy vs LF coverage by one user (lower left) and all users (lower middle and lower right). An 'unsure' response does not provide a label to an LF query but is counted as an annotation.

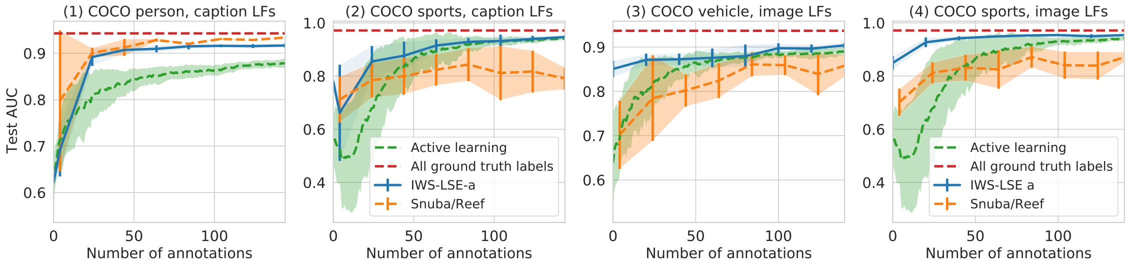


Figure 4.4: COCO image classification. *Images (1) and (2):* Test AUC of image classifiers trained using probabilistic labels obtained from LFs on captions, compared to training with active learning and the full training ground truth. *Images (3) and (4):* Test AUC of image classifiers trained using nearest neighbor based image LFs compared to training with active learning and the full training ground truth. Due to the low coverage of LFs, only IWS-LSE-a is used in the image experiments.

IWS performs well in this setting, and that classifiers can be trained competitively compared to active learning, it is an interesting challenge to develop better image primitives from which labeling functions can be constructed in data programming,

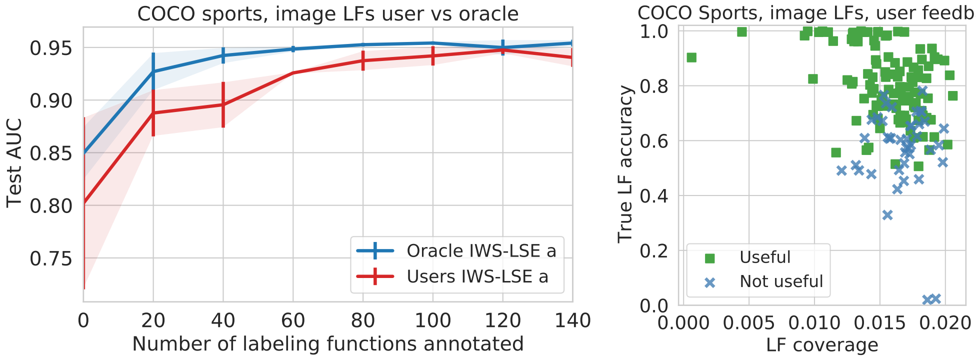


Figure 4.5: **Human user study, image data (Section 4.1.2).** The user experiments in this plot were done using a labeling function family defined directly on the images. *Left:* Test AUC of end classifiers trained on soft labels obtained via IWS-LSE-a. Test set performance of humans closely tracks performance using a simulated oracle after ∼100 iterations on these datasets. *Right:* scatter plots showing the true LF accuracy vs LF coverage of responses to queries by one user.

and generated in IWS and Snuba.

Ablation of IWS parameter settings

This section provides results of ablation experiments for IWS. The IWS-LSE algorithm requires us to set a threshold r on the (unknown) LF accuracy around which the model aims to partition the set of candidate LFs. Fig. 4.6 provides results for different r threshold settings for IWS-LSE-a and IWS-LSE-ac, corresponding to Scenario (A) and Scenario (B). The figure shows that the algorithms perform well across a wide range of r. While there is no clear, distinct performance difference discernible, the figure suggest that a threshold too close to 1.0 can cause the algorithm to underperform. A possible explanation is that as it stifles exploration of LFs within the limited budget of queries to users.

In Scenario (B), which corresponds to the IWS-LSE-ac algorithm, the aim is to find a final set of LFs of limited size. Fig. 4.7 shows that a wide range ($\tilde{m} = 50$ to 200) of final set sizes produce good results. Recall that in the experiments, the size of the final set of LFs at each iteration t is bound by $m = \sum_{i=1}^{t-1} u_i + \tilde{m}$, i.e. the number of LFs so far annotated as $u = 1$ plus a constant $\tilde{m}$.

4.1.3 Discussion

The above results show that a small number of expert interactions with the proposed method can suffice to select good weak supervision sources from a large pool of candidates, leading to competitive end classifiers. IWS shows promise as a way to significantly speed up the process of weak supervision source discovery by domain experts

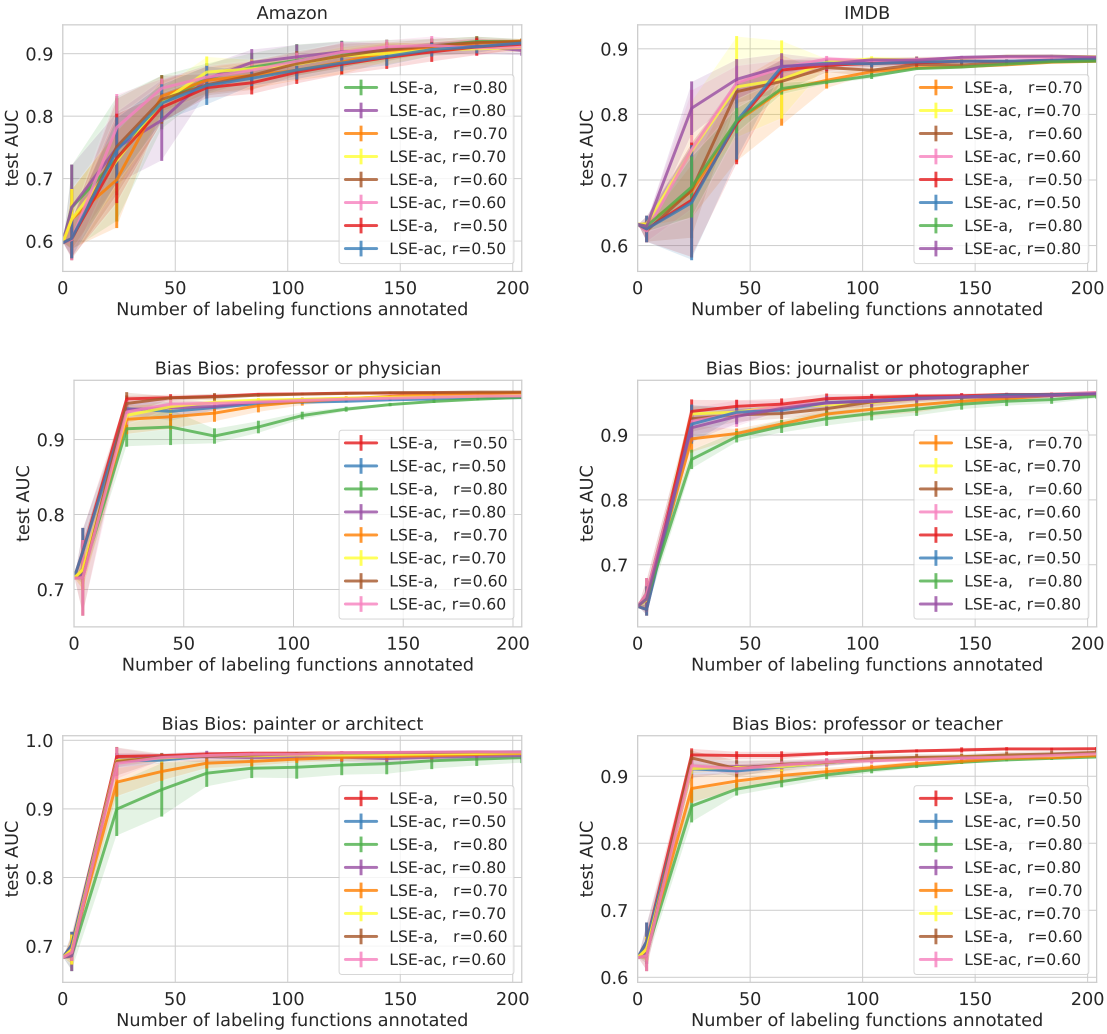


Figure 4.6: IWS-LSE ablation plots for varying thresholds r which are use to partition the set of LFs. On all datasets test set performance is very similar after around 100 iterations, showing that a wide range of such thresholds leads to good test set performance. For IWS-LSE-ac shown in this plot $\tilde{m}$ was set to 100.

as an alternative to devising such sources from scratch. On a large number of tasks, IWS obtains superior predictive performance on downstream test sets compared to the automatic selection of LFs with Snuba [254] and standard active learning (where users annotate samples instead of LFs), when measured with respect to the number of user annotations. Experiments with real users on two text benchmark datasets and one image dataset show that humans recognize and approve high accuracy LFs, yielding models that match performance attainable with a simulated oracle. The text experiments also suggest that tasks exist where users are able to annotate heuristics

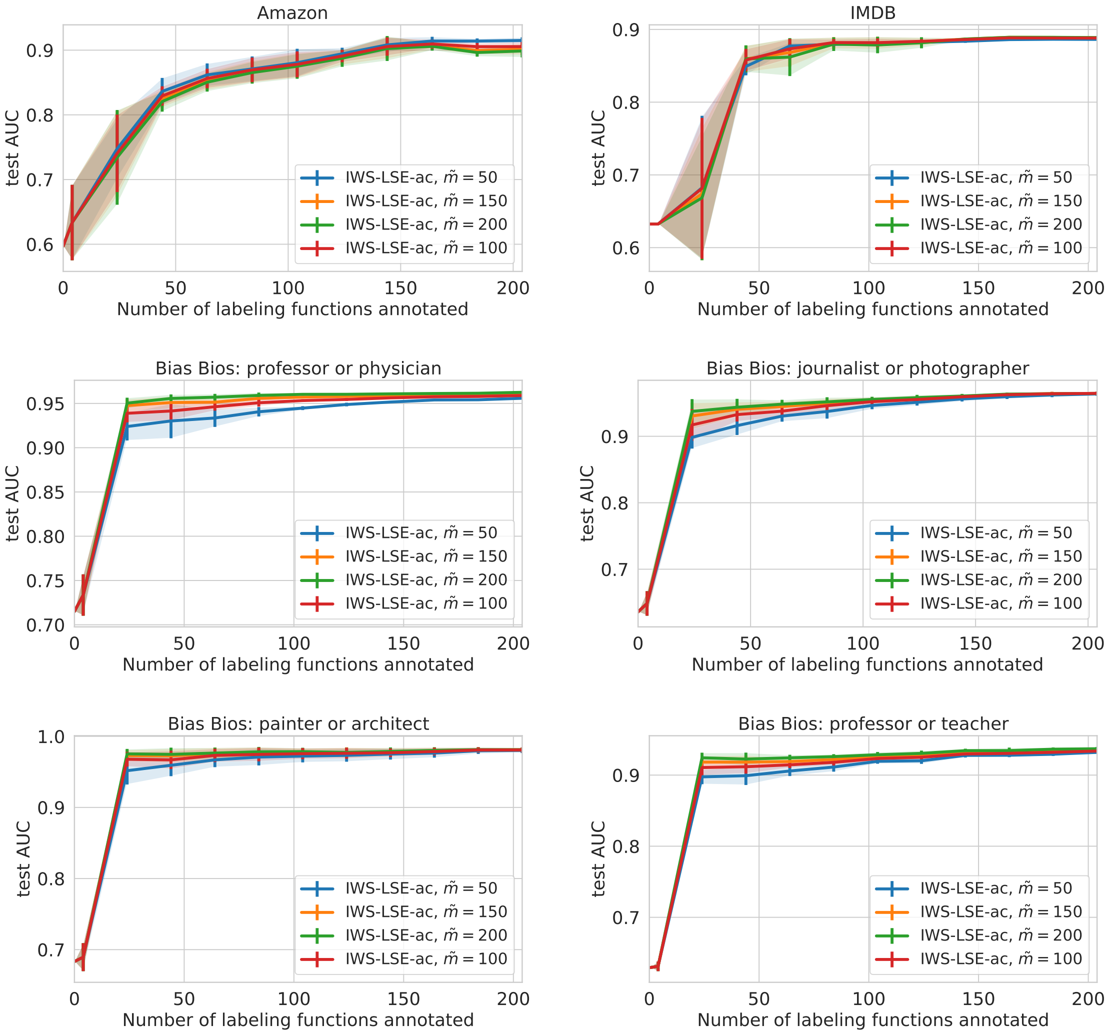


Figure 4.7: IWS-LSE-ac ablation plots for varying final sizes via parameter $\tilde{m}$. Recall that the size of the final set of LFs at each iteration t is bound by $m = \sum_{i=1}^{t-1} u_i + \tilde{m}$, i.e. the number of LFs so far annotated as $u = 1$ plus a constant $\tilde{m}$. Note that the LSE-ac setting takes LF coverage into account to rank LFs according to $(2\alpha_j - 1) * \hat{l}_j$ where $\alpha_j, \hat{l}_j$ are the estimated LF accuracy and observed LF coverage.

faster than individual samples. The proposed approach is not meant to replace active learning or manually created weak supervision. For many tasks, standard active learning, data programming, or crowd-sourcing may be entirely sufficient. However, when a large number of labels is necessary to train good models while experts' effort is precious, an interactive framework for weak supervision can be the right choice.

4.2 Weak Supervision as Paired Multi-Modal Data: Vision–Language Processing in Biomedicine

This Section is based on the work presented in:
Boecking, Benedikt et al. "Making the Most of Text Semantics to Improve Biomedical Vision–Language Processing". In: *European Conference on Computer Vision (ECCV)*. 2022

Benedikt Boecking (myself) and Naoto Usuyama contributed equally to the research article published at ECCV.

This section focuses on vision–language processing (VLP) for paired image and text data in the biomedical domain. The weak form of supervision here is the knowledge of the paring of samples across the modalities, i.e. we know which pairs of images and text documents go together. In particular, in this application we know which text report was written to describe the findings of a specific imaging study. In related work, this learning scenario has been framed under the terms *weak supervision* as well as multi-modal *self-supervised learning*. From hereon, I will refer to this scenario as self-supervised learning, as this terminology has evolved as the predominant one in related studies. However, I want to emphasize that the 'self-learning' is only possible because the latent shared entities across the modalities are linked weakly through knowledge about which samples are associated with each other. This in turn allows for the formulation of contrastive objectives on the basis of imprecise positive and negative samples derived from this known relationship.

The setting of paired multi-modal image-text data is common in weak supervision research. Many domains exist in which large amounts of raw signal data such as time series or images are accompanied by unstructured or semi-structured text documents that capture knowledge about the unobserved target variables. One example of such data are maintenance records in aviation, paired with engine data. On online platforms such as Flickr one can find large numbers of images associated with captions written by users. For videos of sports events, news programs, or movies, large datasets of video segments with closed captions are common and routinely used in vision-language processing.

In healthcare, paired visual and text data is collected routinely during clinical practice, and common examples are X-ray images [85, 132, 266] or computed tomography (CT) scans [52, 85, 88, 251] paired with reports written by medical experts. Importantly, while many remain private, some paired clinical datasets [34, 75, 136] have been released to the research community such as MIMIC-CXR [136].

The multi-modal paired scenario has also been exploited in programmatic weak supervision, for example to train image classifiers [266, 132, 85, 92, 88] in data such as radiology images or CT scans. Here, imperfect rules are defined on the text documents, a label model then estimates the unobserved true label on the basis of the weak supervision votes, and an end model that receives the images as input is learned using this estimate.

Self-supervised Vision–Language Processing The goal of self-supervised VLP is to jointly learn good image and text representations that can be leveraged by downstream applications such as zero-/few-shot image classification, report generation and error detection, and disease localization. Self-supervised VLP has several advantages over supervised learning and programmatic weak supervision, not just because it does not require laborious manual annotations or the creation of weak supervision sources by users, but also because it does not operate on a fixed number of predetermined conditions or object categories, since the joint latent space is learned from raw text.

Self-supervised Vision–Language Processing in the Biomedical Domain Advances in deep learning have enabled automated diagnosis systems that operate near or above expert-level performance, paving the way for the use of machine learning systems to improve healthcare workflows, for example by supporting fast triaging and assisting medical professionals to reduce errors and omissions [52, 87, 189, 251]. A major hurdle to the widespread development of these systems is a requirement for large amounts of detailed ground-truth clinical annotations for supervised training, which are expensive and time-consuming to obtain. Motivated by this challenge, there has been a rising interest in multi-modal self-supervised learning [128, 166] and cross-modal programmatic weak supervision [85, 88, 132, 251, 266], in particular for paired image–text data.

In contrast to the general domain setting, self-supervised VLP with biomedical data poses additional challenges. Take radiology as an example, publicly available datasets [136, 75, 34] are usually smaller, on the order of a few hundred thousand pairs rather than millions in general-domain vision–language processing (e.g. [215] collected 400M text–image pairs on the Internet for self-supervision). Furthermore, linguistic challenges are different in biomedical settings, including common usage of negations, expressions of uncertainty, long-range dependencies, more frequent spatial relations, the use of domain-specific modifiers, as well as scientific terminology rarely found in the general domain. Taking negation as an example, “there is no dog in this picture” would be a highly unusual caption on social media, but “there is no evidence of pneumonia in the left lung” or “there are no new areas of consolidation to suggest the presence of pneumonia” are descriptions commonly found in radiology reports. Moreover, pretrained models including object detectors often used in general domain visual grounding are typically unavailable or under-perform in domain-specific applications (see also Supp. in [128]). Additionally, imbalance in underlying latent entities of interest (e.g., pulmonary findings) can cause larger numbers of false negatives in contrastive learning objectives that sample at random, which can lead models to degrade and memorize irrelevant text and image aspects. For example, radiology images

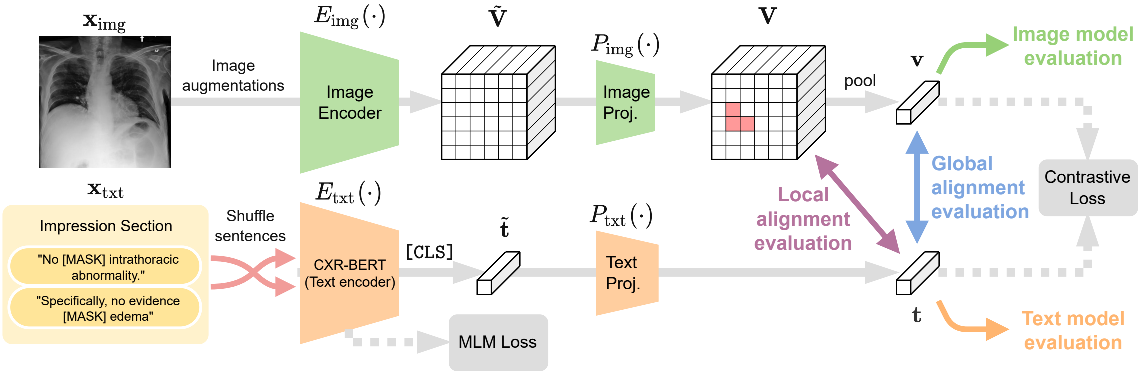


Figure 4.8: BioViL leverages a radiology-specific text encoder (CXR-BERT), text augmentation, regularization, and maintains language model quality via a MLM loss. A broad evaluation of models and representations is conducted which includes zero-shot classification, phrase grounding, and natural language inference.

and text reports with normal findings occur much more frequently compared to exams that reveal abnormal conditions such as pneumonia or pneumothorax (also see [61]). Supp. E.2.1 provides further discussion of these challenges.

Making the Most of Text Semantics to Improve Biomedical Vision–Language Processing Related self-supervised VLP work in biomedicine [124, 128, 166, 194, 292] has achieved impressive downstream classification and zero-shot classification performance. However, the work here reveals that sub-optimal text modeling due to insufficient vocabulary adjustment, fine-tuning, and language grounding during joint training appears to have gone unnoticed, all of which are shown to degrade the quality of the latent representations. In particular, a more thorough benchmarking of the text, image, and shared embeddings, across a multitude of downstream benchmarks, reveals that large improvements in performance are possible by taking care to build highly specialized text models and by maintaining their performance during joint training. Free-text image descriptions provide a semantically dense learning signal compared to image-only contrastive methods and supervised classification [78]. Further, extracting shared semantics of images and text pairs is easier for text, as the modality is already discretized. Thus, making the most of text modeling before and during joint training can lead to large improvements in not just the text model, but also of the image model and joint representations. This section present the following contributions:

1. A new Chest X-ray (CXR) domain-specific language model, CXR-BERT[4] (Fig. 4.9). Through an improved vocabulary, a novel pretraining procedure, regularization, and text augmentation, the model considerably improves radiology natural lan-

[4] Pretrained models available on HuggingFace: `https://aka.ms/biovil-models`

guage inference [189], radiology masked token prediction [79, 174], and downstream VLP task performance.

2. A simple but effective self-supervised VLP approach for paired biomedical data named BioViL[45] (Fig. 4.8), and evaluate in the radiology setting. Through improvements in text modeling, text model grounding, augmentation, and regularization, the approach yields new state-of-the-art performance on a wide range of public downstream benchmarks. Large-scale evaluation conducted in this section (see Table 4.3) includes phrase grounding, natural language inference [189], as well as zero-/few-shot classification and zero-shot segmentation via the RSNA Pneumonia dataset [237, 266]. Notably, the approach achieves improved segmentation performance despite only using a global alignment objective.

3. A dataset for phrase grounding in radiology, `MS-CXR`[6], to encourage reproducible evaluation of shared latent semantics learned by biomedical image-text models. This large, well-balanced phrase grounding benchmark dataset contains carefully curated image regions annotated with descriptions of eight radiology findings, as verified by board-certified radiologists. Unlike existing chest X-ray benchmarks, this challenging phrase grounding task evaluates joint, local image-text reasoning while requiring real-world language understanding, e.g. to parse domain-specific location references, complex negations, and bias in reporting style.

4.2.1 Methodology

Assume that we are given a set $\mathcal{D}$ of pairs of radiology images and reports $(\mathbf{x}_{\text{img}}, \mathbf{x}_{\text{txt}})$. Let $\mathbf{w} = (w_1, \ldots, w_T)$ denote a vector of T (sub-)word tokens of a text document $\mathbf{x}_{\text{txt}}$ (after tokenization). Recall that a BERT [256] encoder E_{txt} outputs a feature vector for each input token w_t as well as a special global `[CLS]` token used for downstream classification. Let $\tilde{\mathbf{t}} = [E_{\text{txt}}(\mathbf{w})]_{\texttt{[CLS]}}$ denote the `[CLS]` token prediction by E_{txt} based on input $\mathbf{w}$, and $\mathbf{t} = P_{\text{txt}}(\tilde{\mathbf{t}})$ its lower-dimensional projection by a model P_{txt}.

CXR-BERT: Domain-Specific Language Model Pretraining

The proposed CXR-BERT (Fig. 4.9) is a specialized Chest X-ray (CXR) language model with an adjusted vocabulary, pretrained in three phases to capture dense semantics in radiology reports [38]. To achieve this specialization to the CXR report domain despite limited data availability, the approach includes pretraining on larger data from closely related domains. The phases proceed as follows: **(I)** First, a custom

[5] Code can be found at: `https://aka.ms/biovil-code`

[6] The `MS-CXR` dataset can be found on PhysioNet `https://aka.ms/ms-cxr`.

WordPiece [272] vocabulary of 30k tokens is constructed based on PubMed abstracts[7] (15 GB), MIMIC-III [137] clinical notes (3.5 GB), and MIMIC-CXR radiology reports (0.1 GB). With this custom vocabulary, the model produces fewer sub-word breakdowns (Table 4.2). **(II)** Second, a randomly initialized BERT model is pretrained via Masked Language Modeling (MLM) on the PubMed + MIMIC-III + MIMIC-CXR corpora, largely follow RoBERTa [174] pretraining configurations, i.e. dynamic whole-word masking for MLM and packing of multiple sentences into one input sequence. This phase aims to build an initial domain-specific BERT model in the biomedical and clinical domains. **(III)** Third, pretraining is continued on MIMIC-CXR only, to further specialize CXR-BERT to the CXR domain. Here, a novel sequence prediction task is added to the objective to obtain better sequence representations, as explained below.

Note that a raw radiology report $\mathbf{x}_{\text{txt}}$ typically consists of several sections, including a 'Findings' section that details clinical observations, and an 'Impression' section summarizing the clinical assessment [261, 268]. The sequence prediction objective of phase (III) aims to take advantage of this structure. Specifically, MLM pretraining is continually run on MIMIC-CXR radiology reports, and a radiology section matching (RSM) pretraining task is added, formulated to match Impression to Findings sections of the same study.

Let θ denote the weights of the language model and $m \subset \{1, \dots, T\}$ denote mask indices for M masked tokens, randomly sampled for each token vector $\mathbf{w}$ at every iteration. Given a batch $\mathcal{B}$ of token vectors $\mathbf{w} = (w_1, \dots, w_T)$, we write the MLM loss as the cross-entropy for predicting the dynamically masked tokens: $\mathcal{L}_{\text{MLM}} = -\frac{1}{|\mathcal{B}|} \sum_{\mathbf{w} \in \mathcal{B}} \log p_\theta(\mathbf{w}_m \,|\, \mathbf{w}_{\setminus m})$. Further, let $(\tilde{\mathbf{t}}_i^{\text{F}}, \tilde{\mathbf{t}}_i^{\text{I}})$ denote a pair of `[CLS]` to-

[7] Obtained via `https://pubmed.ncbi.nlm.nih.gov/download/`

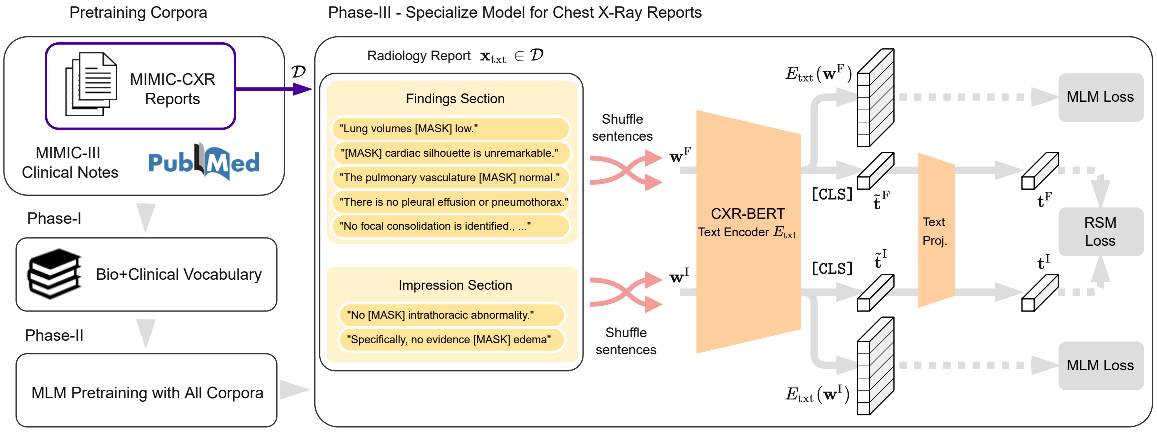

Pretraining Corpora
MIMIC-CXR Reports
$\mathcal{D}$
MIMIC-III Clinical Notes
PubMed
Phase-I
Bio+Clinical Vocabulary
Phase-II
MLM Pretraining with All Corpora
Phase-III - Specialize Model for Chest X-Ray Reports
Radiology Report $\mathbf{x}_{\text{txt}} \in \mathcal{D}$
Findings Section
"Lung volumes [MASK] low."
"[MASK] cardiac silhouette is unremarkable."
"The pulmonary vasculature [MASK] normal."
"There is no pleural effusion or pneumothorax."
"No focal consolidation is identified., ..."
Impression Section
"No [MASK] intrathoracic abnormality."
"Specifically, no evidence [MASK] edema"
Shuffle sentences
Shuffle sentences
$\mathbf{w}^{\text{F}}$
$\mathbf{w}^{\text{I}}$
CXR-BERT
Text Encoder E_{txt}
$E_{\text{txt}}(\mathbf{w}^{\text{F}})$
$E_{\text{txt}}(\mathbf{w}^{\text{I}})$
[CLS]
[CLS]
$\tilde{\mathbf{t}}^{\text{F}}$
$\tilde{\mathbf{t}}^{\text{I}}$
Text Proj.
$\mathbf{t}^{\text{F}}$
$\mathbf{t}^{\text{I}}$
MLM Loss
RSM Loss
MLM Loss

Figure 4.9: The proposed CXR-BERT text encoder has three phases of pretraining and uses a domain-specific vocabulary, masked language modeling (MLM) and radiology section matching (RSM) losses, regularization, and text augmentations.

Table 4.2: Vocabulary comparison of common radiology terms with ClinicalBERT (Wiki/Book, cased), PubMedBERT (PubMed, uncased), and CXR-BERT (PubMed+MIMIC-III/CXR, uncased). ✓ marks that a word appears in the vocabulary, otherwise its sub-tokens are shown.

Full word	ClinicalBERT	PubMedBERT	CXR-BERT
pneumonia	✓	✓	✓
opacity	op-acity	✓	✓
effusion	e-ff-usion	✓	✓
pneumothorax	p-ne-um-oth-orax	✓	✓
atelectasis	ate-lect-asis	ate-le-ct-asis	✓
cardiomegaly	card-io-me-gal-y	cardio-me-gal-y	✓
bibasilar	bi-bas-ila-r	bib-asi-la-r	✓

kens corresponding to the Findings and Impression sections of the same i^{th} report, and let $(\mathbf{t}_i^{\text{F}}, \mathbf{t}_i^{\text{I}})$ denote the pair projected to a lower dimension via a two-layer perceptron P_{txt}. Now, let us define the RSM contrastive loss on the text modality. This loss over N pairs of samples favors Impression and Findings text pairs from the same report over unmatched ones:

$$\mathcal{L}_{\text{RSM}} = -\frac{1}{N}\sum_{i=1}^{N}\left(\log\frac{\exp(\mathbf{t}_i^{\text{F}}\cdot\mathbf{t}_i^{\text{I}}/\tau_1)}{\sum_{j=1}^{N}\exp(\mathbf{t}_i^{\text{F}}\cdot\mathbf{t}_j^{\text{I}}/\tau_1)} + \log\frac{\exp(\mathbf{t}_i^{\text{I}}\cdot\mathbf{t}_i^{\text{F}}/\tau_1)}{\sum_{j=1}^{N}\exp(\mathbf{t}_i^{\text{I}}\cdot\mathbf{t}_j^{\text{F}}/\tau_1)}\right), \tag{4.8}$$

where $\tau_1 > 0$ is a scaling parameter to control the margin. The resulting total loss of the specialization phase (III) is $\mathcal{L}_{\text{III}} = \mathcal{L}_{\text{RSM}} + \lambda_{\text{MLM}}\mathcal{L}_{\text{MLM}}$. An additional important component for regularizing the RSM loss is the use of increased dropout (25%), including on attention. In the experiments, $\tau_1 = 0.5$ and $\lambda_{\text{MLM}} = 0.1$, determined by a limited grid-search measuring $\mathcal{L}_{\text{GA}}$ (Eq. (4.9)) of the joint model on a validation set. Also, note that similar losses to the RSM loss–defined over the same or separate text segments–have been explored successfully for sentence representation learning [95, 177] in other settings. Empirically, experiments conducted for the work presented in this section showed that an objective as in [95] using masked Findings to Findings matching can achieve similar performance and may be an appropriate replacement in other biomedical settings with differing text structure.

Text Augmentation. As domain-specific datasets are often quite small, effective text augmentation can induce large benefits. In the radiology domain, the sentences of the Findings and Impression sections, which contain the detailed description and summary of the radiological findings, are usually permutation-invariant on the sentence level (cf. [214]). Thus, in the experiments of this work sentences are randomly shuffled within each report section as an effective text-augmentation strategy for both pretraining of CXR-BERT as well as during joint model training.

BioViL: Vision-Language Representation Learning

Let us now introduce BioViL, a simple but effective self-supervised VLP setup for the biomedical domain (Fig. 4.8), and here studied in a CXR application setting. BioViLuses a convolutional neural network (CNN) [155] image encoder E_{img}, the CXR-BERT text encoder E_{txt} proposed above, and projection models P_{img} and P_{txt} to learn representations in a joint space. The CNN model provides useful inductive biases given the limited amount of image data available for training, and allows us to obtain a grid of local image embeddings $\tilde{\mathbf{V}} = E_{\text{img}}(\mathbf{x}_{\text{img}})$, which is fine-grained enough to be useful for segmentation (e.g. 16×16). Each encoder is followed by a modality-specific two-layer perceptron projection model P, which projects the encoded modality to a joint space of 128 dimensions–e.g., $\mathbf{V} = P_{\text{img}}(\tilde{\mathbf{V}})$–where the representation is ℓ_2-normalized. Note that projection should be applied to local embeddings before mean-pooling $\mathbf{v} = \text{pool}(P_{\text{img}}(\tilde{\mathbf{V}}))$, which gives us the global image embedding $\mathbf{v}$. The text branch uses the IMPRESSION section's projected `[CLS]` token $\mathbf{t}^{\text{I}}$ as the text representation in the joint space, as it contains a succinct summary of radiological findings. To align the representations and learn a joint embedding, the use two loss terms is proposed. For a batch of size N, a symmetric contrastive loss [205] for *global alignment* of the image and text projections helps us learn the shared latent semantics:

$$\mathcal{L}_{\text{GA}} = -\frac{1}{N}\sum_{i=1}^{N}\left(\log\frac{\exp(\mathbf{v}_i \cdot \mathbf{t}_i^{\text{I}}/\tau_2)}{\sum_{j=1}^{N}\exp(\mathbf{v}_i \cdot \mathbf{t}_j^{\text{I}}/\tau_2)} + \log\frac{\exp(\mathbf{t}_i^{\text{I}} \cdot \mathbf{v}_i/\tau_2)}{\sum_{j=1}^{N}\exp(\mathbf{t}_i^{\text{I}} \cdot \mathbf{v}_j/\tau_2)}\right), \tag{4.9}$$

where $\tau_2 > 0$ is a scaling parameter. Importantly, the $\mathcal{L}_{\text{MLM}}$ loss is maintained during joint training to avoid degradation of language modeling performance, resulting in the final joint loss $\mathcal{L}_{\text{joint}} = \lambda_{\text{GA}}\mathcal{L}_{\text{GA}} + \mathcal{L}_{\text{MLM}}$. In the experiments and released models $\tau_2 = 0.5$ and $\lambda_{\text{GA}} = 0.5$, determined by a limited grid search measuring $\mathcal{L}_{\text{GA}}$ on a validation set.

Augmentations, Regularization, and Image Encoder Pretraining. Due to the small dataset sizes expected in biomedical applications, image and text augmentations are used to help learn known invariances. A ResNet-50 [117] architecture is used as the image encoder, pretrained on MIMIC-CXR images using a SimCLR [45] objective with domain-specific augmentations as detailed in Section 4.2.3. For text, the same sentence-shuffling augmentation as in pretraining of CXR-BERT is used (see Section 4.2.3 for details). Furthermore, as in phase (III) of CXR-BERT training, higher text encoder dropout (25%) than in standard BERT settings is applied [79, 256]. The combination of all these components, including continuous MLM optimization, is important to improve downstream performance across the board (see ablation in Table 4.5).

Zero-shot Classification. After joint training, text prompts are used to cast the zero-shot classification problem into an image–text similarity task as in [128, 215, 219]. For C classes, subject-matter experts design C text prompts representing the target labels $c \in \{1, \ldots, C\}$, e.g. for presence or absence of pneumonia (see Section 4.2.3). Each class prompt is represented as a vector of tokens $\mathbf{w}^c$ and passed to the text encoder and projector of BioViL to obtain ℓ_2-normalized text features $\mathbf{t}^c = P_{\text{txt}}(E_{\text{txt}}(\mathbf{w}^c)) \in \mathbb{R}^{128}$. For each input image $\mathbf{x}_{\text{img}} \in \mathbb{R}^{H \times W}$, the image encoder and projection module are used to obtain patch embeddings $\mathbf{V} = P_{\text{img}}(E_{\text{img}}(\mathbf{x}_{\text{img}})) \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times 128}$ for segmentation tasks or the pooled embedding $\mathbf{v} = \text{pool}(\mathbf{V}) \in \mathbb{R}^{128}$ for instance-classification. Dilated convolutions [282] are used to obtain higher-resolution feature maps. Probabilities for classes/regions can then be computed via a softmax over the cosine similarities between the image (or region) and prompt representations.

Few-shot Tasks with BioViL. To further assess the representation quality, linear probing is applied to local ($\mathbf{V}$) and global ($\mathbf{v}$) image representations, by learning $\boldsymbol{\beta} \in \mathbb{R}^{128 \times C}$ weights and a bias term. Unlike [128, 292], this work leverages the pretrained projectors and class text embedding $\mathbf{t}^c$ from the zero-shot setting by using them for initialization, which leads to improved performance and further reduces the need for manual label collection. Specifically, in few-shot classification settings, the weights and bias are initialized with $\boldsymbol{\beta} = [\mathbf{t}^1, \ldots, \mathbf{t}^C]$ and zeros, respectively.

4.2.2 A New Phrase Grounding Benchmark

Accurate local alignment between modalities is an important characteristic of successful joint image-text training in healthcare, in particular since image and report samples often contain multiple clinical findings, each of which correspond to distinct image regions. Standard global-alignment approaches may attain high classification accuracy by overfitting to spurious image features for a given finding (e.g., chest tubes in images correlating with mentions of pneumothorax in reports). Image classification, the most frequently evaluated downstream task in related work [128, 166, 194, 292], requires only scene-level labels, hence a less sophisticated understanding of natural-language image descriptions. Image classification tasks can largely be solved by simply detecting a small set of words and maintaining some understanding of negation, as exemplified by the development of automated, rule-based text-labelers such as CheXpert [132]. Instance-level image-text retrieval tasks address some evaluation limitations, but do not require the level of language reasoning needed to solve local correspondence between phrases and image regions. Existing public CXR benchmark datasets to evaluate local aspects of VLP have one or more of the following limitations (see Supp. E.3,E.4 for more details): bounding boxes without corresponding free text descriptions, a limited number of samples, a limited number of abnormalities, and non-curated phrases impacting evaluation quality.

With this motivation in mind, the proposed `MS-CXR` was designed as a radiology visual-grounding benchmark that has domain-specific language (e.g., paraphrasing

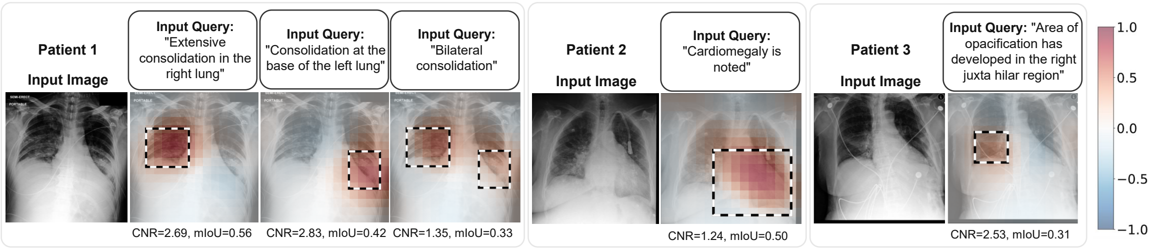

Patient 1
Input Image
Input Query: "Extensive consolidation in the right lung"
Input Query: "Consolidation at the base of the left lung"
Input Query: "Bilateral consolidation"
CNR=2.69, mIoU=0.56
CNR=2.83, mIoU=0.42
CNR=1.35, mIoU=0.33
Patient 2
Input Image
Input Query: "Cardiomegaly is noted"
CNR=1.24, mIoU=0.50
Patient 3
Input Image
Input Query: "Area of opacification has developed in the right juxta hilar region"
CNR=2.53, mIoU=0.31
1.0
0.5
0.0
−0.5
−1.0

Figure 4.10: Examples from the newly released MS-CXR phrase grounding dataset with BioViL latent vector similarity for different input text queries superimposed as heat-maps. Dashed boxes are ground-truth annotations by radiologists. X-ray images are mirrored horizontally.

and negations) and forms a more challenging real-world image-text reasoning task compared to existing evaluation datasets. To name just a few challenges, the phrase grounding task requires the ability to parse domain specific location modifiers, the ability to deal with reporting style biases, and understanding of complex negations, all while relating the correct findings to specific image regions.

MS-CXR – A Chest X-ray Phrase Grounding Benchmark

MS-CXR is a new, publicly released dataset containing image bounding box labels paired with radiology text descriptions. Two board-certified radiologists assigned and verified the annotations in MS-CXR (see examples in Figs. 4.10 and E.5). MS-CXR provides 1153 image–sentence pairs of bounding boxes and corresponding phrases, collected across eight different cardiopulmonary radiological findings, with an approximately equal number of pairs for each finding (see Table E.5). It is curated to ensure gold-standard evaluation of phrase grounding. The phrases in MS-CXR are not simple short captions, but genuine descriptions of radiological findings from original radiology reports [136] and dictated transcripts [20]. Thus, compared to existing evaluation datasets, this proposed benchmark is a more challenging real-world image-text reasoning task.

All the benchmark samples were chosen from the public MIMIC-CXR dataset [99, 136]. To collect a set of bounding-box labels, samples were first selected from a set of studies with pre-existing image annotations (e.g., ellipses) [20, 248] and their correctness was verified by radiologists. To link each image region with candidate phrases, sentences were sampled from the report of each study by extracting the highest matching sentences to the annotated labels using scores of the CheXbert classifier [242], and transcriptions of dictations were also used when available [20]. Next, to better balance findings, additional studies were sampled at random, and ones used in the ImaGenome dataset [271] were added, the latter being a dataset of annotations of anatomical regions. Note that these sampled studies did not have

Table 4.3: Comparing evaluations conducted in recent CXR image-text alignment studies.

Downstream task	Used in ref.*	Image encoder	Text encoder	Phrase reasoning	Findings localization	Latent alignment	Annotation availability
Natural language inference	[B]	-	✓	✓	-	-	Scarce
Phrase grounding	[B]	✓	✓	✓	✓	✓	Scarce
Image classification	[B,C,G,L,M]	✓	-	-	-	-	High
Zero-shot image classif.	[B,G]	✓	✓	-	-	✓	Moderate
Dense image prediction (e.g. segmentation)	[B,G,L]	✓	-	-	✓	-	High
Global image–text retrieval	[C,G]	✓	✓	-	-	✓	High

*B, BioViL (Proposed); C, ConVIRT [292]; G, GLoRIA [128]; L, LoVT [194]; M, Local MI [166].

preexisting region proposals. Radiologists then manually reviewed separate sets of candidates. If a bounding box was not available, the radiologists manually annotated the corresponding region(s) in the image with new bounding boxes. Radiologists rejected studies where no correct phrase candidates were available and where existing bounding boxes were placed incorrectly (e.g., covering too large an area). To ensure a high quality, consistent benchmark, the phrase-image samples that did not adhere to specific guidelines (see Supp. E.3.1) were filtered out, such as phrases containing multiple abnormalities in distinct lung regions.

4.2.3 Experiments

A comprehensive evaluation of the CXR-BERT language model as well as the proposed BioViL self-supervised VLP approach is done, and both are compared to state-of-the art counterparts. Table 4.3 shows how the evaluation coverage of this work compares to recent related studies. This section begins by demonstrating CXR-BERT's superior performance and improved vocabulary, including on a radiology-specific NLI benchmark. Next, the joint image-and-text understanding of BioViL is assessed on the new `MS-CXR` benchmark, which evaluates grounding of phrases describing radiological findings to the corresponding image regions. Zero-shot classification and fine-tuning performance of BioViL is also investigated on image- and pixel-level prediction tasks via the RSNA pneumonia dataset [237, 266].

Setup

Datasets. Experiments are conducted with the MIMIC-CXR v2 [136, 99] chest radiograph dataset, which provides 227,835 imaging studies with associated radiology reports for 65,379 patients, all collected in routine clinical practice. Only frontal view scans (AP and PA) are used, and studies without an Impression section are discarded. From this data, a training set of 146.7k samples is established, and a set of 22.2k validation samples, and it is ensured that all samples used for the different downstream evaluations are kept in a held-out test set. *No labels are used during pre-training*; for early stopping, only a loss on unlabeled validation data is tracked. For evaluation, RadNLI [189] is used to assess the proposed CXR-BERT text model in isolation, the new `MS-CXR` assesses joint image–text understanding via phrase grounding, and the RSNA Pneumonia dataset [237, 266] is used to test zero-shot segmentation, as well as zero-shot and fine-tuned classification performance.

Image and Text Pre-processing. Images are downsized and center cropped to a resolution of 512×512 whilst image aspect ratios are preserved. Image augmentations are performed during training including: random affine transformations, random color jitter, and horizontal flips (only for image fine-tuning tasks). For text model pre-training, the 'Findings' and 'Impression' sections of reports are used, while joint training is performed using only the latter. During training, sentence shuffling is performed within sections as text-augmentation. Additionally, a limited automatic typo correction is done as in [40].

Comparison Approaches. The proposed CXR-BERT text model is compared to the other specialized PubMedBERT [107] and ClinicalBERT [4] models. Note that ClinicalBERT was used in most related studies [128, 166, 292, 194]. BioViL is compared to the closely related, state-of-the-art ConVIRT [292], LoVT [194] and GLoRIA [128] approaches. Lastly, BioViL-L is created by extending BioViL with the local loss term introduced in [128] to illustrate the complementary role of proposed pre-training strategy to recent advances in biomedical VLP.

Metrics. Segmentation results are reported via mean intersection over union (mIoU) and contrast-to-noise ratio (CNR), and the Dice score [59] is reported to compare to [194]. First, the cosine similarity is computed between a projected phrase embedding $\mathbf{t}$ and local image representations $\mathbf{V}$, resulting in a grid of scores between $[-1, 1]$. The similarities are later thresholded to compute mIoU and Dice score. The mIoU is defined as an average over the thresholds [0.1, 0.2, 0.3, 0.4, 0.5]. The CNR measures the discrepancy between scores inside and out of the bounding box region, without requiring hard thresholds. This evaluation of local similarities is important as some clinical downstream applications may benefit from heat-map visualizations as opposed to discrete segmentation. For CNR, let A and $\overline{A}$ denote the interior and exterior of the bounding box, respectively. Then $\text{CNR} = |\mu_A - \mu_{\overline{A}}|/(\sigma_A^2 + \sigma_{\overline{A}}^2)^{\frac{1}{2}}$, where μ_X and σ_X^2 are the mean and variance of the similarity values in region X.

Text Model Evaluation

Natural Language Understanding. The RadNLI benchmark [189] is used to evaluate how well the proposed CXR-BERT text model captures domain-specific semantics. The dataset contains labeled hypothesis and premise pairs, sourced from

Table 4.4: Evaluation of text encoder intrinsic properties and fine-tuning for radiology natural language inference: (1) RadNLI fine-tuning scores (average of 5 runs); (2) Mask prediction accuracy on MIMIC-CXR val. set; (3) Vocabulary comparison, number of tokens vs. original number of words in FINDINGS, increase shown as percentage.

	RadNLI accuracy (MedNLI transfer)	Mask prediction accuracy	Avg. # of tokens after tokenization	Vocabulary size
RadNLI baseline [189]	53.30	-	-	-
ClinicalBERT	47.67	39.84	78.98 (+38.15%)	28,996
PubMedBERT	57.71	35.24	63.55 (+11.16%)	28,895
CXR-BERT (after Phase-III)	60.46	77.72	58.07 (+1.59%)	30,522
CXR-BERT (after Phase-III + Joint Training)	65.21	81.58	58.07 (+1.59%)	30,522

MIMIC-CXR radiology reports, with the following label categories: (1) entailment, i.e. the hypothesis can be inferred from the premise; (2) contradiction, i.e. the hypothesis cannot be inferred from the premise; and (3) neutral, i.e. the inference relation is undetermined. RadNLI provides expert-annotated development and test sets (480 examples each), but no official training set. Thus, following [189], MedNLI [239] is used for training, which has 11k samples sourced from MIMIC-III discharge summaries, with equally distributed NLI labels. The language models are fine-tuned up to 20 epochs, and early stopping is done by monitoring accuracy scores on the RadNLI development set. Table 4.4 summarizes the NLI evaluation, masked token prediction, and sub-word tokenization results. Using only MedNLI training samples, the proposed model achieves a good accuracy of 65.21%, and far outperforms fine-tuned ClinicalBERT, PubMedBERT, and the score reported in RadNLI [189]. Another important result is that RadNLI accuracy improves after joint training with images (last row of Table 4.4).

Mask Prediction Accuracy. While mask prediction accuracy does not always translate to downstream application performance, it is an auxiliary metric that captures important aspects of a language model's grasp of a target domain. Top-1 mask prediction accuracy is reported on radiology reports in the MIMIC-CXR validation set (Table 4.4), and the standard masking configuration (15% masking probability) is followed. Despite being trained on closely related data, the CXR-BERT displays a much better mask prediction accuracy compared to ClinicalBERT (trained on MIMIC-III, which includes radiology reports) and PubMedBERT (trained on biomedical literature text). This suggests that radiology text significantly differs from other clinical text or biomedical literature text, highlighting the need for specialized text encoder models.

Ablation. An ablation of the various aspects of CXR-BERT is also conducted, measuring the impact after joint training. Table 4.5 shows that all components of

Table 4.5: CXR-BERT ablation. CNR and mIoU are macro averages of BioViL performance on all categories of `MS-CXR`. *Syn. sim.* denotes the average cosine similarity between RadNLI entailments. *Cont. gap* is the average similarity gap of RadNLI entailment and contradiction pairs. CXR-BERT is the combination of all components below the first row.

	RadNLI		Grounding	
Model or pretraining stage	Syn. sim.	Cont. gap	mIoU	CNR
ClinicalBERT	.657	.609	.182	0.791
Pretrain & Vocab (I–II)	.749	.646	.194	0.796
+ MLM loss added to joint training	.871	.745	.209	0.860
+ Use of attention drop-out (III)	.893	.802	.217	0.945
+ RSM Pretrain (III)	.877	.779	.220	1.012
+ Sentence shuffling (CXR-BERT)	.884	.798	.220	1.031

CXR-BERT contribute to improved downstream and NLI performance, both in terms of alignment between related sentences (entailments) and of discrimination of contradictions. In particular, note the substantial improvement on these scores due to keeping the MLM objective during joint fine-tuning.

Local Alignment Evaluation – Phrase Grounding

A phrase grounding evaluation of the pretrained BioViL model is performed on the `MS-CXR` dataset. For each image–phrase pair, the image is passed to the CNN image encoder and projected to obtain a grid of image representations $\mathbf{V}$ in the joint space. Similarly, the phrase is embedded via the text encoder and projected to the joint space to obtain $\mathbf{t}$. Cosine similarity between $\mathbf{t}$ and elements of $\mathbf{V}$ produces a similarity grid, which is evaluated against the ground-truth bounding boxes. Table 4.6 shows the superior phrase grounding results achieved by BioViL across radiological findings and further shows that the addition of local losses as in the BioViL-L can improve phrase grounding performance for almost all findings. Moreover, the ablation in Table 4.5 demonstrates that there are clear gains to be had in visual grounding performance by improving the text model.

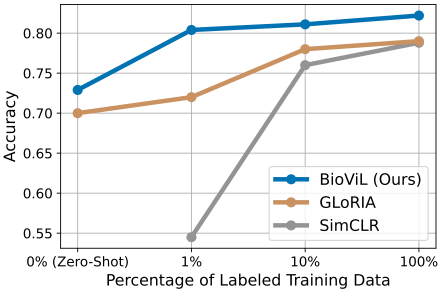
Global Alignment Evaluation – Zero-shot & Linear Probing

To measure global alignment quality, the joint models are also benchmarked on zero-/few-shot binary pneumonia classification problems (image-level) using the external RSNA dataset [237]. Fine-tuning is done via linear probing, i.e. only a last linear layer is trained. The evaluation is conducted on $\mathcal{D}_{\text{test}} = 9006$ images as in [128]

Table 4.6: Contrast-to-noise ratio (CNR) obtained on the newly released `MS-CXR` dataset, averaged over four runs with different seeds. The results are collected using different text encoder and training objectives (e.g., G&L: Global and local loss).

Method	Objective	Text encoder	Atelectasis	Cardiomegaly	Consolidation	Lung opacity	Edema	Pneumonia	Pneumothorax	Pl. effusion	Avg.
Baseline	Global	ClinicalBERT	0.70±.03	0.53±.04	1.15±.07	0.75±.12	0.83±.04	0.85±.09	0.29±.01	1.05±.05	0.769±.02
Baseline	Global	PubMedBERT	0.72±.08	0.64±.05	1.22±.07	0.69±.07	0.80±.04	0.91±.09	0.21±.07	0.99±.03	0.773±.05
ConVIRT [292]	Global	ClinicalBERT	0.86±.04	0.64±.06	1.25±.06	0.78±.07	0.68±.07	1.03±.05	0.28±.08	1.02±.03	0.818±.01
GLoRIA [128]	G&L	ClinicalBERT	0.98±.04	0.53±.31	1.38±.03	1.05±.04	0.66±.03	1.18±.04	0.47±.02	1.20±.04	0.930±.03
BioViL	Global	CXR-BERT	1.02±.06	0.63±.08	1.42±.02	1.05±.06	0.93±.03	1.27±.04	0.48±.06	1.40±.06	1.027±.02
BioViL-L	G&L	CXR-BERT	1.17±.04	0.95±.21	1.45±.03	1.19±.05	0.96±.05	1.19±.01	0.74±.05	1.50±.03	1.142±.04

Table 4.7: RSNA Pneumonia zero-shot and fine-tuned classification. Results are compared to GLoRIA scores reported in [128] which outperforms ConVIRT [292] (see [128]). Training size: GLoRIA ($N = 186k$, private dataset), BioViL ($N = 146.7k$ of MIMIC-CXR).



Method	Type	Text model	Loss	% of labels	Acc.	F1	AUROC
SimCLR [45]	Image only	-	Global	1%	0.545	0.522	0.701
				10%	0.760	0.639	0.802
				100%	0.788	0.675	0.849
GLoRIA [128]	Joint	ClinicalBERT	Global & local	Zero-shot	0.70	0.58	-
				1%	0.72	0.63	0.861
				10%	0.78	0.63	0.880
				100%	0.79	0.65	0.886
Baseline	Joint	ClinicalBERT	Global	Zero-shot	0.719	0.614	0.812
BioViL	Joint	CXR-BERT	Global	Zero-shot	0.732	0.665	0.831
				1%	0.805	0.723	0.881
				10%	0.812	0.727	0.884
				100%	0.822	0.733	0.891

(30% eval. / 70% train.) using the ground-truth labels of the dataset. Two simple text prompts are defined, representing presence/absence of pneumonia: "Findings suggesting pneumonia" and "No evidence of pneumonia". The image encoders are utilized and fine-tuned as described in Section 4.2.1.

The zero-shot and fine-tuned results in Table 4.7 show that the focus of this work on better text modeling results in improved joint modeling of shared latent information between text-image pairs. Note that, to achieve its superior performance here and in Section 4.2.3, BioViL does not require extensive human expert text-prompt engineering (see Supp. E.1.1 for a sensitivity analysis) as for example conducted in GLoRIA [128], where variations over severity and/or location were created.

Local Alignment Evaluation – Semantic Segmentation

Models are evaluated on an RSNA pneumonia segmentation task, using grid-level image representations in the joint latent space. The same text prompts as in the previous section are used for all models, and ground-truth bounding boxes of the RSNA pneumonia dataset ($|\mathcal{D}_{\text{train}}| = 6634$ and $|\mathcal{D}_{\text{test}}| = 2907$) are used for evaluation. Table 4.8 shows that BioViL significantly reduces the need for dense annotations as compared to similar multi-modal and image-only pretraining approaches, outperforming them when using the same number of labeled data points. Note that the proposed modeling framework BioViL(Fig. 4.8), uses neither a local loss term [128,

Table 4.8: RSNA pneumonia segmentation, showing *Zero-shot* and *linear probing* results. Related work is reproduced in the same experimental setup except for LoVT [194].

Method	% of Labels	Supervision	IoU	Dice	CNR
LoVT [194]	100%	Lin. prob.	-	0.518	-
ConVIRT [292]	-	Zero-shot	0.228	0.348	0.849
GLoRIA [128]	-	Zero-shot	0.245	0.366	1.052
BioViL	-	Zero-shot	0.355	0.496	1.477
SimCLR [45]	5%	Lin. prob.	0.382	0.525	1.722
SimCLR [45]	100%	Lin. prob.	0.427	0.570	1.922
BioViL	5%	Lin. prob.	0.446	0.592	2.077
BioViL	100%	Lin. prob.	0.469	0.614	2.178

194], nor a separate object detection [224] or segmentation network [229]. Further, while Table 4.8 shows results using two simple queries, the experiments show that BioViL continues to outperform related work even when more prompts are used for all models as in [128]. Dice and IoU are computed using the same threshold of 0.6 on predictions scaled between $[0, 1]$.

4.2.4 Discussion

The work in this section shows that weak supervision in the form of paired images and text can be an invaluable source of learning signal as relationships of shared latent entities can be exploited to learn good representations for the individual modalities. Furthermore, it shows that careful attention to text modeling can lead to large benefits for all learned models.

The contributions of this section included the introduction of a pretraining procedure and the public release of a radiology domain-specific language model: CXR-BERT. This model has an improved vocabulary and understanding of radiology sentences, contributing to improved downstream performance for all aspects of VLP approaches, e.g., the superior performance on a radiology natural language inference benchmark. This section also presented BioViL, a simple yet effective baseline for self-supervised multi-modal learning for paired image–text radiology data, with a focus on improved text modeling. The approach displays state-of-the-art performance on a large number of downstream tasks evaluating global and local aspects of the image model, text model, and joint latent space. On zero-shot tasks, the model does not require extensive text-prompt engineering compared to prior work. Notably, it outperforms related work on segmentation without requiring a local loss term or an additional vision model to produce region proposals. In that regard, it is complementary to local contrastive losses, and the combination of the two yields improved phrase grounding performance (Table 4.6).

To support the research community in evaluating fine-grained image–text understanding in the radiology domain, a chest X-ray phrase grounding dataset called `MS-CXR` was also released. It presents a more challenging benchmark for joint image–text understanding compared to existing datasets, requiring reasoning over real-world radiology language and scans to ground findings in the correct image locations.

Limitations of the proposed joint approach include that it does not explicitly deal with false negatives in the contrastive losses. Furthermore, co-occurrence of multiple abnormalities could enable contrastive methods to focus only on a subset to match pairs, e.g. pneumothorax and chest tubes commonly occur together [104]. Among its failure cases (see Supp. E.1.2 for more), experiments revealed cases where the approach struggles with very small structures, likely due to image resolution limits.

Chapter 5

Conclusions

In this thesis, I tackled the labeled data bottleneck by developing approaches that support various aspects of learning with weak supervision. The thesis argued that weak supervision provides alternative pathways for acquiring domain knowledge upon which scalable learning mechanisms can be built to train ML models quickly and efficiently. To this end, I introduced a constrained clustering algorithm for improved data partitioning through small amounts of pairwise feedback, new label models for synthesizing programmatic weak supervision votes into an estimate of the unobserved ground truth, an interactive approach to aid users in discovering good sources of weak supervision, and a multi-modal approach to jointly learn representations for paired image-text biomedical data. The results presented in this thesis show that these new weak supervision approaches lead to improved data exploration, improved modeling of unobserved ground truth, and to drastic reductions of user effort. Together, these works provide tools and insights for practitioners and researchers to adopt weak supervision for their ML tasks, in place of having to collect large amounts of manually annotated data. The following sections will summarize the core methodological and open source contributions of this thesis, and finally discuss remaining challenges and open questions.

5.1 Summary of Methodological Contributions

Learning from Pairwise Linkage In Section 2.1, I introduce Kernel Constraint Satisfaction Clustering (KernelCSC) [23], a multiple kernel learning based approach for clustering data under weak supervision in the form of pairwise linkages between samples. The algorithm learns to minimize constraint violation without relaxing the pairwise constraints to distances. Experiments on over 140 benchmark datasets demonstrate that the approach generalizes to unseen pairwise links better than related approaches. I also demonstrate that the approach scales to large datasets through the use of kernel approximations. In Section 2.2 I proposed the introduction of *pairwise* Labeling Functions (LFs) to programmatic weak supervision, i.e. using functions that provide noisy pairwise linkages in addition to the commonly used LFs that outputs imperfect labels for samples directly. I introduce a new, scalable label model based on a Neighborhood Evidence (NE) heuristic. Using synthetic and real data, I demonstrate that the approach can lead to improved downstream test set performance with the introduction of just one pairwise LF.

Label Models In Chapter 3, I develop label models for programmatic weak supervision and show that modeling the distribution of inputs in concert with weak labels leads to improved estimates of the unobserved label, as well as improved downstream results. First, Section 3.1 introduces `WeaSEL` [36], a Weakly Supervised End-to-end Learning model, in which a weak supervision label model (teacher) and the end model (student) are trained jointly. In this work, the label model acts as a differentiable encoder. It receives weak supervision votes and featurized samples as inputs, and produces accuracy parameters that are used to synthesize the weak supervision votes into a label estimate. The algorithm is trained to maximize agreement between the label estimate the end model outputs. `WeaSEL` was evaluated on one crowdsourcing and several weak supervision benchmarks and outperformed related work. Section 3.2 introduces Weakly Supervised GAN (WSGAN) [26] a method which fuses generative adversarial networks and programmatic weak supervision models. WSGAN explicitly models a latent discrete variable in the input data, and aligns its estimate of this variable with the label model estimate on samples where weak sources of signal are available. The experiments on multiple weakly supervised image datasets in Section 3.2 show that WSGAN produces superior latent label estimates compared to other weak supervision label models, improves the quality of generated images compared to unconditional networks trained without weak supervision, and provides evidence that WSGAN can be used for data augmentation (via synthetic samples and pseudolabels) for downstream models trained on weak supervision pseudo labels.

Interactive Weak Supervision Section 4.1 introduces Interactive Weak Supervision (IWS) [25], an approach to help domain experts with fast and query-efficient discovery of good sources of weak supervision. In IWS, users provide feedback to the IWS algorithm in order to find useful sources of weak supervision from a pool of generated candidates with arbitrary accuracy and coverage. Experiments with real users, using both text and image datasets, demonstrate competitive test set performance of the downstream end classifier, quickly approaching that of a fully supervised model. The experiments also show that users can provide accurate feedback on automatically generated LFs.

Vision Language Processing Section 4.2, based on work in [27], introduces BioViL, a self-supervised[1] VLP approach for paired biomedical data, and CXR-

[1] See the introduction of Section 4.2 for a short discussion of why both terms–weakly supervised and self-supervised–are used to refer to this learning scenario.

BERT, a Chest X-ray (CXR) domain-specific language model based on BERT [256]. CXR-BERT is built to have an improved, domain-specific vocabulary. Furthermore, is fine-tuned to CXR data using a masked language modeling (MLM) and a novel section-matching pretraining objective, as well as text augmentation and strong regularization. The model improves radiology natural language inference, radiology masked token prediction, as well as downstream VLP task performance. BioViL is a simple but effective joint VLP model for biomedical image-text data that uses uses CXR-BERT as its text encoder. It optimizes a symmetric contrastive loss [205] on global image and text representations, while maintaining language modeling performance via an MLM loss. With this objective, image and text augmentations, and regularization, BioViL achieves new state-of-the-art performance on various downstream tasks in the CXR domain.

5.2 Summary of Open Source Code and Dataset Contributions

Open Source Code

- For the constrained clustering work presented in Section 2.1, a python library is made available containing code for the KernelCSC and MahalanobisCSC[2] algorithms [23]: `https://github.com/autonlab/constrained-clustering`. The code-base allows one to reproduce all results that were presented in this work. Additionally, the library contains extra features such as farthest-first cluster initialization and alternative base unsupervised algorithm components for KernelCSC and MahalanobisCSC.

- Code for the Weakly Supervised End-to-end Learning model `WeaSEL` [36], presented in Section 3.1, is released at `https://github.com/autonlab/weasel`. The python code base is designed to be flexible, so that any PyTorch downstream model can be used with the approach.

- The python libraries for WSGAN [26] are available at `https://github.com/benbo/WSGAN-paper` and `https://github.com/benbo/stylewsgan`. The libraries contain code to train WSGAN based on simple DCGAN architectures, as well as on more advanced StyleGAN2 networks.

- Code for Interactive Weak Supervision (IWS) [25] presented in Section 4.1 is available at `https://github.com/benbo/interactive-weak-supervision`. This python code can be used to reproduce the experiments discussed in this thesis as well as to run an interactive session where a user interacts with IWS.

[2] Which follows the same principles as KernelCSC but learns a Mahalanobis metric.

- A number of artifacts were produced for the biomedical Vision-Language Processing work presented in [27] and covered in Section 4.2. The language and image models were released on HuggingFace: `https://huggingface.co/models?arxiv=arxiv:2204.09817`. Source code for inference with these models is available at `https://aka.ms/biovil-code`. A python notebook demoing phrase grounding can be found at `http://aka.ms/biovil-demo-notebook`

Datasets With the work presented in Section 4.2 on biomedical Vision-Language Processing [27], MS-CXR [28] was released on PhysioNet: `https://doi.org/10.13026/b90j-vb87`. MS-CXR is a dataset for phrase grounding in radiology images. It contains over 1,000 image–sentence pairs of eight radiology findings. In MS-CXR, the image regions were carefully annotated and matched with real natural language descriptions, verified by board-certified radiologists. The dataset is well-balanced in terms of the clinical findings. Unlike existing CXR benchmarks, this challenging phrase grounding benchmark dataset can be used to evaluate joint, local image-text reasoning while requiring real-world language understanding, e.g. to parse domain-specific location references, complex negations, and bias in reporting style.

5.3 Remaining Challenges and Open Questions

While research has made strides to enable the use of weak supervision as a valid path to build ML applications, numerous challenges remain. For many practical applications of weak supervision, to achieve satisfactory downstream performance requires the definition of multiple high-quality sources of weak supervision. This entails training domain experts in the creation of LFs, diligent LF creation iterations, and frequently the use of labeled development sets to gauge performance. This difficulty of constructing sufficient numbers of LFs with good accuracy and coverage is the core motivation for the interactive weak supervision work presented in [25] and for automated weak supervision work such as [254, 227]. However, the generated sources of weak supervision these works operate on are quite simple in nature still. The development of curriculum learning frameworks for programmatic weak supervision is an attractive proposition, to enable domain experts to specify more fine-grained, high accuracy LFs where required.

Furthermore, the difficulty of defining high quality LFs is not equally distributed across modalities. Natural language data is already discretized into an interpretable vocabulary, making the definition of weak supervision rules on text straightforward for many problems. Defining weak supervision sources is much harder for other modalities such as time series and images. Thus, the ease with which LFs can be created for non-text data remains a critical roadblock. To boost the use of programmatic weak supervision outside of text-centric domains, the development of flexible methods that create an interpretable vocabulary of building blocks in an unsupervised fashion for modalities other than text would be extremely impactful.

I explored the pretraining of large models through paired multi-modal biomedical data in Section 4.2, showing that this domain specific pretraining results in new state-of-the-art performance in natural language inference and image tasks such as phrase grounding. A topic of interest that I did not get to explore is the use of large general domain foundation models such as CLIP [215] or GPT3 [31] within other weak supervision paradigms such as Data Programming (DP). The rise and impact of foundation models is not competition for weak supervision, but rather offers opportunities for the fusion of the two paradigms along many axes. Here, while some initial studies exist [227, 44], much work remains to be done to explore the relative strengths and weaknesses of zero-shot and few-shot foundation models and (programmatic) weak supervision paradigms, and how to best use them in concert.

Finally, the progress made in the development of learning paradigms that aim to reduce the reliance of manually annotated ground truth data has largely been siloed. In the previous paragraph, I pointed to the need to investigate the fusion of (programmatic) weak supervision and zero/few-shot foundation models. In Section 3.2, I studied the fusion of programmatic weak supervision and GANs, with a focus on images. However, no flexible frameworks exist that allow users to apply various combinations of learning paradigms simultaneously, and selectively, in order to make use of all the domain knowledge they possess, no matter the format.

5.3.1 A User-Centric and Human-Centric Perspective

The focus of the work presented in this thesis has been primarily model-centric, insofar as the central goal has been to leverage the proposed weak supervision methodologies to improve model performance metrics. Of course, some sections of in this thesis had their core motivation tied to user needs. Interactive Weak Supervision (IWS) [25] presented in Section 4.1, was designed to aid users in discovering good sources of weak labels. The constrained clustering algorithm presented in 2.1 is in part motivated by use cases where users are exploring a dataset and are yet unsure of the underlying classes they want to partition. However, even for these methods the core metrics of success that the evaluations focused on were tied to collected ground truth labels. A focus on improvements achieved by new weak supervision approaches measured via the target variable–which remained unobserved during training–is a sensible choice to assess if the models can provide typical performance benefits targeted in supervised learning in the first place. But it is also important to consider that ML models are being deployed in diverse application scenarios, including ones with significant societal impact. And in some, the algorithms take a supporting role assisting users, e.g. in decision making or knowledge discovery. In such settings, model performance is only one part of the puzzle, and there are other meaningful endpoints that must be considered. This section will discuss the potential for future work to investigate the utility of the research areas explored in this thesis through a user- and human-centric lens.

Future work on weak supervision approaches such as the ones presented in this thesis may consider if they present pathways towards advancing critical goals of human-centric machine learning [158]. For example, this thesis has assumed the presence of ground truth labels. In many domains however, what constitutes “ground truth” may be subject to debate and not directly observable, e.g. when the targets are unobservable theoretical constructs. This results in a gap between the labels that are observed and used to train models, and the construct of interest to humans. The issue is often referred to as construct validity [134]. To give a concrete example, an algorithm that was meant to prioritize patients for risk-management programs relied on cost as a proxy for healthcare needs, and as a result it exhibited racial bias [201]. In child welfare, models meant to assist call workers in identifying which hotline calls should trigger a social worker investigation often predict the risk of out-of-home placement, but it has been shown that this fails to capture important dimensions of risks which are considered by the human experts in this domain [7]. Weak supervision provides a pathway to potentially capture constructs of interest to experts, circumventing the need to rely on observed labels. A potential fruitful direction for future work is to explore the use of the methodologies proposed in this thesis to mitigate issues of construct validity, by shaping the target label through weak supervision sources designed by domain experts.

Another issue that can be connected to human-centric machine learning is that of model monitoring, the goals of which have been described as ensuring “[. . .]that models are making accurate predictions, are robust to shifts in the data, are not relying on spurious features, and are not unduly discriminating against minority groups” [236, p.173]. First, the issue of distributions that shift during production is one where the use of weak supervision is an attractive proposition to circumvent manual annotation of new, shifted data. Second, as the relationships between input and target variables may change over time, programmatic weak supervision is a promising paradigm as it can considerably speed up the process of redefining labels and relabeling data. Here, IWS [25] may provide further time savings and performance benefits as it aids users in the discovery of new LFs. Furthermore, if a programmatic weak supervision framework is adopted instead of a supervised one, it offers the ability to not just monitor the distribution of features, but also the distribution of LF outputs.

Of course, in a human-centric context, future work should also investigate if and when learning from user defined weak supervision sources and methods as proposed in this thesis can mitigate fairness concerns, or in what cases it may risk reproducing the biases of the users, potentially exacerbating and compounding them. To this end, weak supervision with fairness constraints, e.g. by only accepting LFs that hold equal predictive power across different demographic groups, is a concrete direction that could be explored.

Future work should also investigate the utility of weak supervision work, such as the methods presented in this thesis, for user-centric machine learning. By user-centric machine learning, I refer to work that aims to assist users in their needs and goals related to the data they are modeling. Adopting weak supervision paradigms in lieu of manual data annotation has obvious usability-related benefits, such as the efficiency and scalability aspects that weak supervision offers. However, here I want to emphasize user-centric scenarios where the user takes an active role in building and using models to address their needs, e.g. to explore and understand a domain, to support decision making, or to collect data.

Weak supervision may play an important role for user-centric ML as it can provide pathways to capture user knowledge and intuition. Future work should investigate if the trained models indeed align better with human intuition about a target construct than alternative ways of data collection for training such as manual labeling or using observed historical outcomes of proxy measures. In relation to user-centric design and decision support, a growing body of research is showing that model performance is only one piece of the puzzle, and *appropriate reliance* is essential to algorithms enabling better decision making [6]. For example, the types of errors that a model makes can influence if users trust it. It is worthwhile to explore if models trained with user-defined weak supervision sources can lead to improved human-AI collaboration, since target labels may be shaped through user defined weak supervision sources. The process of defining LFs in order to annotate a dataset programmatically can be viewed as a vehicle to incorporate high level, conceptual feedback into the data labeling process. Thus, the programmatic weak supervision methods presented in this thesis may be used to train models that align well with users' intuition about a target variable as captured by the weak supervision sources they design, which in turn could lead to more trust in a model's prediction once deployed. On the flip side, there is a risk that the alignment between user and model objectives could induce over-reliance if the model always confirms the human intuitions (even when the human is wrong), and it could reduce the potential for complementarity.

A specific decision support paradigm that future work may explore is that of case-based decision support. In this paradigm, algorithms are used to produce examples that help users make sense of predicted labels to aid them in decision-making. For example, nearest neighbors in a representation space learned via supervised learning may be shown to a user, see e.g. [171]. However, such representations learned by supervised models may not align well with human intuition [171]. Weak forms of supervision may offer a valuable alternative to supervised frameworks in this case if the sources of weak supervision are designed with the help of the users. The constrained clustering and metric learning work presented in Section 2.1 can form an excellent basis for such case-based decision support. Here, the algorithm can learn from a users intuition about group membership, expressed through pairwise constraints, and do so even when the underlying partitions are still unclear to a user. Further, the programmatic weak supervision work presented in Section 2.2, which combines intuition about the latent classes by using pairwise weak sources of labels as well as the traditional LFs, presents an opportunity to study if such rich and varying sources of weak supervision can be combined to produce representations that align even better with users' understanding of similarity.

Two more user-centric areas of interested concern knowledge discovery and data collection. For example, the work presented in Section 2.1 is a good candidate for scenarios where users aim find new data/documents related to an area of interest. Here, the algorithm only requires users to specify documents they consider to belong to the same group or different groups, even if they are as yet unable to exactly articulate the underlying groups they are targeting (as discussed in [56]). Programmatic weak supervision work such as the one presented in Section 2.2 may also be a great fit in this context, as it is easier to adjust user-defined LFs when a user's understanding of the underlying classes evolves, compared to having to correct labels previously assigned through manual annotation.

Bibliography

[1] Amreen Abbas, Sweta Jain, Mahesh Gour, and Swetha Vankudothu. "Tomato plant disease detection using transfer learning with C-GAN synthetic images". In: *Computers and Electronics in Agriculture* 187 (2021), p. 106279.

[2] Hassan Akbari, Svebor Karaman, Surabhi Bhargava, Brian Chen, Carl Vondrick, and Shih-Fu Chang. "Multi-level multimodal common semantic space for image-phrase grounding". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 2019, pp. 12476–12486. DOI: `10.1109/CVPR.2019.01276`.

[3] S. Albarqouni, C. Baur, F. Achilles, V. Belagiannis, S. Demirci, and N. Navab. "AggNet: Deep Learning From Crowds for Mitosis Detection in Breast Cancer Histology Images". In: *IEEE Transactions on Medical Imaging* 35.5 (2016), pp. 1313–1321.

[4] Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. "Publicly Available Clinical BERT Embeddings". In: *Proceedings of the 2nd Clinical Natural Language Processing Workshop*. Minneapolis, Minnesota, USA: Association for Computational Linguistics, 2019, pp. 72–78. DOI: `10.18653/v1/W19-1909`.

[5] Saket Anand, Sushil Mittal, Oncel Tuzel, and Peter Meer. "Semi-supervised kernel mean shift clustering". In: *IEEE transactions on pattern analysis and machine intelligence* 36.6 (2013), pp. 1201–1215.

[6] Maria De-Arteaga, Riccardo Fogliato, and Alexandra Chouldechova. "A Case for Humans-in-the-Loop: Decisions in the Presence of Erroneous Algorithmic Scores". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Honolulu, HI, USA: Association for Computing Machinery, 2020, pp. 1–12. ISBN: 9781450367080. DOI: `10.1145/3313831.3376638`.

[7] Maria De-Arteaga, Vincent Jeanselme, Artur Dubrawski, and Alexandra Chouldechova. "Leveraging expert consistency to improve algorithmic decision support". In: *arXiv preprint arXiv:2101.09648* (2021).

[8] Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. "Bias in bios: A case study of semantic representation bias in a high-stakes setting". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency.* 2019, pp. 120–128.

[9] Hassan Ashtiani, Shai Ben-David, Nicholas JA Harvey, Christopher Liaw, Abbas Mehrabian, and Yaniv Plan. "Nearly tight sample complexity bounds for learning mixtures of Gaussians via sample compression schemes". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems.* 2018, pp. 3416–3425.

[10] Stephen H Bach, Bryan He, Alexander Ratner, and Christopher Ré. "Learning the structure of generative models without labeled data". In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70.* 2017, pp. 273–282.

[11] Stephen H Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alex Ratner, Braden Hancock, Houman Alborzi, et al. "Snorkel drybell: A case study in deploying weak supervision at industrial scale". In: *Proceedings of the 2019 International Conference on Management of Data.* 2019, pp. 362–375.

[12] Liang Bai, JiYe Liang, and Fuyuan Cao. "Semi-Supervised Clustering with Constraints of Different Types from Multiple Information Sources". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[13] Shruthi Bannur, Stephanie Hyland, Qianchu Liu, Fernando Perez-Garcia, Maximilian Ilse, Daniel C Castro, Benedikt Boecking, Harshita Sharma, Kenza Bouzid, Anja Thieme, et al. "Learning to Exploit Temporal Structure for Biomedical Vision-Language Processing". In: *arXiv preprint arXiv:2301.04558* (2023).

[14] Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. "Learning distance functions using equivalence relations". In: *International Conference on Machine Learning.* 2003, pp. 11–18.

[15] Matthew Barnes. "Learning with Clusters". PhD thesis. Carnegie Mellon University, 2019.

[16] Sugato Basu, Arindam Banerjee, and R. Mooney. "Semi-supervised Clustering by Seeding". In: *International Conference on Machine Learning.* 2002.

[17] Sugato Basu, Mikhail Bilenko, Arindam Banerjee, and Raymond J Mooney. "Probabilistic semi-supervised clustering with constraints". In: *Semi-supervised learning* (2006), pp. 71–98.

[18] Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney. "A Probabilistic Framework for Semi-supervised Clustering". In: *SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2004.

[19] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. "The power of ensembles for active learning in image classification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9368–9377.

[20] Ricardo Bigolin Lanfredi, Mingyuan Zhang, William F Auffermann, Jessica Chan, Phuong-Anh T Duong, Vivek Srikumar, Trafton Drew, Joyce D Schroeder, and Tolga Tasdizen. "REFLACX, a dataset of reports and eye-tracking data for localization of abnormalities in chest x-rays". In: *Scientific data* 9.1 (2022), p. 350.

[21] Mikhail Bilenko, Sugato Basu, and Raymond J. Mooney. "Integrating Constraints and Metric Learning in Semi-supervised Clustering". In: *International Conference on Machine Learning*. 2004.

[22] Benedikt Boecking and Artur Dubrawski. "Pairwise Feedback for Data Programming". In: *NeurIPS Workshop on Learning with Rich Experience (LIRE)* (2019).

[23] Boecking, Benedikt, Vincent Jeanselme, and Artur Dubrawski. "Constrained clustering and multiple kernel learning without pairwise constraint relaxation". In: *Advances in Data Analysis and Classification* (June 2022).

[24] Boecking, Benedikt, Kyle Miller, Emily Kennedy, and Artur Dubrawski. "Quantifying the relationship between large public events and escort advertising behavior". In: *Journal of human trafficking* 5.3 (2019), pp. 220–237.

[25] Boecking, Benedikt, Willie Neiswanger, Eric Xing, and Artur Dubrawski. "Interactive Weak Supervision: Learning Useful Heuristics for Data Labeling". In: *International Conference on Learning Representations (ICLR)*. 2021.

[26] Boecking, Benedikt, Nicholas Roberts, Willie Neiswanger, Stefano Ermon, Frederic Sala, and Artur Dubrawski. "Generative Modeling Helps Weak Supervision (and Vice Versa)". In: *International Conference on Learning Representations (ICLR)*. 2023.

[27] Boecking, Benedikt et al. "Making the Most of Text Semantics to Improve Biomedical Vision–Language Processing". In: *European Conference on Computer Vision (ECCV)*. 2022.

[28] Boecking, Benedikt et al. "MS-CXR: Making the Most of Text Semantics to Improve Biomedical Vision–Language Processing". In: *PhysioNet* (2022).

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

[30] Andrew Brock, Jeff Donahue, and Karen Simonyan. "Large Scale GAN Training for High Fidelity Natural Image Synthesis". In: *International Conference on Learning Representations*. 2019.

[31] 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: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[32] Brent Bryan, Robert C Nichol, Christopher R Genovese, Jeff Schneider, Christopher J Miller, and Larry Wasserman. "Active learning for identifying function threshold boundaries". In: *Advances in Neural Information Processing Systems*. 2006, pp. 163–170.

[33] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. "Understanding disentangling in β-VAE". In: *arXiv preprint arXiv:1804.03599* (2018).

[34] Aurelia Bustos, Antonio Pertusa, Jose-Maria Salinas, and Maria de la Iglesia-Vayá. "PadChest: A large chest X-ray image dataset with multi-label annotated reports". In: *Medical image analysis* 66 (2020), p. 101797.

[35] Salva Rühling Cachay, Benedikt Boecking, and Artur Dubrawski. "Dependency Structure Misspecification in Multi-Source Weak Supervision Models". In: *ICLR Workshop on Weakly Supervised Learning (WeaSuL)* (2021). arXiv: 2106.10302. URL: https://arxiv.org/abs/2106.10302.

[36] Salva Rühling Cachay, Boecking, Benedikt, and Artur Dubrawski. "End-to-End Weak Supervision". In: *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS)*. 2021.

[37] Peng Cao, Yilun Xu, Yuqing Kong, and Yizhou Wang. "Max-MIG: an Information Theoretic Approach for Joint Learning from Crowds". In: *International Conference on Learning Representations* (2019).

[38] Arlene Casey, Emma Davidson, Michael Poon, Hang Dong, Daniel Duma, Andreas Grivas, Claire Grover, Víctor Suárez-Paniagua, Richard Tobin, William Whiteley, et al. "A systematic review of natural language processing applied to radiology reports". In: *BMC medical informatics and decision making* 21.1 (2021), pp. 1–18.

[39] Oishik Chatterjee, Ganesh Ramakrishnan, and Sunita Sarawagi. "Robust data programming with precision-guided labeling functions". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 04. 2020, pp. 3397–3404.

[40] Geeticka Chauhan, Ruizhi Liao, William Wells, Jacob Andreas, Xin Wang, Seth Berkowitz, Steven Horng, Peter Szolovits, and Polina Golland. "Joint modeling of chest radiographs and radiology reports for pulmonary edema assessment". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer. 2020, pp. 529–539.

[41] Junxiang Chen and Kayhan Batmanghelich. "Weakly supervised disentanglement by pairwise similarities". In: *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 34. 2020, pp. 3495–3502.

[42] Mayee Chen, Benjamin Cohen-Wang, Stephen Mussmann, Frederic Sala, and Christopher Ré. "Comparing the Value of Labeled and Unlabeled Data in Method-of-Moments Latent Variable Estimation". In: *International Conference on Artificial Intelligence and Statistics.* PMLR. 2021, pp. 3286–3294.

[43] Mayee F Chen, Daniel Y Fu, Frederic Sala, Sen Wu, Ravi Teja Mullapudi, Fait Poms, Kayvon Fatahalian, and Christopher Ré. "Train and You'll Miss It: Interactive Model Iteration with Weak Supervision and Pre-Trained Embeddings". In: *arXiv preprint arXiv:2006.15168* (2020).

[44] Mayee F Chen, Daniel Yang Fu, Dyah Adila, Michael Zhang, Frederic Sala, Kayvon Fatahalian, and Christopher Re. "Shoring Up the Foundations: Fusing Model Embeddings and Weak Supervision". In: *The 38th Conference on Uncertainty in Artificial Intelligence.* 2022.

[45] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A Simple Framework for Contrastive Learning of Visual Representations". In: *Proceedings of the 37th International Conference on Machine Learning.* Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 1597–1607.

[46] Vincent Chen, Sen Wu, Alexander J Ratner, Jen Weng, and Christopher Ré. "Slice-based learning: A programming model for residual learning in critical data slices". In: *Advances in Neural Information Processing Systems.* 2019, pp. 9392–9402.

[47] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. "InfoGAN: interpretable representation learning by information maximizing Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems.* 2016.

[48] Xinlei Chen and Abhinav Gupta. "Webly supervised learning of convolutional networks". In: *Proceedings of the IEEE International Conference on Computer Vision.* 2015, pp. 1431–1439.

[49] Xinlei Chen and Kaiming He. "Exploring simple siamese representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2021, pp. 15750–15758.

[50] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. "UNITER: Universal image-text representation learning". In: *European conference on computer vision*. Springer. 2020, pp. 104–120.

[51] Zhihong Chen, Yan Song, Tsung-Hui Chang, and Xiang Wan. "Generating Radiology Reports via Memory-driven Transformer". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Nov. 2020. DOI: `10.18653/v1/2020.emnlp-main.112`.

[52] Sasank Chilamkurthy, Rohit Ghosh, Swetha Tanamala, Mustafa Biviji, Norbert G Campeau, Vasantha Kumar Venugopal, Vidur Mahajan, Pooja Rao, and Prashant Warier. "Deep learning algorithms for detection of critical findings in head CT scans: a retrospective study". In: *The Lancet* 392.10162 (2018), pp. 2388–2396.

[53] Kashyap Chitta, Jose M Alvarez, and Adam Lesnikowski. "Large-scale visual active learning with deep probabilistic ensembles". In: *arXiv preprint arXiv:1811.03575* (2018).

[54] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. "Weakly supervised object localization with multi-fold multiple instance learning". In: *IEEE transactions on pattern analysis and machine intelligence* 39.1 (2016), pp. 189–203.

[55] Benjamin Cohen-Wang, Stephen Mussmann, Alex Ratner, and Chris Ré. "Interactive Programmatic Labeling for Weak Supervision". In: *KDD Data Collection, Curation, and Labeling for Mining and Learning Workshop* (2019).

[56] David Cohn, Rich Caruana, and Andrew McCallum. "Semi-supervised clustering with user feedback". In: *Constrained Clustering: Advances in Algorithms, Theory, and Applications* 4.1 (2003), pp. 17–32.

[57] David Corney, Dyaa Albakour, Miguel Martinez-Alvarez, and Samir Moussa. "What do a million news articles look like?" In: *Workshop on Recent Trends in News Information Retrieval* (2016), pp. 42–47.

[58] Danilo Croce, Alessandro Moschitti, and Roberto Basili. "Structured lexical similarity via convolution kernels on dependency trees". In: *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2011, pp. 1034–1046.

[59] William R Crum, Oscar Camara, and Derek LG Hill. "Generalized overlap measures for evaluation and validation in medical image analysis". In: *IEEE transactions on medical imaging* 25.11 (2006), pp. 1451–1461.

[60] Marco Cuturi. "Fast global alignment kernels". In: *International Conference on Machine Learning*. 2011, pp. 929–936.

[61] Songtai Dai, Quan Wang, Yajuan Lyu, and Yong Zhu. “BDKG at MEDIQA 2021: System Report for the Radiology Report Summarization Task”. In: *Proceedings of the 20th Workshop on Biomedical Language Processing*. Association for Computational Linguistics, 2021, pp. 103–111. DOI: 10.18653/v1/2021.bionlp-1.11.

[62] Nilesh Dalvi, Anirban Dasgupta, Ravi Kumar, and Vibhor Rastogi. “Aggregating crowdsourced binary ratings”. In: *Proceedings of the 22nd International Conference on World Wide Web*. 2013, pp. 285–294.

[63] Nilaksh Das, Sanya Chaba, Renzhi Wu, Sakshi Gandhi, Duen Horng Chau, and Xu Chu. “Goggles: Automatic image labeling with affinity coding”. In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 2020, pp. 1717–1732.

[64] Sanjoy Dasgupta, Akansha Dey, Nicholas Roberts, and Sivan Sabato. “Learning from discriminative feature feedback”. In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. 2018, pp. 3959–3967.

[65] Samyak Datta, Karan Sikka, Anirban Roy, Karuna Ahuja, Devi Parikh, and Ajay Divakaran. “Align2Ground: Weakly supervised phrase grounding guided by image-caption alignment”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 2019, pp. 2601–2610. DOI: 10.1109/ICCV.2019.00269.

[66] Surabhi Datta and Kirk Roberts. “A hybrid deep learning approach for spatial trigger extraction from radiology reports”. In: *Proceedings of the Third International Workshop on Spatial Language Understanding*. Vol. 2020. Association for Computational Linguistics, 2020, pp. 50–55. DOI: 10.18653/v1/2020.splu-1.6.

[67] Surabhi Datta, Yuqi Si, Laritza Rodriguez, Sonya E Shooshan, Dina Demner-Fushman, and Kirk Roberts. “Understanding spatial language in radiology: Representation framework, annotation, and spatial relation extraction from chest X-ray reports using deep learning”. In: *Journal of biomedical informatics* 108 (2020), p. 103473.

[68] Hal Daumé and D. Marcu. “A Bayesian model for supervised clustering with the Dirichlet process prior”. In: *Journal of Machine Learning Research* 6 (2006), pp. 1551–1551.

[69] Ian Davidson and SS Ravi. “The complexity of non-hierarchical clustering with instance and cluster level constraints”. In: *Data mining and knowledge discovery* 14.1 (2007), pp. 25–61.

[70] Ian Davidson, Kiri L Wagstaff, and Sugato Basu. "Measuring constraint-set utility for partitional clustering algorithms". In: *European conference on principles of data mining and knowledge discovery.* Springer. 2006, pp. 115–126.

[71] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. "Information-theoretic Metric Learning". In: *International Conference on Machine Learning.* Corvalis, Oregon, USA, 2007, pp. 209–216.

[72] Alexander Philip Dawid and Allan M Skene. "Maximum likelihood estimation of observer error-rates using the EM algorithm". In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 28.1 (1979), pp. 20–28.

[73] Mostafa Dehghani, Arash Mehrjou, Stephan Gouws, Jaap Kamps, and Bernhard Schölkopf. "Fidelity-Weighted Learning". In: *Proceedings of the International Conference on Learning Representations.* 2018.

[74] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. "Neural Ranking Models with Weak Supervision". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval.* Association for Computing Machinery, 2017, pp. 65–74.

[75] Dina Demner-Fushman, Marc D Kohli, Marc B Rosenman, Sonya E Shooshan, Laritza Rodriguez, Sameer Antani, George R Thoma, and Clement J McDonald. "Preparing a collection of radiology examinations for distribution and retrieval". In: *Journal of the American Medical Informatics Association* 23.2 (2016), pp. 304–310.

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

[77] Emily Denton, Jason Weston, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. "User conditional hashtag prediction for images". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 2015, pp. 1731–1740.

[78] Karan Desai and Justin Johnson. "VirTex: Learning visual representations from textual annotations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2021, pp. 11162–11173.

[79] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).* Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[80] Zijian Ding, Shan Qiu, Yutong Guo, Jianping Lin, Li Sun, Dapeng Fu, Zhen Yang, Chengquan Li, Yang Yu, Long Meng, et al. “LabelECG: A Web-Based Tool for Distributed Electrocardiogram Annotation”. In: *Machine Learning and Medical Engineering for Cardiovascular Health and Intravascular Imaging and Computer Assisted Stenting*. Springer, 2019, pp. 104–111.

[81] Dmitriy Dligach, Steven Bethard, Lee Becker, Timothy Miller, and Guergana K Savova. “Discovering body site and severity modifiers in clinical texts”. In: *Journal of the American Medical Informatics Association* 21.3 (2014), pp. 448–454.

[82] Gregory Druck, Gideon Mann, and Andrew McCallum. “Learning from Labeled Features Using Generalized Expectation Criteria”. In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Singapore, Singapore: ACM, 2008, pp. 595–602. DOI: 10.1145/1390334.1390436.

[83] Artur Dubrawski, Kyle Miller, Matthew Barnes, Boecking, Benedikt, and Emily Kennedy. “Leveraging publicly available data to discern patterns of human-trafficking activity”. In: *Journal of Human Trafficking* 1.1 (2015), pp. 65–85.

[84] John Duchi. “Lecture notes for statistics 311/electrical engineering 377”. In: *URL: https://web.stanford.edu/class/stats311/lecture-notes.pdf. Last visited June 2022* 2 (2022), p. 23.

[85] Jared A. Dunnmon et al. “Cross-Modal Data Programming Enables Rapid Medical Machine Learning”. In: *Patterns* 1.2 (2020), p. 100019.

[86] David Eisenberg, Edward M Marcotte, Ioannis Xenarios, and Todd O Yeates. “Protein function in the post-genomic era”. In: *Nature* 405.6788 (2000), p. 823.

[87] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. “Dermatologist-level classification of skin cancer with deep neural networks”. In: *Nature* 542.7639 (2017), pp. 115–118.

[88] Sabri Eyuboglu, Geoffrey Angus, Bhavik N Patel, Anuj Pareek, Guido Davidzon, Jin Long, Jared Dunnmon, and Matthew P Lungren. “Multi-task weak supervision enables anatomically-resolved abnormality detection in whole-body FDG-PET/CT”. In: *Nature communications* 12.1 (2021), pp. 1–15.

[89] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. “From captions to visual concepts and back”. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. IEEE Computer Society, 2015, pp. 1473–1482. DOI: 10.1109/CVPR.2015.7298754.

[90] Thomas Finley and Thorsten Joachims. "Supervised Clustering with Support Vector Machines". In: *International Conference on Machine Learning*. 2005, pp. 217–224.

[91] Sharon Fogel, Hadar Averbuch-Elor, Daniel Cohen-Or, and Jacob Goldberger. "Clustering-driven deep embedding with pairwise constraints". In: *IEEE computer graphics and applications* 39.4 (2019), pp. 16–27.

[92] Jason A Fries, Paroma Varma, Vincent S Chen, Ke Xiao, Heliodoro Tejeda, Priyanka Saha, Jared Dunnmon, Henry Chubb, Shiraz Maskatia, Madalina Fiterau, et al. "Weakly supervised classification of aortic valve malformations using unlabeled cardiac MRI sequences". In: *Nature Communications* 10.1 (2019), pp. 1–10.

[93] Daniel Y. Fu, Mayee F. Chen, Frederic Sala, Sarah M. Hooper, Kayvon Fatahalian, and Christopher Ré. *Fast and Three-rious: Speeding Up Weak Supervision with Triplet Methods*. 2020. arXiv: `2002.11955 [stat.ML]`.

[94] Aviv Gabbay and Yedid Hoshen. "Demystifying Inter-Class Disentanglement". In: *International Conference on Learning Representations*. 2020.

[95] Tianyu Gao, Xingcheng Yao, and Danqi Chen. "SimCSE: Simple Contrastive Learning of Sentence Embeddings". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 6894–6910.

[96] Roman Garnett, Thomas Gärtner, Martin Vogt, and Jürgen Bajorath. "Introducing the 'active search'method for iterative virtual screening". In: *Journal of Computer-Aided Molecular Design* 29.4 (2015), pp. 305–314.

[97] Alex Gittens and Michael W Mahoney. "Revisiting the Nyström method for improved large-scale machine learning". In: *Journal of Machine Learning Research* 17.1 (2016), pp. 3977–4041.

[98] Garrett B Goh, Charles Siegel, Abhinav Vishnu, and Nathan Hodas. "Using rule-based labels for weak supervised learning: a ChemNet for transferable chemical property prediction". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 302–310.

[99] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals". In: *Circulation* 101.23 (2000), e215–e220.

[100] Mehmet Gönen and Ethem Alpaydın. "Multiple kernel learning algorithms". In: *Journal of Machine Learning Research* 12 (2011), pp. 2211–2268.

[101] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems*. Vol. 27. 2014.

[102] Mononito Goswami, Boecking, Benedikt, and Artur Dubrawski. “Weak Supervision for Affordable Modeling of Electrocardiogram Data”. In: *AMIA Annual Symposium*. 2021.

[103] Alkis Gotovos. “Active learning for level set estimation”. MA thesis. Eidgenössische Technische Hochschule Zürich, Department of Computer Science, 2013.

[104] Benedikt Graf, Arkadiusz Sitek, Amin Katouzian, Yen-Fu Lu, Arun Krishnan, Justin Rafael, Kirstin Small, and Yiting Xie. “Pneumothorax and chest tube classification on chest X-rays for detection of missed pneumothorax”. In: *Machine Learning for Health (ML4H) NeurIPS Workshop: Extended Abstract* (2020). URL: `https://arxiv.org/abs/2011.07353`.

[105] Mary L Gray and Siddharth Suri. *Ghost work: How to stop Silicon Valley from building a new global underclass*. Eamon Dolan Books, 2019.

[106] Jean-Bastien Grill et al. “Bootstrap your own latent: A new approach to self-supervised learning”. In: *Advances in Neural Information Processing Systems* (2020).

[107] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. “Domain-specific language model pretraining for biomedical natural language processing”. In: *ACM Transactions on Computing for Healthcare (HEALTH)* 3.1 (2021), pp. 1–23.

[108] Melody Guan, Varun Gulshan, Andrew Dai, and Geoffrey Hinton. “Who said what: Modeling individual labelers improves classification”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2018.

[109] Mathieu Guillame-Bert and Artur Dubrawski. “Classification of time sequences using graphs of temporal constraints”. In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 4370–4403.

[110] Sonal Gupta and Christopher D Manning. “Improved pattern learning for bootstrapped entity extraction”. In: *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. 2014, pp. 98–108.

[111] Tanmay Gupta, Arash Vahdat, Gal Chechik, Xiaodong Yang, Jan Kautz, and Derek Hoiem. “Contrastive Learning for Weakly Supervised Phrase Grounding”. In: *16th European Conference on Computer Vision (ECCV)*. Springer. 2020, pp. 752–768.

[112] Laleh Haghverdi, Aaron TL Lun, Michael D Morgan, and John C Marioni. “Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors”. In: *Nature biotechnology* 36.5 (2018), p. 421.

[113] Yoni Halpern, Youngduck Choi, Steven Horng, and David Sontag. “Using anchors to estimate clinical state without labeled data”. In: *AMIA Annual Symposium Proceedings*. Vol. 2014. American Medical Informatics Association. 2014, p. 606.

[114] Braden Hancock, Paroma Varma, Stephanie Wang, Martin Bringmann, Percy Liang, and Christopher Ré. “Training Classifiers with Natural Language Explanations”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 2018, pp. 1884–1895.

[115] Awni Y Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H Tison, Codie Bourn, Mintu P Turakhia, and Andrew Y Ng. “Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network”. In: *Nature medicine* 25.1 (2019), pp. 65–69.

[116] Nasir Hayat, Hazem Lashen, and Farah E Shamout. “Multi-Label Generalized Zero Shot Learning for the Classiffcation of Disease in Chest Radiographs”. In: *Machine Learning for Healthcare Conference*. PMLR. 2021, pp. 461–477.

[117] 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, pp. 770–778.

[118] Ruining He and Julian McAuley. “Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering”. In: *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2016, pp. 507–517.

[119] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. “Gans trained by a two time-scale update rule converge to a local nash equilibrium”. In: *Advances in neural information processing systems* 30 (2017).

[120] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. “Towards a definition of disentangled representations”. In: *arXiv preprint arXiv:1812.02230* (2018).

[121] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. “Knowledge-based weak supervision for information extraction of overlapping relations”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics. 2011, pp. 541–550.

[122] Steven CH Hoi, Rong Jin, and Michael R Lyu. "Learning nonparametric kernel matrices from pairwise constraints". In: *International Conference on Machine Learning*. 2007, pp. 361–368.

[123] Sarah Hooper, Michael Wornow, Ying Hang Seah, Peter Kellman, Hui Xue, Frederic Sala, Curtis Langlotz, and Christopher Ré. "Cut out the annotator, keep the cutout: better segmentation with weak supervision". In: *International Conference on Learning Representations (ICLR)* (2021).

[124] Tzu-Ming Harry Hsu, Wei-Hung Weng, Willie Boag, Matthew McDermott, and Peter Szolovits. "Unsupervised multimodal representation learning across medical images and reports". In: *Machine Learning for Health (ML4H) NeurIPS Workshop* (2018). URL: `https://arxiv.org/abs/1811.08615`.

[125] Yen-Chang Hsu and Zsolt Kira. "Neural network-based clustering using pairwise constraints". In: *ICLR Workshop track* (2016).

[126] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. "Learning to cluster in order to transfer across domains and tasks". In: *International Conference on Learning Representations*. 2018.

[127] B. Hu, Y. Tang, E. I. -C. Chang, Y. Fan, M. Lai, and Y. Xu. "Unsupervised Learning for Cell-Level Visual Representation in Histopathology Images With Generative Adversarial Networks". In: *IEEE Journal of Biomedical and Health Informatics* 23.3 (2019), pp. 1316–1328. DOI: `10.1109/JBHI.2018.2852639`.

[128] Shih-Cheng Huang, Liyue Shen, Matthew P Lungren, and Serena Yeung. "GLoRIA: A Multimodal Global-Local Representation Learning Framework for Label-Efficient Medical Image Recognition". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 3942–3951.

[129] Kyle Hundman, Thamme Gowda, Mayank Kejriwal, and Boecking, Benedikt. "Always lurking: understanding and mitigating bias in online human trafficking detection". In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. 2018, pp. 137–143.

[130] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. "Sequential model-based optimization for general algorithm configuration". In: *International conference on learning and intelligent optimization*. Springer. 2011, pp. 507–523.

[131] Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR. 2015, pp. 448–456.

[132] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. "CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison". In: *Thirty-Third AAAI Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 590–597. DOI: 10.1609/aaai.v33i01.3301590.

[133] Hamid Izadinia, Bryan C Russell, Ali Farhadi, Matthew D Hoffman, and Aaron Hertzmann. "Deep classifiers from image tags in the wild". In: *Proceedings of the 2015 Workshop on Community-Organized Multimodal Mining: Opportunities for Novel Solutions*. 2015, pp. 13–18.

[134] Abigail Z. Jacobs and Hanna Wallach. "Measurement and Fairness". In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 375–385. ISBN: 9781450383097. DOI: 10.1145/3442188.3445901.

[135] Shali Jiang, Gustavo Malkomes, Geoff Converse, Alyssa Shofner, Benjamin Moseley, and Roman Garnett. "Efficient nonmyopic active search". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 1714–1723.

[136] A Johnson, T Pollard, SJ Berkowitz, R Mark, and S Horng. *MIMIC-CXR Database (version 2.0.0)*. PhysioNet. 2019.

[137] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. "MIMIC-III, a freely accessible critical care database". In: *Scientific data* 3.1 (2016), pp. 1–9.

[138] Armand Joulin, Laurens Van Der Maaten, Allan Jabri, and Nicolas Vasilache. "Learning visual features from large weakly supervised data". In: *European Conference on Computer Vision*. Springer. 2016, pp. 67–84.

[139] Takuhiro Kaneko, Yoshitaka Ushiku, and Tatsuya Harada. "Label-noise robust generative adversarial networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2467–2476.

[140] Giannis Karamanolakis, Subhabrata Mukherjee, Guoqing Zheng, and Ahmed Hassan. "Self-Training with Weak Supervision". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2021, pp. 845–863.

[141] David R Karger, Sewoong Oh, and Devavrat Shah. "Iterative learning for reliable crowdsourcing systems". In: *Advances in Neural Information Processing Systems*. 2011, pp. 1953–1961.

[142] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. "Training generative adversarial networks with limited data". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 12104–12114.

[143] Tero Karras, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4401–4410.

[144] Ashish Khetan, Zachary C Lipton, and Anima Anandkumar. "Learning from noisy singly-labeled data". In: *International Conference on Learning Representations* (2018).

[145] Valentin Khrulkov, Leyla Mirvakhabova, Ivan Oseledets, and Artem Babenko. "Disentangled Representations from Non-Disentangled Models". In: *arXiv preprint arXiv:2102.06204* (2021).

[146] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[147] Dan Klein, Sepandar D Kamvar, and Christopher D Manning. *From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering*. Tech. rep. Stanford, 2002.

[148] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[149] Chayakrit Krittanawong, HongJu Zhang, Zhen Wang, Mehmet Aydar, and Takeshi Kitai. "Artificial intelligence in precision cardiovascular medicine". In: *Journal of the American College of Cardiology* 69.21 (2017), pp. 2657–2664.

[150] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto, Toronto, Ontario, 2009.

[151] Alex Krizhevsky. "One weird trick for parallelizing convolutional neural networks". In: *arXiv preprint arXiv:1404.5997* (2014).

[152] Mark-A Krogel and Tobias Scheffer. "Multi-relational learning, text mining, and semi-supervised learning for functional genomics". In: *Machine Learning* 57.1 (2004), pp. 61–81.

[153] Brian Kulis, Sugato Basu, Inderjit Dhillon, and Raymond Mooney. "Semi-supervised graph clustering: a kernel approach". In: *Machine Learning* 74.1 (Jan. 2009), pp. 1–22.

[154] Hunter Lang and Hoifung Poon. "Self-supervised self-supervision by combining deep learning and probabilistic logic". In: *In Proceedings of the Thirty Fifth Annual Meeting of the Association for the Advancement of Artificial Intelligence (AAAI)*. 2021.

[155] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4 (1989), pp. 541–551.

[156] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[157] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* 36.4 (2020), pp. 1234–1240.

[158] Emmanuel Letouzé and Alex Pentland. "Towards a human artificial intelligence for human development". In: *International Telecommunications Union (ITU) Journal: ICT Discoveries, Special Issue* 2 (2018), pp. 1–8.

[159] Ang Li, Allan Jabri, Armand Joulin, and Laurens Van Der Maaten. "Learning visual n-grams from web data". In: *IEEE International Conference on Computer Vision (ICCV)*. Venice, Italy, Oct. 2017, pp. 4183–4192. DOI: `10.1109/ICCV.2017.449`.

[160] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. "Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 11336–11344.

[161] Hongwei Li, Bin Yu, and Dengyong Zhou. "Error rate analysis of labeling by crowdsourcing". In: *ICML Workshop: Machine Learning Meets Crowdsourcing. Atalanta, Georgia, USA*. Citeseer. 2013.

[162] Li-Jia Li and Li Fei-Fei. "Optimol: automatic online picture collection via incremental model learning". In: *International journal of computer vision* 88.2 (2010), pp. 147–168.

[163] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. "VisualBERT: A simple and performant baseline for vision and language". In: *arXiv preprint arXiv:1908.03557* (2019).

[164] Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. "Supervision Exists Everywhere: A Data Efficient Contrastive Language-Image Pre-training Paradigm". In: *International Conference on Learning Representations*. 2022.

[165] Yikuan Li, Hanyin Wang, and Yuan Luo. "A comparison of pre-trained vision-and-language models for multimodal representation learning across medical images and reports". In: *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE. 2020, pp. 1999–2004.

[166] Ruizhi Liao, Daniel Moyer, Miriam Cha, Keegan Quigley, Seth Berkowitz, Steven Horng, Polina Golland, and William M Wells. "Multimodal Representation Learning via Maximization of Local Mutual Information". In: *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)* (2021).

[167] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Pranav Patel. "Finding motifs in time series". In: *Proc. of the 2nd Workshop on Temporal Data Mining.* 2002, pp. 53–68.

[168] 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.* Springer. 2014, pp. 740–755.

[169] Zinan Lin, Kiran Thekumparampil, Giulia Fanti, and Sewoong Oh. "Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans". In: *International Conference on Machine Learning.* PMLR. 2020, pp. 6127–6139.

[170] Guanxiong Liu, Tzu-Ming Harry Hsu, Matthew McDermott, Willie Boag, Wei-Hung Weng, Peter Szolovits, and Marzyeh Ghassemi. "Clinically accurate chest X-ray report generation". In: *Machine Learning for Healthcare Conference.* PMLR. 2019, pp. 249–269.

[171] Han Liu, Yizhou Tian, Chacha Chen, Shi Feng, Yuxin Chen, and Chenhao Tan. "Towards Effective Case-Based Decision Support with Human-Compatible Representations". In: *ICML Workshop on Human-Machine Collaboration and Teaming* (2022).

[172] Hongfu Liu, Zhiqiang Tao, and Yun Fu. "Partition level constrained clustering". In: *IEEE transactions on pattern analysis and machine intelligence* 40.10 (2017), pp. 2469–2483.

[173] Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. "Adversarial training for large neural language models". In: *arXiv preprint arXiv:2004.08994* (2020).

[174] 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". In: *arXiv preprint arXiv:1907.11692* (2019).

[175] Yongfei Liu, Bo Wan, Lin Ma, and Xuming He. "Relation-aware Instance Refinement for Weakly Supervised Visual Grounding". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2021, pp. 5612–5621.

[176] Francesco Locatello, Gabriele Abbati, Thomas Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. "On the fairness of disentangled representations". In: *Advances in neural information processing systems* 32 (2019).

[177] Lajanugen Logeswaran and Honglak Lee. "An efficient framework for learning sentence representations". In: *6th International Conference on Learning Representations (ICLR)*. Vancouver, BC, Canada, Apr. 2018.

[178] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2018.

[179] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. "ViLBERT: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vancouver, BC, Canada, 2019, pp. 13–23.

[180] Mario Lučić, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. "High-fidelity image generation with fewer labels". In: *International conference on machine learning*. PMLR. 2019, pp. 4183–4192.

[181] Gabor Lugosi. "Learning with an unreliable teacher". In: *Pattern Recognition* 25.1 (1992), pp. 79–87.

[182] Andrew L. 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*. June 2011, pp. 142–150.

[183] 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)*. 2018, pp. 181–196.

[184] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. "Generation and comprehension of unambiguous object descriptions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 11–20. DOI: `10.1109/CVPR.2016.9`.

[185] Alessio Mazzetto, Cyrus Cousins, Dylan Sam, Stephen H Bach, and Eli Upfal. "Adversarial Multi Class Learning under Weak Supervision with Performance Guarantees". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 7534–7543.

[186] Alessio Mazzetto, Dylan Sam, Andrew Park, Eli Upfal, and Stephen Bach. "Semi-supervised aggregation of dependent weak supervision sources with performance guarantees". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 3196–3204.

[187] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. “Distant supervision for relation extraction without labeled data”. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. 2009, pp. 1003–1011.

[188] Mehdi Mirza and Simon Osindero. “Conditional generative adversarial nets”. In: *arXiv preprint arXiv:1411.1784* (2014).

[189] Yasuhide Miura, Yuhao Zhang, Emily Tsai, Curtis Langlotz, and Dan Jurafsky. “Improving Factual Completeness and Consistency of Image-to-Text Radiology Report Generation”. In: *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2021, pp. 5288–5304. DOI: 10.18653/v1/2021.naacl-main.416.

[190] Takeru Miyato and Masanori Koyama. “cGANs with Projection Discriminator”. In: *International Conference on Learning Representations*. 2018.

[191] George B Moody and Roger G Mark. “The impact of the MIT-BIH arrhythmia database”. In: *IEEE Engineering in Medicine and Biology Magazine* 20.3 (2001), pp. 45–50.

[192] Saman Motamed, Patrik Rogalla, and Farzad Khalvati. “Data augmentation using Generative Adversarial Networks (GANs) for GAN-based detection of Pneumonia and COVID-19 in chest X-ray images”. In: *Informatics in Medicine Unlocked* 27 (2021), p. 100779. DOI: https://doi.org/10.1016/j.imu.2021.100779.

[193] Zongshen Mu, Siliang Tang, Jie Tan, Qiang Yu, and Yueting Zhuang. “Disentangled Motif-aware Graph Learning for Phrase Grounding”. In: *AAAI Conference on Artificial Intelligence* (2021).

[194] Philip Müller, Georgios Kaissis, Congyu Zou, and Daniel Rueckert. “Joint Learning of Localized Representations from Medical Images and Reports”. In: *17th European Conference on Computer Vision*. Tel Aviv, Israel, Oct. 2022, pp. 685–701.

[195] Kevin P. Murphy. *Machine Learning : A Probabilistic Perspective.* Adaptive Computation and Machine Learning Series. The MIT Press, 2012.

[196] Chirag Nagpal, Kyle Miller, Boecking, Benedikt, and Artur Dubrawski. “An entity resolution approach to isolate instances of human trafficking online”. In: *arXiv preprint arXiv:1509.06659* (2015).

[197] Mona Nashaat, Aindrila Ghosh, James Miller, Shaikh Quader, Chad Marston, and Jean-Francois Puget. “Hybridization of Active Learning and Data Programming for Labeling Large Industrial Datasets”. In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 46–55.

[198] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. “Learning with noisy labels”. In: *Advances in neural information processing systems* 26 (2013), pp. 1196–1204.

[199] Ha Q Nguyen, Khanh Lam, Linh T Le, Hieu H Pham, Dat Q Tran, Dung B Nguyen, Dung D Le, Chi M Pham, Hang TT Tong, Diep H Dinh, et al. “VinDr-CXR: An open dataset of chest X-rays with radiologist’s annotations”. In: *Scientific Data* 9.1 (2022), p. 429.

[200] Weili Nie, Tero Karras, Animesh Garg, Shoubhik Debnath, Anjul Patney, Ankit Patel, and Animashree Anandkumar. “Semi-supervised stylegan for disentanglement learning”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 7360–7369.

[201] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. “Dissecting racial bias in an algorithm used to manage the health of populations”. In: *Science* 366.6464 (2019), pp. 447–453.

[202] Augustus Odena. “Semi-supervised learning with generative adversarial networks”. In: *arXiv preprint arXiv:1606.01583* (2016).

[203] Augustus Odena, Christopher Olah, and Jonathon Shlens. “Conditional image synthesis with auxiliary classifier gans”. In: *International conference on machine learning*. PMLR. 2017, pp. 2642–2651.

[204] Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. “PMLB: a large benchmark suite for machine learning evaluation and comparison”. In: *BioData Mining* 10.1 (Dec. 2017), p. 36.

[205] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. “Representation learning with contrastive predictive coding”. In: *arXiv preprint arXiv:1807.03748* (2018). URL: `https://arxiv.org/abs/1807.03748`.

[206] Devi Parikh and Kristen Grauman. “Relative attributes”. In: *2011 International Conference on Computer Vision*. IEEE. 2011, pp. 503–510.

[207] Damián Pascual, Amir Aminifar, David Atienza, Philippe Ryvlin, and Roger Wattenhofer. “Synthetic epileptic brain activities using GANs”. In: *Machine Learning for Health (ML4H) at NeurIPS* (2019).

[208] Dan Pelleg and Dorit Baras. “K-means with large and noisy constraint sets”. In: *18th European Conference on Machine Learning*. Springer. Warsaw, Poland, Sept. 2007, pp. 674–682.

[209] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. “Moment matching for multi-source domain adaptation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1406–1415.

[210] Darren Plant and Anne Barton. “Machine learning in precision medicine: lessons to learn”. In: *Nature Reviews Rheumatology* 17.1 (2021), pp. 5–6.

[211] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. “Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2641–2649.

[212] Barnabás Póczos, Liang Xiong, Danica J Sutherland, and Jeff Schneider. “Nonparametric kernel estimators for image classification”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012, pp. 2989–2996.

[213] Stefanos Poulis and Sanjoy Dasgupta. “Learning with Feature Feedback: from Theory to Practice”. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Vol. 54. Proceedings of Machine Learning Research. Apr. 2017, pp. 1104–1113.

[214] Konpat Preechakul, Chawan Piansaddhayanon, Burin Naowarat, Tirasan Khandhawit, Sira Sriswasdi, and Ekapol Chuangsuwanich. “Set Prediction in the Latent Space”. In: *Advances in Neural Information Processing Systems* 34 (2021).

[215] 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*. PMLR. 2021, pp. 8748–8763.

[216] Alec Radford, Luke Metz, and Soumith Chintala. “Unsupervised representation learning with deep convolutional generative adversarial networks”. In: *arXiv preprint arXiv:1511.06434* (2015).

[217] Hema Raghavan, Omid Madani, and Rosie Jones. “Active Learning with Feedback on Features and Instances”. In: *Journal of Machine Learning Research* 7 (Dec. 2006), pp. 1655–1686. ISSN: 1532-4435.

[218] Gonzalo Ramos, Christopher Meek, Patrice Simard, Jina Suh, and Soroush Ghorashi. “Interactive machine teaching: a human-centered approach to building machine-learned models”. In: *Human–Computer Interaction* 35.5-6 (Nov. 2020), pp. 413–451.

[219] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. “DenseCLIP: Language-Guided Dense Prediction with Context-Aware Prompting”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 18082–18091.

[220] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. “Snorkel: Rapid training data creation with weak supervision”. In: *The VLDB Journal* 29.2 (2020), pp. 709–730.

[221] Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. "Training complex models with multi-task weak supervision". In: *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 33. 2019, pp. 4763–4771.

[222] Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. "Data Programming: Creating Large Training Sets, Quickly". In: *Advances in Neural Information Processing Systems.* 2016, pp. 3567–3575.

[223] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. "Learning From Crowds". In: *Journal of Machine Learning Research* 11.43 (2010), pp. 1297–1322.

[224] Joseph Redmon and Ali Farhadi. "YOLOv3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767* (2018).

[225] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards real-time object detection with region proposal networks". In: *Advances in Neural Information Processing Systems* 28 (2015), pp. 91–99.

[226] Sebastian Riedel, Limin Yao, and Andrew McCallum. "Modeling relations and their mentions without labeled text". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer. 2010, pp. 148–163.

[227] Nicholas Roberts, Xintong Li, Tzu-Heng Huang, Dyah Adila, Spencer Schoenberg, Cheng-Yu Liu, Lauren Pick, Haotian Ma, Aws Albarghouthi, and Frederic Sala. "AutoWS-Bench-101: Benchmarking Automated Weak Supervision with 100 Labels". In: *Thirty-sixth Conference on Neural Information Processing Systems - Datasets and Benchmarks Track.* 2022.

[228] Filipe Rodrigues and Francisco Pereira. "Deep learning from crowds". In: *Proceedings of the AAAI Conference on Artificial Intelligence.* 2018.

[229] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention.* Springer. 2015, pp. 234–241.

[230] Khaled Saab, Jared Dunnmon, Roger Goldman, Alex Ratner, Hersh Sagreiya, Christopher Ré, and Daniel Rubin. "Doubly weak supervision of deep learning models for head ct". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer. 2019, pp. 811–819.

[231] Khaled Saab, Jared Dunnmon, Christopher Ré, Daniel Rubin, and Christopher Lee-Messer. "Weak supervision as an efficient approach for automated seizure detection in electroencephalography". In: *npj Digital Medicine* 3.1 (2020), pp. 1–12.

[232] H. Sahbi, J. Audibert, and R. Keriven. “Context-Dependent Kernels for Object Classification”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.4 (2011), pp. 699–708.

[233] Frederic Sala et al. “Multi-Resolution Weak Supervision for Sequential Data”. In: *Advances in Neural Information Processing Systems.* Canada, 2019.

[234] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. “Improved techniques for training gans”. In: *Advances in neural information processing systems* 29 (2016), pp. 2234–2242.

[235] Burr Settles. “Closing the Loop: Fast, Interactive Semi-supervised Annotation with Queries on Features and Instances”. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).* Association for Computational Linguistics, 2011.

[236] Murtuza N Shergadwala, Himabindu Lakkaraju, and Krishnaram Kenthapadi. “A Human-Centric Perspective on Model Monitoring”. In: *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing.* Vol. 10. 1. 2022, pp. 173–183.

[237] George Shih, Carol C Wu, Safwan S Halabi, Marc D Kohli, Luciano M Prevedello, Tessa S Cook, Arjun Sharma, Judith K Amorosa, Veronica Arteaga, Maya Galperin-Aizenberg, et al. “Augmenting the national institutes of health chest radiograph dataset with expert annotations of possible pneumonia”. In: *Radiology: Artificial Intelligence* 1.1 (2019), e180041.

[238] Changho Shin, Wunfred Li, Harit Vishwakarma, Nicholas Roberts, and Frederic Sala. “Universalizing Weak Supervision”. In: *International Conference on Learning Representations (ICLR)* (2022).

[239] Chaitanya Shivade. *MedNLI - A natural language inference dataset for the clinical domain.* PhysioNet. Oct. 2019.

[240] PY Simard, D Steinkraus, and JC Platt. “Best practices for convolutional neural networks applied to visual document analysis”. In: *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.* IEEE. 2003, pp. 958–963.

[241] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[242] Akshay Smit, Saahil Jain, Pranav Rajpurkar, Anuj Pareek, Andrew Y Ng, and Matthew Lungren. “Combining Automatic Labelers and Expert Annotations for Accurate Radiology Report Labeling Using BERT”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).* Association for Computational Linguistics, 2020, pp. 1500–1519. DOI: 10.18653/v1/2020.emnlp-main.117.

[243] Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. “Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design”. In: *International Conference on Machine Learning*. Haifa, Israel, 2010, pp. 1015–1022.

[244] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. “Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition”. In: *Neural Networks* 0 (2012). DOI: 10.1016/j.neunet.2012.02.016.

[245] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. “VL-BERT: Pre-training of Generic Visual-Linguistic Representations”. In: *8th International Conference on Learning Representations (ICLR)*. Addis Ababa, Ethiopia, Apr. 2020.

[246] N. Subrahmanya and Y. C. Shin. “Sparse Multiple Kernel Learning for Signal Processing Applications”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.5 (2010), pp. 788–798.

[247] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. “Revisiting unreasonable effectiveness of data in deep learning era”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 843–852.

[248] L.K. Tam, X. Wang, E. Turkbey, K. Lu, Y. Wen, and D. Xu. “Weakly supervised one-stage vision and language disease detection using large scale pneumonia and pneumothorax studies”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2020*. Santa Clara, CA, USA, Mar. 2020.

[249] Kiran K Thekumparampil, Ashish Khetan, Zinan Lin, and Sewoong Oh. “Robustness of conditional GANs to noisy labels”. In: *Advances in Neural Information Processing Systems*. Vol. 31. 2018.

[250] Yuandong Tian, Xinlei Chen, and Surya Ganguli. “Understanding self-supervised learning dynamics without contrastive pairs”. In: *International Conference on Machine Learning* (2021).

[251] Joseph J Titano, Marcus Badgeley, Javin Schefflein, Margaret Pain, Andres Su, Michael Cai, Nathaniel Swinburne, John Zech, Jun Kim, Joshua Bederson, et al. “Automated deep-neural-network surveillance of cranial images for acute neurologic events”. In: *Nature medicine* 24.9 (2018), pp. 1337–1341.

[252] Paroma Varma, Bryan He, Dan Iter, Peng Xu, Rose Yu, Christopher De Sa, and Christopher Ré. “Socratic learning: Augmenting generative models to incorporate latent subsets in training data”. In: *arXiv preprint arXiv:1610.08123* (2016).

[253] Paroma Varma, Bryan D He, Payal Bajaj, Nishith Khandwala, Imon Banerjee, Daniel Rubin, and Christopher Ré. “Inferring generative model structure with static analysis”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 240–250.

[254] Paroma Varma and Christopher Ré. “Snuba: automating weak supervision to label training data”. In: *Proceedings of the VLDB Endowment* 12.3 (2018), pp. 223–236.

[255] Paroma Varma, Frederic Sala, Ann He, Alexander Ratner, and Christopher Ré. “Learning Dependency Structures for Weak Supervision Models”. In: *International Conference on Machine Learning* (2019).

[256] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems 30.* 2017, pp. 5998–6008.

[257] S. V. N. Vishwanathan, Karsten M. Borgwardt, and Nicol N. Schraudolph. “Fast Computation of Graph Kernels”. In: *Advances in Neural Information Processing Systems.* 2006.

[258] Kiri Wagstaff and Claire Cardie. “Clustering with instance-level constraints”. In: *AAAI Conference on Artificial Intelligence* 1097 (2000), pp. 577–584.

[259] Kiri Wagstaff, Claire Cardie, Seth Rogers, Stefan Schrödl, et al. “Constrained k-means clustering with background knowledge”. In: *International Conference on Machine Learning.* 2001, pp. 577–584.

[260] Kiri Lou Wagstaff. “Intelligent Clustering with Instance-level Constraints”. PhD thesis. Ithaca, NY, USA: Cornell University, 2002.

[261] A Wallis and P McCoubrie. “The radiology report—are we getting the message across?” In: *Clinical radiology* 66.11 (2011), pp. 1015–1022.

[262] Fei Wang, Jimeng Sun, and Shahram Ebadollahi. “Integrating distance metrics learned from multiple experts and its application in patient similarity assessment”. In: *Proceedings of the 2011 SIAM International Conference on Data Mining.* SIAM. 2011, pp. 59–70.

[263] Hai Wang and Hoifung Poon. “Deep Probabilistic Logic: A Unifying Framework for Indirect Supervision”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing.* 2018, pp. 1891–1902.

[264] Qinxin Wang, Hao Tan, Sheng Shen, Michael Mahoney, and Zhewei Yao. “MAF: Multimodal Alignment Framework for Weakly-Supervised Phrase Grounding”. In: *Conference on Empirical Methods in Natural Language Processing (EMNLP).* 2020, pp. 2030–2038.

[265] Shusen Wang, Alex Gittens, and Michael W. Mahoney. “Scalable Kernel K-means Clustering with Nyström Approximation: Relative-error Bounds”. In: *Journal of Machine Learning Research* 20.1 (Jan. 2019), pp. 431–479.

[266] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. “ChestX-Ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017, pp. 2097–2106. DOI: 10.1109/CVPR.2017.369.

[267] Peter Welinder, Steve Branson, Pietro Perona, and Serge Belongie. “The Multidimensional Wisdom of Crowds”. In: *Advances in Neural Information Processing Systems*. Vol. 23. 2010, pp. 2424–2432.

[268] John R Wilcox. “The written radiology report”. In: *Applied Radiology* 35.7 (2006), p. 33.

[269] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. “Transformers: State-of-the-art natural language processing”. In: *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*. 2020, pp. 38–45.

[270] Baoyuan Wu, Yifan Zhang, Bao-Gang Hu, and Qiang Ji. “Constrained clustering and its application to face clustering in videos”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2013, pp. 3507–3514.

[271] Joy T Wu, Nkechinyere Nneka Agu, Ismini Lourentzou, Arjun Sharma, Joseph Alexander Paguio, Jasper Seth Yao, Edward Christopher Dee, William G Mitchell, Satyananda Kashyap, Andrea Giovannini, et al. “Chest ImaGenome Dataset for Clinical Reasoning”. In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021.

[272] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. “Google’s neural machine translation system: Bridging the gap between human and machine translation”. In: *arXiv preprint arXiv:1609.08144* (2016). URL: https://arxiv.org/abs/1609.08144.

[273] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. Aug. 28, 2017. arXiv: cs.LG/1708.07747 [cs.LG].

[274] Eric P Xing, Michael I Jordan, Stuart J Russell, and Andrew Y Ng. “Distance metric learning with application to clustering with side-information”. In: *Advances in Neural Information Processing Systems*. 2003, pp. 521–528.

[275] Jia Xu, Alexander G Schwing, and Raquel Urtasun. “Learning to segment under various forms of weak supervision”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3781–3790.

[276] Yichong Xu, Hariank Muthakana, Sivaraman Balakrishnan, Aarti Singh, and Artur Dubrawski. "Nonparametric Regression with Comparisons: Escaping the Curse of Dimensionality with Ordinal Information". In: *International Conference on Machine Learning.* July 2018.

[277] Yichong Xu, Hongyang Zhang, Kyle Miller, Aarti Singh, and Artur Dubrawski. "Noise-Tolerant Interactive Learning Using Pairwise Comparisons". In: *Advances in Neural Information Processing Systems.* 2017, pp. 2431–2440.

[278] Bojun Yan and Carlotta Domeniconi. "An adaptive kernel method for semi-supervised clustering". In: *European Conference on Machine Learning.* 2006, pp. 521–532.

[279] Xin Yi, Ekta Walia, and Paul Babyn. "Generative adversarial network in medical imaging: A review". In: *Medical image analysis* 58 (2019), p. 101552.

[280] Xuesong Yin, Songcan Chen, Enliang Hu, and Daoqiang Zhang. "Semi-supervised clustering with metric learning: An adaptive kernel method". In: *Pattern Recognition* 43.4 (2010), pp. 1320–1333.

[281] Yang You, Igor Gitman, and Boris Ginsburg. "Large batch training of convolutional networks". In: *arXiv preprint arXiv:1708.03888* (2017). URL: `https://arxiv.org/pdf/1708.03888v3.pdf`.

[282] Fisher Yu and Vladlen Koltun. "Multi-Scale Context Aggregation by Dilated Convolutions". In: *4th International Conference on Learning Representations (ICLR).* San Juan, Puerto Rico, May 2016.

[283] 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". In: *arXiv preprint arXiv:1506.03365* (2015).

[284] Tianyu Yu, Tianrui Hui, Zhihao Yu, Yue Liao, Sansi Yu, Faxi Zhang, and Si Liu. "Cross-modal omni interaction modeling for phrase grounding". In: *The 28th ACM International Conference on Multimedia.* 2020, pp. 1725–1734. DOI: `10.1145/3394171.3413846`.

[285] Hamed Zamani and W. Bruce Croft. "On the Theory of Weak Supervision for Information Retrieval". In: *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval.* 2018, pp. 147–154.

[286] Hamed Zamani, W. Bruce Croft, and J. Shane Culpepper. "Neural Query Performance Prediction Using Weak Supervision from Multiple Signals". In: *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval.* 2018, pp. 105–114.

[287] Andrea Zanette, Junzi Zhang, and Mykel J Kochenderfer. "Robust super-level set estimation using gaussian processes". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* Springer. 2018, pp. 276–291.

[288] Jieyu Zhang, Cheng-Yu Hsieh, Yue Yu, Chao Zhang, and Alexander Ratner. *A Survey on Programmatic Weak Supervision.* 2022. arXiv: 2202.05433 [cs.LG].

[289] Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. "WRENCH: A Comprehensive Benchmark for Weak Supervision". In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2).* 2021.

[290] Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I Jordan. "Spectral methods meet EM: A provably optimal algorithm for crowdsourcing". In: *Advances in Neural Information Processing Systems.* 2014, pp. 1260–1268.

[291] Yuhao Zhang, Daisy Yi Ding, Tianpei Qian, Christopher D Manning, and Curtis P Langlotz. "Learning to Summarize Radiology Findings". In: *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis.* 2018, pp. 204–213. DOI: 10.18653/v1/W18-5623.

[292] Yuhao Zhang, Hang Jiang, Yasuhide Miura, Christopher D Manning, and Curtis P Langlotz. "Contrastive learning of medical visual representations from paired images and text". In: *Machine Learning for Healthcare Conference.* PMLR. 2022, pp. 2–25.

[293] Zhilu Zhang and Mert R Sabuncu. "Generalized cross entropy loss for training deep neural networks with noisy labels". In: *32nd Conference on Neural Information Processing Systems (NeurIPS).* 2018.

[294] Zhu Zhang, Zhou Zhao, Zhijie Lin, Xiuqiang He, et al. "Counterfactual Contrastive Learning for Weakly-Supervised Vision-Language Grounding". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 18123–18134.

[295] Zhi-Hua Zhou. "A brief introduction to weakly supervised learning". In: *National science review* 5.1 (2018), pp. 44–53.

Acronyms

ARI Adjusted Rand Index

AUC Area Under the Curve

CMU Carnegie Mellon University

CT Computed Tomography

CXR Chest X-ray

DP Data Programming

ECG Electrocardiogram

FID Fréchet Inception Distance

GAN Generative Adversarial Network

GANs Generative Adversarial Networks

LF Labeling Function

LFs Labeling Functions

LSE Level Set Estimation

LSTM Long Short-term Memory Networks

MKL Multiple Kernel Learning

MKNN Mutual k-Nearest Neighbors

ML Machine Learning

MLM Masked Language Modeling

MLP Multilayer Perceptron

MRF Markov Random Field

NE Neighborhood Evidence

NMI Normalized Mutual Information

PGM Probabilistic Graphical Model

ReLU Rectified Linear Unit

SMBO Sequential Model Based Optimization

VLP Vision Language Processing

Appendices

Appendix A

Constrained Clustering and Multiple Kernel Learning without Pairwise Constraint Relaxation

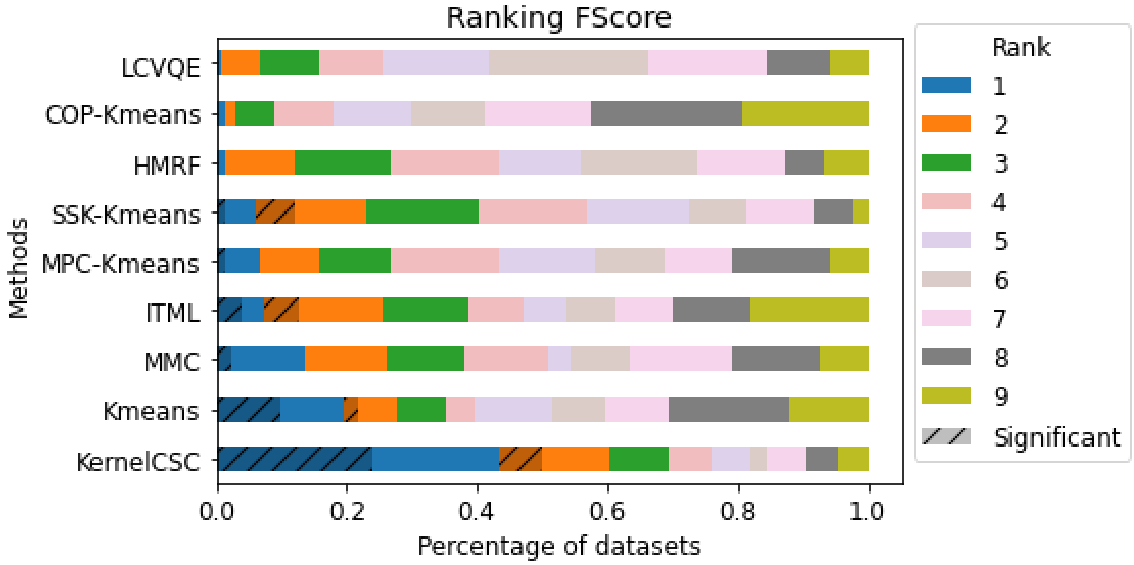


Figure A.1: Percentage of times over all datasets each algorithm is ranked first on the test set (y-axis), vs. the number of pairwise training constraints (x-axis) used in training. The ranks were established on test-sets using mean F-score over 10 random trials.

A.1 Varying Evaluation Metrics

To evaluate and compare the performance of the proposed approach, numerous evaluation metrics were computed such as the ARI, F-score and NMI. The superior performance of the proposed approach remains consistent across these different evaluation metrics. In addition to the ARI results shown in the main document, Figure A.1 and Figure A.2 demonstrate that the proposed approach also outperforms related approaches under F-score and NMI. Under NMI, the percentage of datasets where the proposed approach ranks first is slightly lower compared to evaluations done with F-score and ARI.

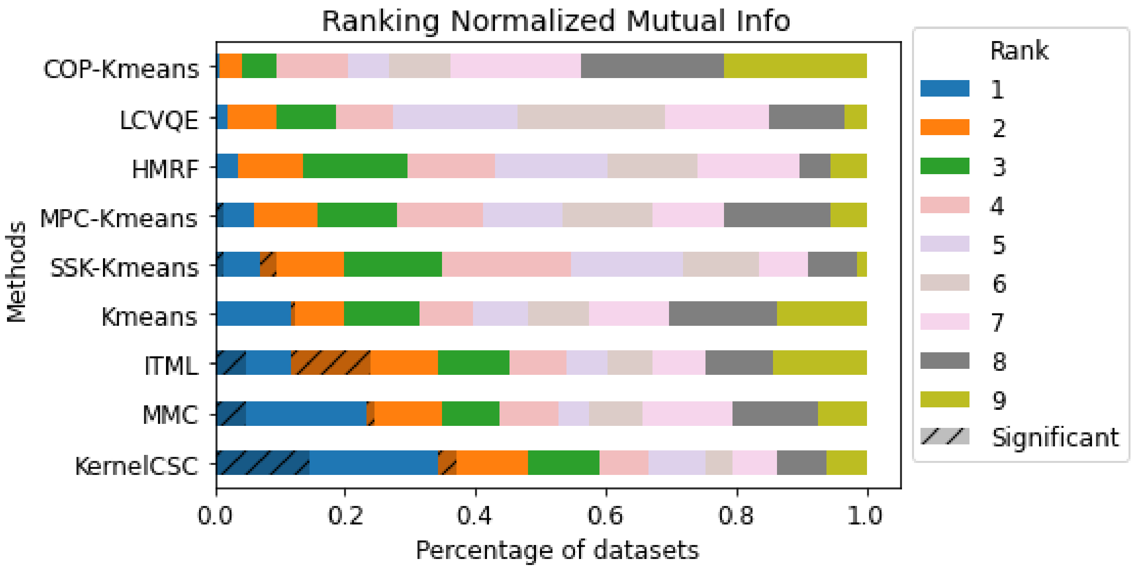


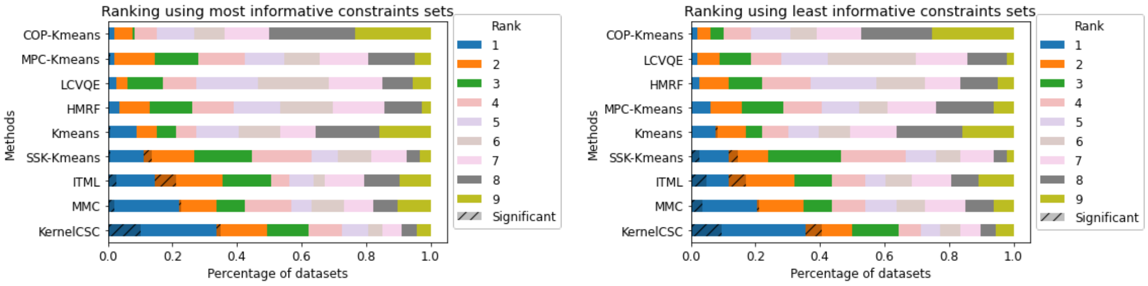
Figure A.2: Percentage of times over all datasets each algorithm is ranked first on the test set (y-axis), vs. the number of pairwise training constraints (x-axis) used in training. The ranks were established on test-sets using mean NMI over 10 random trials.

A.2 Informativeness and Coherence Measures

[70] suggested that averaging over different randomly chosen constraint sets may mask interesting properties of the individual constraint sets. The authors introduce two quantitative measures, *informativeness* and *coherence*, and among other things, use these measures to inspect disparities in performance of different clustering algorithms. Since the evaluation of the propose approach of this thesis averages algorithm performance over randomly chosen constraint sets, this section discusses an additional analysis that was performed in which the relative performance is inspected when more

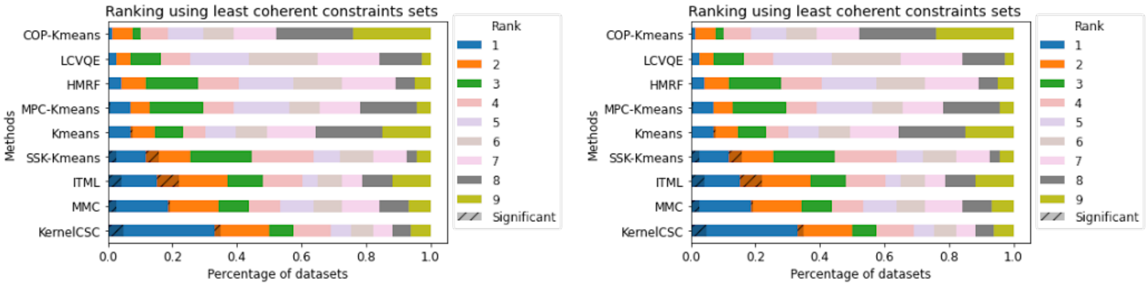
and less informative or coherent constraint sets are used, to see if differences in the relative ranking of algorithms emerge when these metrics vary.

No systematic differences in the relative performance of the top performing algorithms is observed when relative algorithm performance is compared across datasets using different levels of *informative* and *coherent* constraint sets. A small drop in performance occurs across all algorithms when less coherent constraints are used. In the relative comparison between algorithms, this does lead to some ranking differences being less significant. To illustrate this, in Figure A.3 recreates the results of Figure 2.1 but uses the 5 least informative constraint sets compared to the 5 most informative constraint sets. Similarly, Figure A.4 shows the 5 least coherent constraint sets compared to the 5 most coherent constraint sets. All rankings in these figures rely on ARI scores.



(a) Using most informative constraint sets. (b) Using least informative constraint sets.

Figure A.3: Percentage of times over all datasets each algorithm is ranked first on the test set (y-axis), vs. the number of pairwise training constraints (x-axis) used in training. The ranks were established on test-sets using mean ARI over the 5 out of 10 most and least informative random trials.



(a) Most coherent constraint sets. (b) Least coherent constraint sets.

Figure A.4: Percentage of times over all datasets each algorithm is ranked first on the test set (y-axis), vs. the number of pairwise training constraints (x-axis) used in training. The ranks were established on test-sets using mean Adjusted Rand Index over the 5 out of 10 most and least coherent constraint sets.

Appendix B

End-to-End Weak Supervision

B.1 Posterior Reparameterization

This section motivates the design choices and inductive biases of the proposed approach which were encoded into the neural encoder network e, i.e. the network that is used to model the relative accuracies of the weak supervision sources $\boldsymbol{\lambda}$. Recall that we model the probability of a particular sample $\mathbf{x} \in \mathcal{X}$ having the class label $y \in \mathcal{Y} = \{1, \dots, C\}$ as

$$P_\theta(y|\,\boldsymbol{\lambda}) = \operatorname{softmax}\left(\mathbf{s}\right)_y P(y), \tag{B.1}$$

$$\mathbf{s} = \theta(\boldsymbol{\lambda}, \mathbf{x})^T \bar{\boldsymbol{\lambda}} \in \mathbb{R}^C\,. \tag{B.2}$$

where $\theta(\boldsymbol{\lambda}, \mathbf{x}) \in \mathbb{R}^m$ weighs the LF votes on a sample-by-sample basis and the softmax for class y on s is defined as

$$\operatorname{softmax}\left(\mathbf{s}\right)_y = \frac{\exp\left(\theta(\boldsymbol{\lambda}, \mathbf{x})^T \mathbb{1}\{\boldsymbol{\lambda} = y\}\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\theta(\boldsymbol{\lambda}, \mathbf{x})^T \mathbb{1}\{\boldsymbol{\lambda} = y'\}\right)}.$$

Connection to prior PGM models This choice will be motivated by deriving a less expressive variant of it from the standard MRF label model approach. If we view the attention scores $\theta(\boldsymbol{\lambda}, \mathbf{x}) \in \mathbb{R}^m$, that assign sample-dependent accuracies to each labeling function, as sample-independent parameters θ_1 and, by that, eliminate the features from the equation, we can rewrite Eq. B.1 as

$$\frac{\exp\left(\theta_1^T \mathbb{1}\{\boldsymbol{\lambda} = y\}\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\theta_1^T \mathbb{1}\{\boldsymbol{\lambda} = y'\}\right)} P(y)$$

Let $\phi_1(\boldsymbol{\lambda}, y) = \mathbb{1}\{\boldsymbol{\lambda} = y\}$. For clarity, we will drop the class balance, so that the expression becomes

$$
\begin{aligned}
&= \frac{\exp\left(\theta_1^T \phi_1(\boldsymbol{\lambda}, y)\right)}{\sum_{y' \in \mathcal{Y}} \exp\left(\theta_1^T \phi_1(\boldsymbol{\lambda}, y')\right)} \\
&= \frac{Z_\theta^{-1} \exp\left(\theta_1^T \phi_1(\boldsymbol{\lambda}, y) + \theta_2^T \phi_2(\boldsymbol{\lambda})\right)}{\sum_{y' \in \mathcal{Y}} Z_\theta^{-1} \exp\left(\theta_1^T \phi_1(\boldsymbol{\lambda}, y') + \theta_2^T \phi_2(\boldsymbol{\lambda})\right)} \\
&= \frac{P_\theta\left(\boldsymbol{\lambda}, y\right)}{\sum_{y' \in \mathcal{Y}} P_\theta\left(\boldsymbol{\lambda}, y'\right)} \\
&= \frac{P_\theta\left(\boldsymbol{\lambda}, y\right)}{P_\theta\left(\boldsymbol{\lambda}\right)} \\
&= P_\theta\left(y \middle| \boldsymbol{\lambda}\right),
\end{aligned}
$$

where in the second step the denominator and numerator are multiplied with the same quantity $\frac{1}{Z_\theta} \exp\left(\theta_2^T \phi_2(\boldsymbol{\lambda})\right)$, and θ now parameterizes the joint distribution of the latent label and weak sources as

$$
P_\theta(\boldsymbol{\lambda}, y) = \frac{1}{Z_\theta} \exp\left(\theta_1^T \phi_1(\boldsymbol{\lambda}, y) + \theta_2^T \phi_2(\boldsymbol{\lambda})\right) = \frac{1}{Z_\theta} \exp\left(\theta^T \phi(\boldsymbol{\lambda}, y)\right).
$$

We can recognize P_θ as a distribution from the exponential family, and more specifically as a pairwise MRF, or factor graph, with canonical parameters $\theta = (\theta_1, \theta_2)$ and corresponding sufficient statistics, or factors, $\phi(\boldsymbol{\lambda}, y) = (\phi_1(\boldsymbol{\lambda}, y), \phi_2(\boldsymbol{\lambda}))$, as well as the log partition function Z_θ. The accuracy factors and parameters ϕ_1, θ_1 are the core components of this model and sometimes take the form $\phi_1(\boldsymbol{\lambda} y) = \boldsymbol{\lambda} y$ in binary models as in [222, 93, 42]. The label-independent factors $\phi_2(\boldsymbol{\lambda})$ have, as can be seen from the derivation above, no direct influence on the latent label posterior, but are often used to model labeling propensities $\mathbb{1}\{\boldsymbol{\lambda} \neq 0\}$ and correlation dependencies $\mathbb{1}\{\lambda_i = \lambda_j\}$, which can be important for PGM parameter learning, but are susceptible to misspecification [255, 42, 35]. *The parameterization of the proposed approach is therefore a more expressive variant of the posterior of the latent-variable PGM models, where LF accuracies can now be assigned on a sample-by-sample basis. Furthermore, the neural encoder network outputs them as a function of the LF outputs* ***and*** *features, and may learn the dependencies and label-independent statistics. Indeed, the empirical findings of this work support this.*

B.2 Extended Results

We provide more detailed results in Table B.1. Here, we include `WeaSEL`-random, which corresponds to `WeaSEL` with a randomly initialized encoder network that is not trained/updated. As expected, this setting produces performance often similar compared to training an end model on the hard majority vote labels. This is due to

Table B.1: The final test F1 performance of various multi-source weak supervision methods over seven runs, using different random seeds, are averaged out ± standard deviation. The top 2 performance scores are highlighted as **First**, **Second**. Triplet-median [42] is not listed as it only converged for IMDB with 12 LFs (F1 = 73.0±0.22), and Spouse (F1 = 48.7 ± 1.0). Sup. (Val. set) is the performance of the downstream model trained in a supervised manner on the labeled validation set. The rest are state-of-the-art latent label models. For reference, the table also shows the *Ground truth* performance of a fully supervised model trained on true training labels (which are unused by all other models, and not available for Spouse). The performance of `WeaSEL`-random is also shown, where only the downstream model of `WeaSEL` is trained (and the encoder network is left at its randomly initialized state). All models are run twice, where only the learning rate differs (either 10^{-4} or $4 \cdot 10^{-5}$), and the model with best ROC-AUC on the validation set is reported. The probabilistic labels from Snorkel used for downstream model training are chosen over six different configurations of the learning rate and number of epochs (again with respect to validation set ROC-AUC).

Model	**Spouse** (9 LFs)	**ProfTeacher** (99 LFs)	**IMDB** (136 LFs)	**IMDB** (12 LFs)	**Amazon** (175 LFs)
Ground truth	–	90.65 ± 0.29	86.72 ± 0.40	86.72 ± 0.40	92.93 ± 0.68
Sup. (Val. set)	20.4 ± 0.2	73.34 ± 0.00	68.76 ± 0.00	68.76 ± 0.00	84.18 ± 0.00
Snorkel	48.79 ± 2.69	85.12 ± 0.54	**82.22 ± 0.18**	**74.45 ± 0.58**	80.54 ± 0.41
Triplet	45.88 ± 3.64	74.43 ± 10.59	75.36 ± 1.92	73.15 ± 0.95	75.44 ± 3.21
Triplet-Mean	**49.94 ± 1.47**	82.58 ± 0.32	79.03 ± 0.26	73.18 ± 0.23	79.44 ± 0.68
`WeaSEL`-random	46.43 ± 3.29	83.47 ± 0.64	79.80 ± 0.48	74.22 ± 0.45	82.22 ± 0.57
Majority vote	40.67 ± 2.01	**85.44 ± 0.37**	80.86 ± 0.28	74.13 ± 0.31	**84.20 ± 0.52**
`WeaSEL`	**51.98 ± 1.60**	**86.98 ± 0.45**	**82.10 ± 0.45**	**77.22 ± 1.02**	**86.60 ± 0.71**

the strong inductive bias in our encoder model that constrains the encoder labels to be a normalized linear combination of the LF votes, weighted by positive accuracy scores. In fact, `WeaSEL`-random itself is often able to outperform the PGM-based baselines, in particular the triplet methods. Our results show that `WeaSEL` consistently improves significantly upon these baselines via training the encoder network to maximize its agreement with the downstream model.

B.3 Extended Implementation Details

Weak supervision sources For the Spouses dataset, and the IMDB variant with 12 LFs, the same LFs are used as in [93] and [42], respectively[1]. The set of 12 IMDB LFs was specifically chosen to have a large coverage, see Table 3.3. These LFs and the larger set of LFs that this work introduces for the second IMDB experiment are all pattern- and regex-based heuristics, i.e. LFs that label whenever a certain word or bi-gram appears in a text document. For instance, 'excellent' would label a sample as a positive movie review (and would do so with 80% accuracy on the samples where it does not abstain). This holds for the other text datasets as well, while the Spouse experiments also contain LFs that are distant supervision sources based on DBPedia. For the remaining datasets (IMDB with 136 LFs, Bias Bios, and Amazon), the respective LF sets were created by the authors of this work, and remained fixed throughout the experiments.

Encoder network architectures In all experiments, we use a simple MLP as the encoder e, with two hidden layers, batch normalization, and Rectified Linear Unit (ReLU) activation functions. For the Spouse dataset, the hidden layers of the network are a bottleneck of 50, 5. This is motivated by the small size of the set of samples labeled by at least one LF. For all other datasets, hidden dimensions of 70, 70 are set. Ablations in Table B.2 show that the proposed end-to-end model also succeeds for different encoder architecture choices.

Downstream models For all datasets besides Spouse, a three-layer MLP with hidden dimensions of 50, 50, 25 is used. For Spouse, a single-layer bidirectional LSTM is used, with a hidden dimension of 150, followed by two fully-connected readout layers with dimensions 64, 32. All fully-connected layers use ReLU activation functions. Simple downstream architectures were chosen as the relative improvements over other label models are the core interest, rather than the best possible downstream performance due to the downstream architecture. Naturally, sophisticated architectures are expected to further improve the performance further for the larger datasets.

[1] All necessary label matrices are available in the code release. At the time of writing, the Spouse LFs and data are also available at the following URL: `https://github.com/snorkel-team/snorkel-tutorials/blob/master/spouse/spouse_demo.ipynb`

Hyperparameters Unless explicitly mentioned, all reported experiments are averaged out over seven random seeds. An L2 weight decay of 7e-7 and a dropout of 0.3 are used for both encoder and downstream models for all datasets but Spouse (where the LSTM does not use dropout). All models are optimized with Adam, with early-stopping based on AUC performance on the small validation set, and a maximum number of 150 epochs (75 for Spouse). The batch size is set to 64. The loss function is set to the (binary) cross-entropy. For each dataset and each model/baseline, the same experiment is run for learning rates of 1e-4 and 3e-5. The model performance that is reported is chosen according to the best ROC-AUC performance on the small validation set. For the Spouse dataset, additional experiments are run with an L2 weight decay of 1e-4. For the proposed `WeaSEL` model, additional experiments are conducted for the Spouses dataset with different configurations of the temperature hyperparameter, $\tau_1 \in \{1, 1/3\}$. Again, the test performance as measured by the best validation ROC-AUC is reported.

The probabilistic labels from Snorkel used for downstream model training are chosen over six different configurations of the learning rate and number of epochs for Snorkel's label model (again with respect to validation set ROC-AUC). For all binary classification datasets (i.e. all except for LabelMe), the downstream model's decision threshold is tuned based on the resulting F1 validation score for all models. All label model baselines are provided with the class balance, which `WeaSEL` does not use (but which is expected to be helpful for unbalanced classes, where no validation set is available).

B.4 Extended Ablations

The full ablations are reported in Table B.2, where exactly one component of the proposed `WeaSEL` model is changed or removed. Most changes consistently underperform the base `WeaSEL` design shown in the main document, and the occasional positive changes – 1e-4 weight decay, and the Squared Hellinger loss instead of the symmetric cross-entropy – only beat the base `WeaSEL` performance in at most two datasets, and never significantly. In practice, it is of course advised to explore such configurations if a validation set is available.

Letting the accuracy scores depend on the input features (first row), usually boosts performance, but not by much (1.2 F1 points at most). On the other hand, it proves very important to allow the accuracy scores to depend non-linearly on the LF votes and the features: A linear encoder network, as in [37], significantly underperforms `WeaSEL` with at least one hidden layer by up to 4.9 F1 score points. Conversely, a deeper encoder network (of hidden dimensionalities $75, 50, 25, 50, 75$, see fourth row) does not improve results.

While the effect of the inverse temperature parameter τ_1–which controls the softness of the encoder-predicted accuracy scores–on downstream performance is not large, it can have significant effects on the learning dynamics and robustness, see Fig B.1 for

Table B.2: Ablative study on the sub-components of our algorithm as in Algorithm 2 (over 5 random seeds). In each row below, exactly one component of WeaSEL is changed, and the resulting F1 score is reported. Note that the scores for WeaSEL are slightly different to the ones in the main results table, since these ablations here were run separately, with fewer seeds, and for only one learning rate (1e-4). Configurations that **outperform base WeaSEL are highlighted in bold font**, while the **four worst performing configurations** are highlighted in red for each dataset. Note that bold font does not indicate significant differences.

Change	ProfTeacher	IMDB-136 LFs	IMDB-12 LFs	Amazon
WeaSEL	86.8 ± 0.4	82.1 ± 0.7	77.3 ± 0.5	86.6 ± 0.5
$\theta(\boldsymbol{\lambda}, \mathbf{x}) = \theta(\boldsymbol{\lambda})$	85.6 ± 1.6	82.1 ± 0.5	75.9 ± 0.8	86.6 ± 0.4
Linear e	81.9 ± 0.7	80.0 ± 0.6	73.2 ± 0.6	82.6 ± 0.5
1 hidden layer e	**87.1 ± 0.7**	81.8 ± 0.6	76.8 ± 0.9	85.3 ± 0.8
75x50x25x50x75 e	84.3 ± 2.1	81.9 ± 0.6	75.8 ± 1.1	86.1 ± 0.6
$\tau_1 = 2$	86.7 ± 1.0	81.9 ± 0.3	77.3 ± 0.5	85.5 ± 1.0
$\tau_1 = 1/2$	86.5 ± 0.8	81.8 ± 0.5	76.0 ± 1.4	86.4 ± 0.3
$\tau_1 = 1/4$	84.5 ± 1.2	81.8 ± 0.2	73.9 ± 0.9	85.6 ± 1.0
$\tau_2 = 1$	85.2 ± 1.6	**82.2 ± 0.4**	76.6 ± 1.0	84.3 ± 1.2
$\tau_2 = m$	86.1 ± 0.7	81.2 ± 0.6	76.4 ± 0.4	85.7 ± 0.2
No BatchNorm	82.6 ± 1.4	81.9 ± 0.5	74.7 ± 0.7	85.3 ± 0.8
1e-4 weight decay	**87.4 ± 0.4**	80.9 ± 1.3	**77.9 ± 0.6**	85.2 ± 0.5
MIG loss	86.7 ± 0.4	78.7 ± 0.4	74.1 ± 0.4	84.7 ± 1.8
L1 loss	86.2 ± 0.6	81.1 ± 0.5	75.6 ± 0.9	84.1 ± 0.9
Squared Hellinger loss	**87.4 ± 0.3**	**82.2 ± 0.6**	75.7 ± 1.1	86.3 ± 0.4
CE(P_f, P_e) asymm. loss	**77.3 ± 3.7**	**77.7 ± 1.1**	71.7 ± 0.3	**78.7 ± 1.2**
CE(P_e, P_f) asymm. loss	**73.1 ± 6.8**	**71.9 ± 1.9**	**69.7 ± 0.7**	**70.1 ± 1.1**
No stop-grad	**80.4 ± 2.1**	**76.2 ± 0.5**	**71.0 ± 0.6**	79.3 ± 0.6
$\theta(\boldsymbol{\lambda}, \mathbf{x}) = \sqrt{m} \cdot \text{sigmoid}(e(\boldsymbol{\lambda}, \mathbf{x}))$	85.5 ± 0.6	81.8 ± 0.5	**78.0 ± 0.7**	**86.9 ± 0.3**
$\theta(\boldsymbol{\lambda}, \mathbf{x}) = \text{ReLU}(e(\boldsymbol{\lambda}, \mathbf{x}))+$ 1e-5	83.0 ± 2.3	78.3 ± 1.1	**69.1 ± 2.1**	**74.2 ± 2.7**
$\theta(\boldsymbol{\lambda}, \mathbf{x}) = \text{Tanh}(e(\boldsymbol{\lambda}, \mathbf{x}))$	**71.9 ± 4.0**	**67.0 ± 0.8**	**67.0 ± 1.1**	**67.3 ± 1.1**

such learning curves as a function of epoch number. In particular, a lower τ_1 helps to stabilize the training dynamics, since the accuracy score weights are more evenly distributed across LFs, which appears to help avoid overfitting. When overfitting is not easily detectable due to a lack of a validation set, it is therefore advisable to use a lower τ_1. It also proves helpful to scale the softmax in Eq. 3.3 by $\sqrt{m}$, rather than not scaling it ($\tau_2 = 1$ row) or scaling by m.

Changing the loss function from the symmetric cross-entropy to the MIG function [37] or the L1 loss consistently leads to worse performance. The former is interesting, since using the MIG loss for the crowdsourcing dataset LabelMe, see subsection 3.1.2, was important in order to achieve state-of-the-art crowdsourcing performance (with a similar lift in performance observable for Snorkel using MIG for downstream model training). The result provides some evidence that the MIG loss may be inappropriate for weak supervision settings other than crowdsourcing.

The ablations show that it is important to constrain the accuracy score space to a positive interval, either by viewing them as an aggregation of the LFs via the scaled softmax in Eq. 3.3, or by replacing the softmax with a sigmoid function. Indeed, using a less constrained activation function for the estimated accuracies (last two rows, where the 1e-5 in the ReLU row avoids accuracy scores equal to zero) significantly under-performs: Allowing the accuracies to be negative (last row) leads to collapse and bad downstream performance. This is likely due to the removal of the inductive bias that LFs are designed by users to be better-than-random, which makes the joint optimization more likely to find trivial solutions. Additionally, the choice of using the symmetric cross-entropy loss with `stop-grad` applied to the targets is crucial for the performance of `WeaSEL`. Removing the `stop-grad` operation, or using the standard cross-entropy (without `stop-grad` on the target) leads to significantly worse scores and a very brittle model. This is expected, since conceptually our goal is to have an objective that maximizes the agreement between a pair of models that predict based on two different views of the latent label, the features and the LF votes. The cross-entropy with `stop-grad` on the target[2] naturally encodes this understanding, since each model uses the other model's predictions as a reference distribution. Losses that already are symmetric (e.g. L1 or Squared Hellinger loss) neither need to be symmetrized nor use `stop-grad`. While the L1 loss consistently underperforms, we find that the Squared Hellinger loss can lead to better performance on two out of four datasets.

However, only the symmetric cross-entropy loss with `stop-grad` on the targets is shown to be robust and able to recover the true labels in the synthetic experiments in Section B.6, see Fig. B.3 in particular. The synthetic ablation in Section B.6 gives interesting insights, and strongly supports the proposed design of `WeaSEL`. Indeed, many choices for `WeaSEL` that perform well enough on the real datasets, such as no features for the encoder, $\tau_2 = 1$, sigmoid parameterized accuracies, and all other objectives that were evaluated, lead to significantly worse performance and less robust

[2] or, due to the `stop-grad` operation, equivalently the KL divergence

learning on the synthetic adversarial setups.

B.5 A Crowdsourcing Dataset

The multi-class LabelMe image classification dataset that was previously used in the most related crowdsourcing literature [228, 37] was chosen for an additional evaluation of the proposed `WeaSEL` approach. Note that this dataset consists of $10k$ samples, of which only $1k$ are unique, in the sense that the rest are augmented versions of the $1k$. The samples were annotated by 59 crowdworkers, with a mean overlap of 2.55 annotations per image. The downstream model is identical to the previously reported one in [228, 37]. That is, a VGG-16 neural network is used as feature extractor, and a single fully-connected layer (with 128 units and ReLU activation) and one output layer is put on top, using 50 % dropout.

Experiments were conducted over seven random seeds with a learning rate of 1e-4 and 50 epochs. The reported scores are the ones with best validation set accuracy for a L2 weight decay $\in$ { 7e-7, 1e-4 }. The validation set is of size 200, and was split at random from the training set prior to running the experiments.
As is usual in the related work for multi-class settings [220], class-conditional accuracies $\theta(\boldsymbol{\lambda}, \mathbf{x}) \in \mathbb{R}^{m \times C}$ are used instead of only m class-independent accuracies. Recall the LF outputs indicator matrix, $\bar{\boldsymbol{\lambda}} \in \mathbb{R}^{m \times C}$. To compute the resulting output softmax logits $\mathbf{s} \in \mathbb{R}^C$, we set $\mathbf{A} = \theta(\boldsymbol{\lambda}, \mathbf{x}) \odot \bar{\boldsymbol{\lambda}} \in \mathbb{R}^{m \times C}$ and $\mathbf{s}_j = \sum_i \mathbf{A}_{ij} \in \mathbb{R}$, where $\odot$ is the element-wise matrix product and we sum up the resulting matrix $\mathbf{A}$ across the LF votes dimension.
Snorkel+MIG indicates that the downstream model f was trained on the MIG loss with respect to soft labels generated by the first Snorkel step, label modeling. Snorkel+CE refers analogously to the same training setup, but using the cross-entropy (CE) loss. All crowdsourcing baseline models are based on the open-source code from [37].

B.6 Robustness Experiments

This section provides more details on the experiments that validate the robustness of the proposed approach against (strongly) correlated LFs that are not better than a random coin flip. In addition, this section presents one further experiment where the random LFs are independent of each other – a more difficult setup for learning (but which does not violate any assumptions of the PGM-based methods) – and the proposed approach, `WeaSEL`, again is shown to be robust to a large extent.
In contrast to `WeaSEL`, prior PGM-based approaches [220, 93, 42] attain significantly worse performance under these settings, due to assuming a Naive Bayes generative model where the weak label sources are conditionally independent given the latent label.

B.6.1 Adversarial LF duplication

For this experiment, a set of 12 LFs from the IMDB dataset is used, and fake adversarial sources are generated by flipping the abstain votes of the 80%-accurate LF that labels for the positive sentiment on 'excellent' to negative ones.

B.6.2 Recovery of True Labels

This set of synthetic experiments focuses on the Bias in Bios dataset, and uses its features and true labels y^*. We let the initial LF set consist of 1) a 100% accurate LF, that is $\lambda_1 = y^*$, and 2) a LF that votes according to the class balance (i.e. a coin flip with probabilities for tail/head set according to the class balance), i.e. $\lambda_2 \sim P(y)$. In the first experiment, the same random LF λ_2 is then duplicated multiple times into the LF set, see Section B.6.2. In the second experiment, random LFs are incrementally added independently of λ_2 (and independently of any other LF already in the LF set), see Section B.6.2. For both setups, the proposed WeaSEL approach is able to recover the performance of the same downstream model, f, that is directly trained on the true labels, y^* (F1 = 90.65, ROC-AUC = 0.967, see Table B.1). In contrast, the PGM-based baselines quickly collapse.

Random LF Duplication

This experiment is inspired by the theoretical comparison in Appendix E of [37] between the authors' end-to-end system and maximum likelihood estimation (MLE) approaches that assume mutually independent LFs. The authors show that such MLE methods are not robust against the following simple example with correlated LFs. Based on the setup described above in B.6.2, the random LF λ_2 is duplicated multiple times, i.e. $\lambda_3 = \cdots = \lambda_m = \lambda_2$. Experiments for varying numbers of duplicates $\in \{2, 25, 100, 500, 2000\}$ are conducted.

WeaSEL is able to consistently and almost completely *recover the fully supervised performance, even when the number of duplicates is very high* ($m = 2001$). Snorkel and triplets methods, on the other hand, fare far worse (AUC ≈ 0.5) for all numbers of duplicates. This behavior is similar to the one observed in B.6.1 (see Fig. 3.2 for the performance of the baselines and WeaSEL averaged out over the varying number of duplicates, and Fig. B.3a-c for the separate performance of WeaSEL for each number of duplicates).

An additional ablation study is run on this synthetic experiment that shows that the observed robustness does not hold for all configurations of WeaSEL. Fig. B.3 shows the test performance curves over the training epochs for the different number of LF duplications.

The proposed WeaSEL model enjoys a stable and robust test curve (Fig. B.3c) and quickly recovers the fully supervised performance, even with 2000 LF duplicates (convergence becomes slower as the LF set contains more duplicates). Many other config-

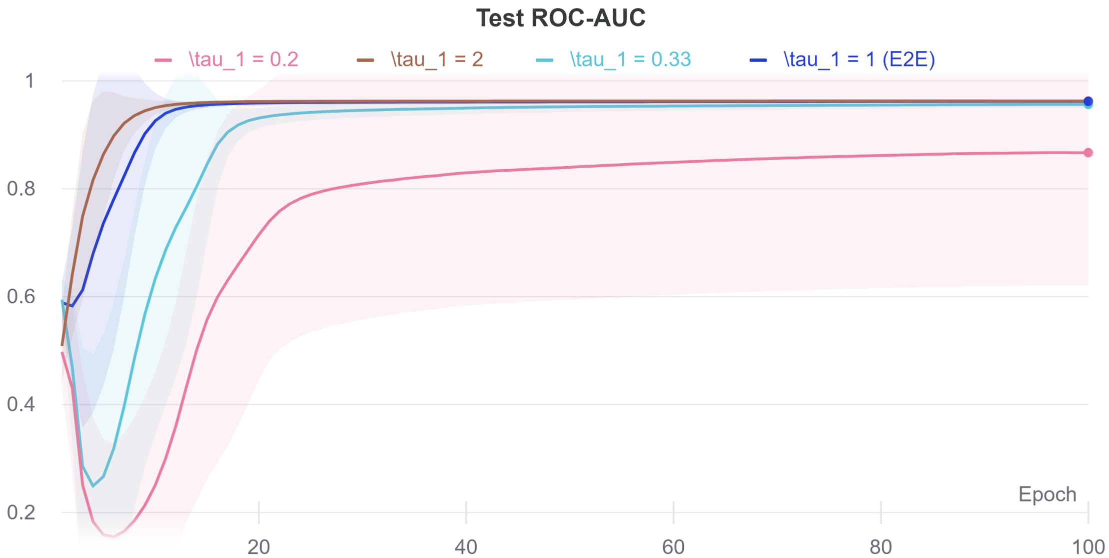


Figure B.1: Test AUC performance at each training epoch for different choices of $\tau_1 \in \{1/5, 1/3, 1, 2\}$ of the synthetic experiment, see Section B.6.2, averaged out over the number of duplicates and five random seeds. A lower τ_1 leads to slower or worse convergence in this specific case. A lower τ_1 corresponds to smoother accuracies, which makes their induced label depend on more LFs. Since in this specific case only one LF is 100% accurate and the rest are not better than a coin flip, the shown behavior is expected.

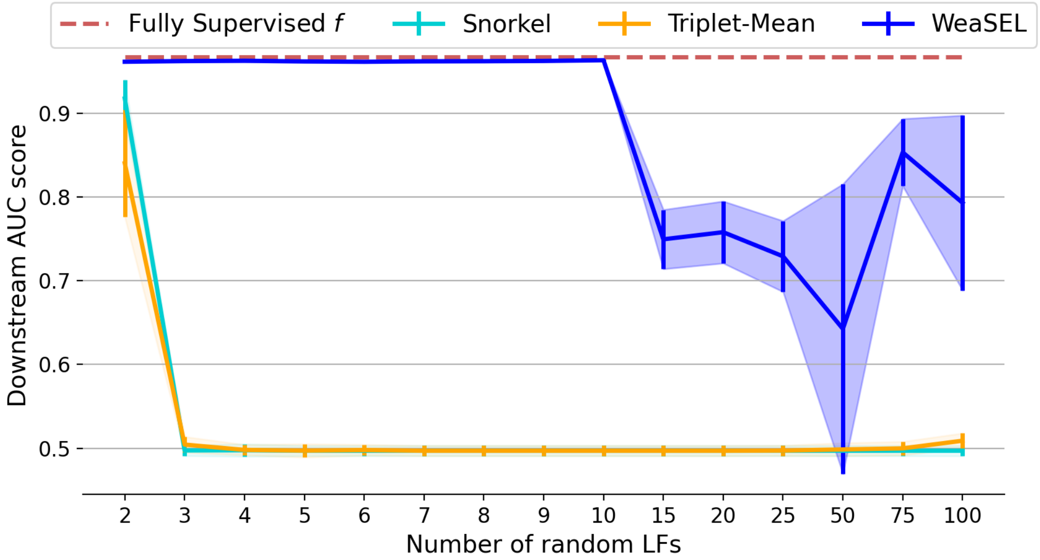


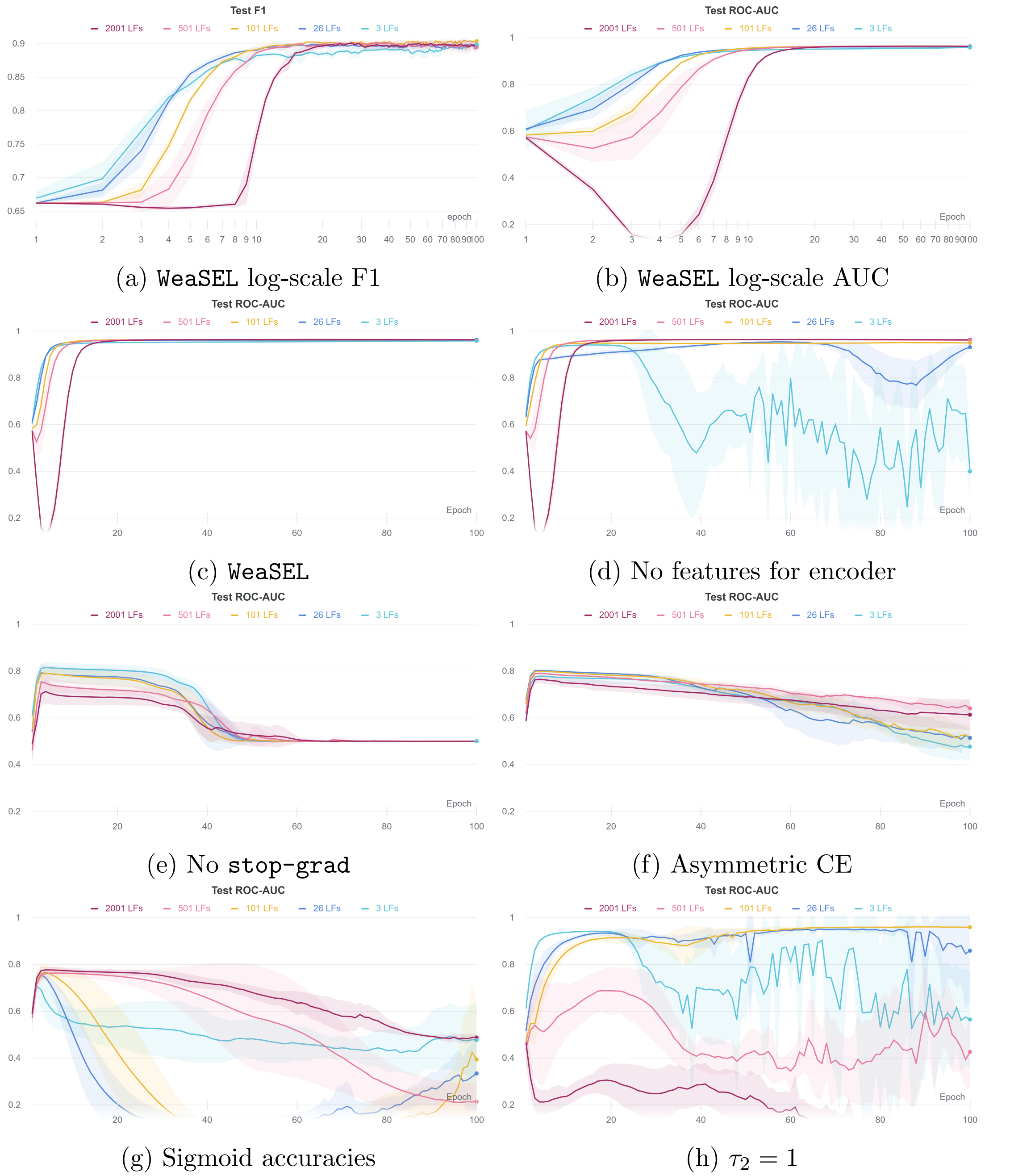
Figure B.2: The experiments start with a 100% accurate LF (i.e. ground truth labels) and incrementally add new, independent LFs that are no better than a random guess. `WeaSEL` recovers the performance of training directly on the ground truth labels (Fully Supervised f), for up to 10 such randomly voting LFs that are independent of each other. The PGM-based prior work rapidly degrades in performance (AUC ≈ 0.5) and is not able to recover any of the 100% accurate signal of the true-labels-LF, as soon as the LF set is corrupted by three or more random LFs. Performances are averaged out over five random seeds, and the standard deviation is shaded. For more details, see Section B.6.2

urations and designs for `WeaSEL` on the other hand lead to worse results. Indeed, for this experiment it is key to use the proposed symmetric cross-entropy with `stop-grad` applied to the targets (see Fig. B.3e, Fig. B.3f), accuracies parameterized by a scaled (Fig. B.3h) softmax (Fig. B.3g), and, to a lesser extent, using the features an input to the encoder (Fig. B.3d).

While the impact of not using `stop-grad`, or using an asymmetric cross-entropy loss is similarly bad in the main ablations on the real datasets, other configurations, and in particular sigmoid-parameterized accuracies (the choice in [140]), an unscaled softmax, and no features for the encoder, often perform well there.

Random Independent LFs

The experiment starts with the same setup as in Section B.6.2, but instead of duplicating the same LF multiple times as in Section B.6.2 a new, independent random LF is drawn at each iteration. That is, the experiment start with $\lambda_1 = y^*, \lambda_2 \sim P(y)$ as the initial LF set, and then new LFs $\lambda_i \sim P(y)$ are added that have no better performance than a coin flip. Notably, since these $\lambda_2, \dots, \lambda_m$ are independent, we are not violating the independence assumptions of PGM-based methods. Nonetheless, the experiment shows that these PGM-based baselines break with only three ($m = 4$) of such random, but independent LFs, while `WeaSEL` is shown to be stable and able to recover the ground truth LF λ_1 for up to 10 random LFs ($m = 11$). For more random LFs, `WeaSEL` starts deteriorating in performance, but is still able to consistently outperform the trivial solution of voting randomly according to the class balance (i.e. based on $\lambda_2, \ldots, \lambda_m$) and the baselines, see Fig. B.2.



(a) `WeaSEL` log-scale F1

(b) `WeaSEL` log-scale AUC

(c) `WeaSEL`

(d) No features for encoder

(e) No `stop-grad`

(f) Asymmetric CE

(g) Sigmoid accuracies

(h) $\tau_2 = 1$

Figure B.3: As a robustness check, this experiment starts with one 100% accurate LF (i.e. ground truth labels), and test performance is plotted at each training epoch for a varying number of duplicates LFs $\in \{2, 25, 100, 500, 2000\}$ of an LF that is no better than a coin flip. Performance is averaged over five random seeds, and the standard deviation is shaded. Details are given in Section B.6.2.

Appendix C

Weakly Supervised GAN (WSGAN)

C.1 Implementation Details and Complexity

(WS)GAN Models The following design choices were used for the experiments conducted with simple DCGAN base networks (as opposed to the settings used in the StyleGAN ablations). *Generator G, Discriminator D, and auxiliary model Q*: Figures C.2 and C.1 show the simple DCGAN [216] based generator and discriminator architectures we use in WSGAN for experiments with 32×32 images. Q and D are neural networks that share all convolutional layers, with a final fully connected layer to output predictions. The dimension of the noise variable z is set to 100, and of b equal to the number of classes. The variable z is sampled from a normal distribution and b from a uniform discrete distribution.
Accuracy Encoder A: For WSGAN-Vector, A is simply a parameter vector of the same length as the number of labeling functions. For WSGAN-Encoder, image features are obtained from the shared convolutional layers of Q and D, which are detached from the computational graph before being passed to an MLP prediction head. For images with 32×32 pixels, the feature vector obtained from the shared convolutional layers is of size $512 * 16$. The MLP head of A is set to have three hidden layers of size $(256, 128, 64)$, with ReLU activations, and an output layer the size of the number of labeling functions followed by a sigmoid function. Significant changes in performance were not observed when the MLP was changed to be shallower or wider. However, for large numbers of LFs, one should consider increasing the width the MLP.
Mappings $F1, F2$: $F1$ and $F2$ are set to each be simple linear models with a softmax at the output. The input and output size of each are set to the number of classes.

(WS)GAN Training The same hyperparameter settings were used for all datasets. All GANs were trained for a maximum of 200 epochs. A batch size of 16 is used, which leads to more stable training dynamics with a DCGAN than larger batch sizes.

Ablation experiments with batch sizes of 8 and 32 did not lead to a significant difference in FID image generation quality or label model accuracy, but in worse training dynamics, i.e. more frequent failures to converge. For WSGAN, four optimizers are used, one for each of the different loss terms: discriminator training, generator training, the Info loss term, and the WSGAN loss term. Adam is used for all optimizers, and the learning rates are set as follows: 4×10^{-4} for D, 1×10^{-4} for G, 1×10^{-4} for the info loss term, and 8×10^{-5} for the WSGAN loss term. The same settings are followed for the training of an InfoGAN training (for the components it shares with WSGAN).

(WS)GAN Training and Failure Cases While WSGAN is still susceptible to the common GAN failure cases of its base networks, such as mode collapse, we empirically find WSGAN training to be more stable than training a GAN that also learns a discrete latent code but uses no weak supervision signals (InfoGAN), despite the high level of noise in our weak supervision sources. InfoGAN failed to converge more frequently.

To help train the DCGAN networks successfully, employing discriminator label flipping (randomly calling a tiny percentage of real samples fake and vice versa) and label smoothing (adding small amounts of noise to the real target of 1.0 and fake target of 0.0) appears to stabilize and improve GAN training. Despite employing such tricks, it was not possible to completely avoid the occasional convergence failure. Fortunately, monitoring the generator and discriminator losses, inspecting the quality of generated images, or tracking image quality metrics such as FID allows one to easily discard failed runs or to pick model checkpoints from earlier iterations before a failure, without requiring labeled data.

StyleWSGAN Model Setup and Training We adapt StyleGAN2-ADA [142] to build a StyleWSGAN Model as well as a StyleInfoGAN. The generator architecture follows the same approach as a class-conditional StyleGAN generator: the sampled code is embedded to a d-dimensional vector via a linear layer and then concatenated with the original latent code, after each is normalized. This concatenated vector is then passed to the StyleGAN mapping network. The relationship between the number of layers of the StyleGAN mapping network and the size of the embedded sampled code d turns out to be crucial for StyleWSGAN convergence. When the mapping network is too shallow, as in the tuned CIFAR10 settings in [142], a large d can lead to training instability for StyleWSGAN and StyleInfoGAN, likely due to strong dependencies in the latent space of the mapping network output.

Separate optimizer settings are used for each loss term, and the learning rate for the Info term (added term of Equation 3.5) and the WSGAN term (added term of Equation 3.7 plus decay penalty) are set as a factor of 2/10 of the base learning rate in StyleGAN. This results in a learning rate of 0.0005 for the added WSGAN terms in the StyleWSGAN experiments, while a learning rate of 0.0025 is maintained for the original StyelGAN terms. Due to the use of different learning rates in the separate optimizers, the added loss terms are not scaled, and the hyper-parameters α, β are set to 1.

The CIFAR10 StyleWSGAN experiments largely follow the settings used in [142]: no style mixing, no path length regularization, no ResNet D. However, the depth of the mapping network is increasedd from 2 to 6, the size of the code embedding is decreased to 200, and training is continued until the discriminator has seen a total of 50M real images. A mapping network of depth 4, and a code embedding size of 50 also lead to good performance, performing only slightly worse measured by both FID and label model accuracy.

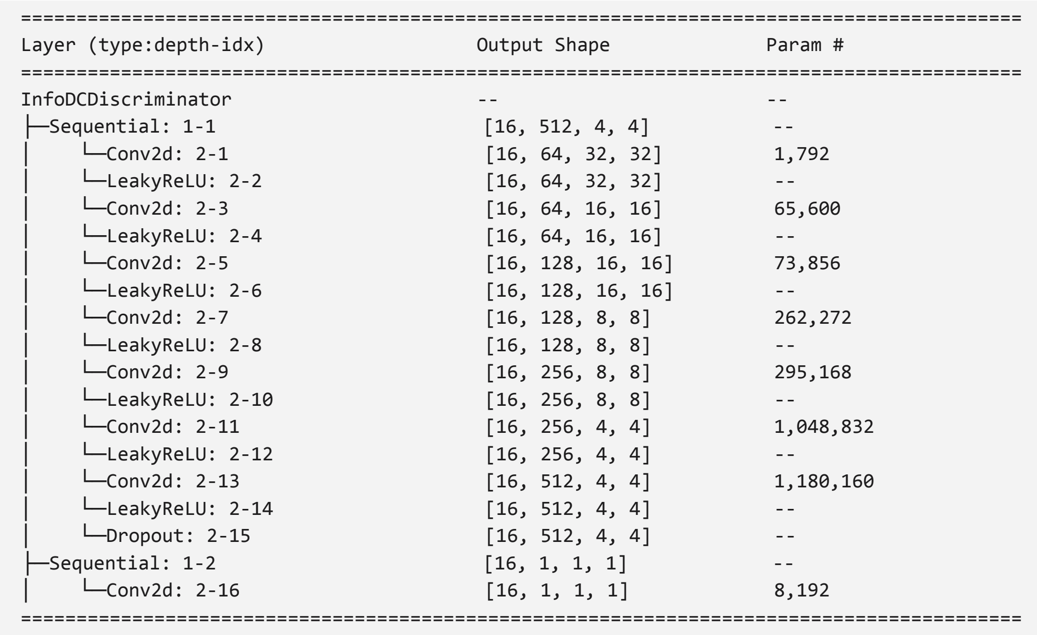
For the LSUN experiments, StyleWSGAN is trained until the discriminator has seen a total of 35M real images, and the baseline StyleGAN2-ADA until the discriminator has seen a total of 50M real images. The experiment largely follows the settings used for 256 x 256 images in [142], but style mixing and path length regularization are disabled. The size of the discrete code embedding is set to 50.

End Model Training For all datasets, a ResNet-18 [117] is trained for 100 epochs using Adam and a learning rate scheduler. The learning rate scheduler uses a small validation set to make adjustments to the learning rate.

Image Augmentation The following random image augmentation functions are used during DCGAN and endmodel training for color images: random crop and resize (cropping out a maximum height/width of 13%), random sharpness adjustment ($p = 0.2$), random color jitter, and random Gaussian blur ($p = 0.1$).

Label Models To compare to related work, the implementations of label models made available via WRENCH [289] were used.

Complexity WSGAN shares the same operations as InfoGAN and adds some additional steps on real samples that have at least one LF vote, which slightly increases the required computation. Recall that C denotes the number of classes, m the number of LFs, and x an image of a real sample. Further, let n_w denote the number of samples that have at least one weak label vote from any LF, let q denote the number of steps required for a forward pass through Q to obtain image features and the discrete code prediction, and a denote the number of steps for a forward pass through the MLP A. For a forward pass, WSGAN increases the complexity compared to InfoGAN in each epoch by $\Theta(n_w(a + q + m + 2C^2 + C(m + 8)))$. Note that q may be eliminated for the forward pass through careful implementation as the image features are already obtained for the basic InfoGAN update. In the experiments of this work, the computational overhead, including for additional data loading of the LFs, lead to a modest increase in runtime (measured in bps, denoting batches per second) of the weakly supervised WSGAN over the unsupervised InfoGAN, as follows. InfoGAN:



```
===========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
===========================================================================================
InfoDCDiscriminator                      --                        --
├─Sequential: 1-1                        [16, 512, 4, 4]           --
│    └─Conv2d: 2-1                       [16, 64, 32, 32]          1,792
│    └─LeakyReLU: 2-2                    [16, 64, 32, 32]          --
│    └─Conv2d: 2-3                       [16, 64, 16, 16]          65,600
│    └─LeakyReLU: 2-4                    [16, 64, 16, 16]          --
│    └─Conv2d: 2-5                       [16, 128, 16, 16]         73,856
│    └─LeakyReLU: 2-6                    [16, 128, 16, 16]         --
│    └─Conv2d: 2-7                       [16, 128, 8, 8]           262,272
│    └─LeakyReLU: 2-8                    [16, 128, 8, 8]           --
│    └─Conv2d: 2-9                       [16, 256, 8, 8]           295,168
│    └─LeakyReLU: 2-10                   [16, 256, 8, 8]           --
│    └─Conv2d: 2-11                      [16, 256, 4, 4]           1,048,832
│    └─LeakyReLU: 2-12                   [16, 256, 4, 4]           --
│    └─Conv2d: 2-13                      [16, 512, 4, 4]           1,180,160
│    └─LeakyReLU: 2-14                   [16, 512, 4, 4]           --
│    └─Dropout: 2-15                     [16, 512, 4, 4]           --
├─Sequential: 1-2                        [16, 1, 1, 1]             --
│    └─Conv2d: 2-16                      [16, 1, 1, 1]             8,192
===========================================================================================
```

Figure C.1: The DCGAN discriminator architecture used in experiments with 32 x 32 images.

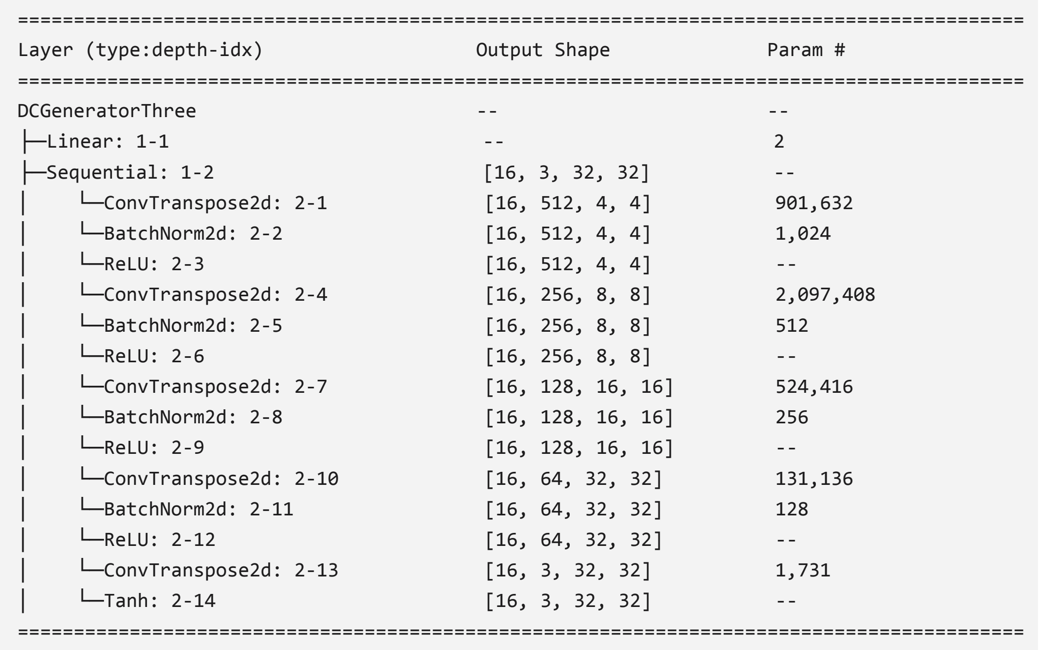
14bps, WSGAN Encoder: 7.8bps, WSGAN Vector: 8.6bps (NVIDIA RTX A6000, batch size 16). In terms of parameters, WSGAN shares the same generator G and discriminator components D,Q as InfoGAN, and adds additional label model parameters. The overall number of parameters in the experiments with 32 x 32 images are: InfoGAN 6.7M, WSGAN 8.8M.

C.2 Dataset Details

- *CIFAR10* contains 32x32 color images of 10 different classes. Two different subsets of CIFAR10 are created. One set (used for experiments CIFAR10-C,D) uses the full training set of CIFAR10 (minus 300 samples held out for downstream validation), while the second (used for experiments CIFAR10-A,B,E,F) is a random subset of 30,000 training images.

- *MNIST* and *FashionMNIST* both contain 28x28 gray-scale images, which were resized to 32x32. For both, a random sample of 30,000 images is taken from the training data. SSL-based labeling functions are fine-tuned on small, random subsets of the remaining training data of each dataset.

- *GTSRB* contains 64x64 color images of German traffic signs. 22,640 random images from the full training dataset are used for the experiments, while random



```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
DCGeneratorThree                         --                        --
├─Linear: 1-1                             --                        2
├─Sequential: 1-2                         [16, 3, 32, 32]           --
│    └─ConvTranspose2d: 2-1               [16, 512, 4, 4]           901,632
│    └─BatchNorm2d: 2-2                   [16, 512, 4, 4]           1,024
│    └─ReLU: 2-3                          [16, 512, 4, 4]           --
│    └─ConvTranspose2d: 2-4               [16, 256, 8, 8]           2,097,408
│    └─BatchNorm2d: 2-5                   [16, 256, 8, 8]           512
│    └─ReLU: 2-6                          [16, 256, 8, 8]           --
│    └─ConvTranspose2d: 2-7               [16, 128, 16, 16]         524,416
│    └─BatchNorm2d: 2-8                   [16, 128, 16, 16]         256
│    └─ReLU: 2-9                          [16, 128, 16, 16]         --
│    └─ConvTranspose2d: 2-10              [16, 64, 32, 32]          131,136
│    └─BatchNorm2d: 2-11                  [16, 64, 32, 32]          128
│    └─ReLU: 2-12                         [16, 64, 32, 32]          --
│    └─ConvTranspose2d: 2-13              [16, 3, 32, 32]           1,731
│    └─Tanh: 2-14                         [16, 3, 32, 32]           --
==========================================================================================
```

Figure C.2: The DCGAN generator architecture used in experiments with 32x32 images.

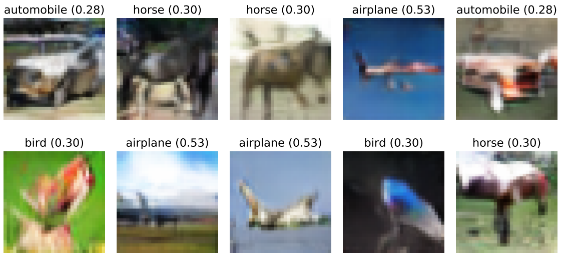


Figure C.3: Some synthetic images and pseudolabels generated by the proposed WSGAN with a DCGAN base-architecture, learned from weakly supervised CIFAR10. Note that WSGAN is able to generate images and estimate their labels, even for images where no weak supervision sources provide information (see end of Section 3.2.2 for details).

subsets of the remaining images in the original training data are used to finetune the SSL-based labeling functions.

- The original *DomainNet* [209] dataset contains 345 classes of images in 6 different domains [1]. As the dataset for the proposed work, following [185] the images in the real domain are used and the 10 classes with the largest number of instances in this domain are selected. Because of the small size of the resulting dataset, images are resized to 32 x 32.

- *Animals with Attributes 2* (AwA2) [185] is an image dataset with known general attributes for each class, divided into 40 seen and 10 unseen classes. Because of the small size of the resulting dataset once LFs are created, the images are resized to 32 x 32 in the experiments.

- *LSUN scene categories* A random small set of up to 2,000 held-out images is used to finetune SSL-based labeling functions.

C.2.1 Labeling Function Details

- *Synthetic*: based on the true class label, synthetic, unipolar LFs were created via the following procedure: for each LF, a class label, an error rate, and a propensity (i.e., the percentage of samples where the LF casts a vote, also referred to as coverage) are sampled. Given the target label, true positives and false positives are then sampled at random to achieve the desired LF accuracy and propensity.

- *Domain transfer*: these LFs are used in the DomainNet dataset experiments. Following [185], weak supervision sources are derived for a multiclass classification task of the real images contained in the DomainNet [209] dataset. First, the target domain is set to real images and the 10 classes with the largest number of instances in this domain are selected. As LFs, classifiers are trained using the selected classes within the remaining five domains, and these trained classifiers are applied to the unseen images in the target domain of real images to obtain imperfect labels.

- *Attribute heuristics*: two sets of LFs are created for the for the Animals with Attributes 2 (AwA2) [185] image classification dataset. Following [186, 185], one-vs-rest attribute classifiers are trained using the 40 seen classes of the AwA2 dataset. These classifiers are applied to the 10 unseen classes to produce weak attribute labels. At this stage, attribute classifiers which perform worse than random are discarded. An 85%/5%/10% train/validation/test split of the 10 unseen classes is created to define decision trees to produce weak labels on the

[1] Real, painting, sketch, clipart, infograph, quickdraw.

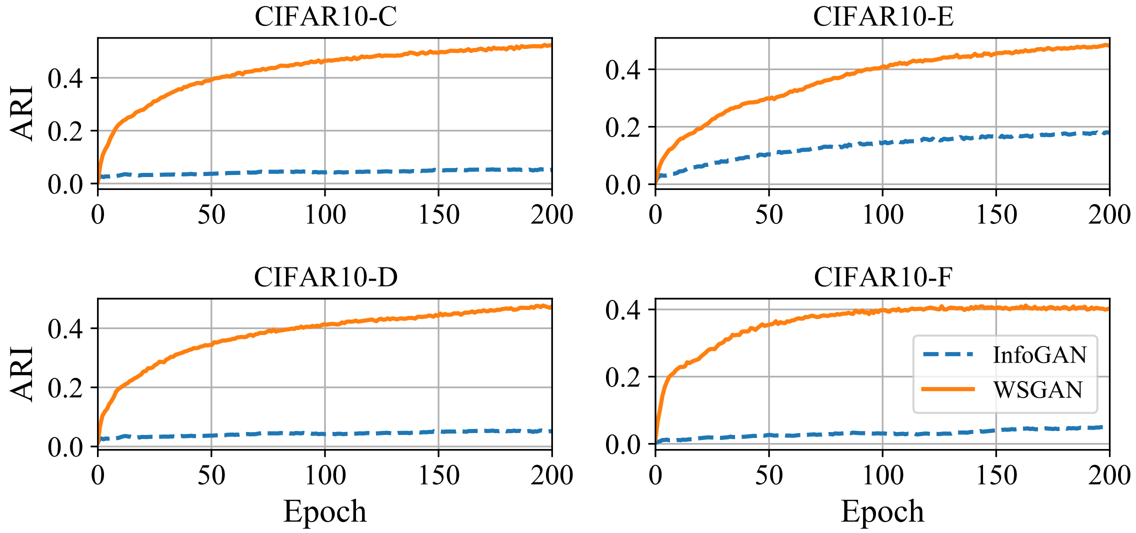


Figure C.4: ARI for additional CIFAR10 experiments. The plots show the ARI between the unobserved class label y and the discrete code prediction by the auxiliary model $Q(x)$ on real image x, during training. Weak supervision allows WSGAN to better uncover the latent class structure compared to an unsupervised InfoGAN.

bases of weak attribute predictions. The final 29 unipolar LFs for *AwA2-A* are created by training 3 one-vs-rest decision trees per each of the 10 classes on 100 random samples from the training set. To create a slightly easier set, the 32 unipolar LFs used in *AwA2-B* are created by training 80 decision trees, retaining one random tree specializing in each class, and then selecting all remaining ones where validation accuracy is higher than 0.65.

- *SSL-based*: The base representations are learned on unsupervised ImageNET with SimCLR [45]. The trained network is used to obtain features for the image datasets. Then, shallow MLP networks are trained on a few hundred held-out samples to predict a randomly sampled target label at a randomly sampled target accuracy. The accuracy is validated to be within range of the target accuracy on another small amount of held-out data. Thus, during their creation these unipolar LFs are never trained or evaluated on the WSGAN training data or the downstream test data.

C.3 Additional Experiments

C.3.1 WSGAN with a DCGAN Base-architecture

For further evaluation of WSGAN with a DCGAN base architecture, additional weakly supervised image datasets were created based on CIFAR10, by varying the

Table C.1: Additional datasets and labeling function (LF) characteristics used to evaluate the proposed WSGAN model. Acc denotes accuracy, while Coverage denotes the number of samples where the LF does not abstain.

Dataset	#Classes	#LFs	#Samples	Mean LF Acc	Min LF Acc	Max LF Acc	Mean Coverage	LF Type
CIFAR10-C	10	20	49,700	0.747	0.621	0.879	0.048	Synthetic
CIFAR10-D	10	40	49,700	0.760	0.621	0.898	0.052	Synthetic
CIFAR10-E	10	40	30,000	0.761	0.624	0.896	0.056	Synthetic
CIFAR10-F	10	40	30,000	0.728	0.531	0.912	0.046	SSL, finetuning

Table C.2: Additional datasets to evaluate WSGAN with a DCGAN base architecture. This table shows the average posterior accuracy of various label models on training samples with at least one LF vote. The best result is highlighted in **blue** and the second best result in **bold**.

Dataset	MV	DawidSkene	MeTaL	FS	Snorkel	WSGAN-Vector	WSGAN-Encoder
CIFAR10-C	0.762	**0.778**	0.751	0.764	0.757	**0.778**	**0.796**
CIFAR10-D	0.831	**0.861**	0.819	0.805	0.812	0.854	**0.865**
CIFAR10-E	0.865	**0.902**	0.845	0.827	0.849	0.898	**0.917**
CIFAR10-F	0.687	0.601	0.682	0.677	0.678	**0.691**	**0.702**

Table C.3: Additional datasets: color image generation quality measured by average Fréchet Inception Distance (FID). The best scores for each dataset are highlighted in **blue**.

Dataset	InfoGAN	WSGAN-V	WSGAN-E
CIFAR10-C	33.64	**24.11**	26.00
CIFAR10-D	33.64	24.09	**23.78**
CIFAR10-E	28.93	**21.97**	22.63
CIFAR10-F	33.50	24.59	**22.54**

number of samples and the type of labeling function, see dataset details in Table C.1. The proposed WSGAN approach outperforms related approaches in these experiments as well. The label model accuracy results are shown in Table C.2, while additional metrics including F1 are shown in Section C.5. Image generation quality results are provided in Table C.3. Finally, a comparison between the latent discrete variable of WSGAN and InfoGAN is given in Figure C.4, which shows how the ARI evolves between the unobserved class labels and the latent discrete variable modeled by auxiliary model Q.

C.4 Additional Baselines

Two additional baselines are created for comparison. First, a generative model that is conditioned on pseudolabel information is created with the aim of improving image generation performance; pseudolabels provided by established weak-supervision label models are used in this role. Second, a basic generative model is used to produce synthetic samples that augment a downstream classifier (with weak labels provided by outputs of weak supervision sources applied to the synthetic images). These two baselines represent the straightforward way to use weak supervision to improve generative modeling (and vice-versa). Experiments with these models showed that such naive combinations struggle compared to the proposed WSGAN approach.

C.4.1 Conditional Image Generation with Pseudolabels or Raw Weak Supervision Votes

As an additional GAN baseline to compare the proposed WSGAN, an Auxiliary Classifier Generative Adversarial Network (ACGAN) [203] is adapted to be conditioned on pseudolabels provided by a label model. The ACGAN is run on all data, but the auxiliary loss on real data with pseudolabels is only used for samples where at least one labeling function does not abstain. Two versions were created: (1) using probabilistic pseudolabels with a soft cross-entropy loss, and (2) using *hard/crisp* labels with a cross-entropy loss. To provide the strongest possible baseline in this experiment, the pseudolabels are obtained via the Dawid-Skene label model as it attains the best performance on average over all datasets compared to other related label models. Results are provided in Table C.4, showing that this baseline approach is frequently unable to overcome the noise in the pseudolabels to improve over the InfoGAN results, and that it does not perform better than WSGAN with an encoder. Furthermore, the models were difficult to train and converged rarely.

We also attempted to train different types of conditional GANs (ACGAN and a GAN with projection discrimination) conditioned on the raw weak supervision votes, but were unable to obtain reasonable performance as the models failed to converge.

C.4.2 Data Augmentation for Downstream Classification with Synthetic Images

In this experiment, the training set for a downstream classifier is augmented with synthetic images. As baselines, synthetic images $\tilde{x}$ were generated with an InfoGAN, and the image labeling functions λ were then applied to the generated images to obtain LF votes $\lambda(\tilde{x})$. Pseudolabels for the synthetic images are then obtained by fitting label models to the real training data and then applying the label models to the labeling function outputs on the synthetic data. Table C.5 compares InfoGAN + Snorkel and InfoGAN + DawidSkene baselines to the improvements in test accuracy obtained

Table C.4: A comparison to using an ACGAN with pseudolabels. Image generation quality is measured by average Fréchet Inception Distance (FID). The best scores for each dataset are highlighted in **blue**.

Dataset	InfoGAN	ACGAN	ACGAN (crisp)	WSGAN-V	WSGAN-E
AwA2 - A	41.62	51.32	47.21	36.74	**34.71**
AwA2 - B	41.62	53.73	50.03	36.79	**34.52**
DomainNet	51.88	61.96	47.32	51.16	**45.6**
CIFAR10-A	28.93	79.15	25.53	25.7	**22.71**
CIFAR10-C	33.64	36.53	26.61	**24.11**	26.0
CIFAR10-D	33.64	36.81	45.1	24.09	**23.78**
CIFAR10-E	28.93	80.43	33.05	**21.97**	22.63

Table C.5: Baseline comparisons using an InfoGAN to create synthetic images, applying LFs to the synthetic images, and then using established label models to synthesize the weak labels in to a pseudolabel resulting in weakly labeled fake images. The table shows the change in test accuracy by augmenting the downstream classifier training data with such 1,000 synthetic images and corresponding pseudo labels. Experiments are conducted on a subset of the datasets where labeling functions can be applied to synthetic images.

Dataset	WSGAN	InfoGAN + Snorkel	InfoGAN + DawidSkene
AwA2 - A	0.79%	-0.63%	-1.26%
AwA2 - B	3.90%	-1.01%	-1.77%
DomainNet	1.50%	0.02%	-3.14%

by using WSGAN and shows that this naive baseline does not provide performance improvements in downstream accuracy.

Table C.6: This table includes standard deviations for the posterior accuracy of various label models on training samples with at least one LF vote, computed over five random runs. Due to a limited computational budget, it was not feasible to accumulate five runs for all datasets and model combinations.

Dataset	DawidSkene	MeTaL	FS	Snorkel	WSGAN-Encoder
AwA2 - A	0.607(±0.029)	0.632(±0.002)	0.615(±0.003)	0.641(±0.001)	**0.681**(±0.011)
DomainNet	**0.658**(±0.000)	0.487(±0.004)	0.635(±0.000)	0.499(±0.015)	0.643(±0.003)
MNIST	0.729(±0.000)	0.766(±0.001)	0.773(±0.000)	0.766(±0.001)	**0.813**(±0.004)
FashionMNIST	0.717(±0.002)	0.730(±0.001)	0.734(±0.001)	0.729(±0.001)	**0.744**(±0.002)
GTSRB	0.619(±0.001)	**0.815**(±0.002)	0.679(±0.001)	**0.814**(±0.000)	**0.823**(±0.001)
CIFAR10-A	**0.850**(±0.001)	0.806(±0.001)	0.800(±0.000)	0.807(±0.002)	**0.874**(±0.002)
CIFAR10-B	0.677(±0.000)	0.708(±0.001)	0.708(±0.000)	0.707(±0.000)	**0.731**(±0.004)

Table C.7: Weighted mean average precision of various label models on training samples with at least one LF vote. The best result are highlighted in **blue** and the second best result in **bold**.

Dataset	MV	DawidSkene	MeTaL	FS	Snorkel	WSGAN-Vector	WSGAN-Encoder
AwA2 - A	0.616	0.661	0.653	0.627	0.653	**0.672**	**0.737**
AwA2 - B	0.591	0.652	0.662	0.642	0.668	**0.681**	**0.743**
DomainNet	0.599	**0.702**	0.630	0.654	0.621	0.679	**0.795**
MNIST	0.684	0.772	0.784	0.765	0.785	**0.792**	**0.870**
FashionMNIST	0.620	0.712	0.691	0.686	0.692	**0.703**	**0.742**
GTSRB	0.718	0.731	0.761	0.714	**0.772**	0.768	**0.808**
CIFAR10-A	0.796	0.866	0.855	0.838	0.854	**0.878**	**0.912**
CIFAR10-B	0.594	0.659	0.658	0.631	0.666	**0.678**	**0.732**
CIFAR10-C	0.664	0.763	0.758	0.737	0.751	**0.780**	**0.825**
CIFAR10-D	0.788	0.896	0.889	0.876	0.878	**0.901**	**0.908**
CIFAR10-E	0.880	**0.954**	0.942	0.924	0.940	0.950	**0.959**
CIFAR10-F	0.561	0.658	0.66	0.647	0.659	**0.675**	**0.708**

C.5 Additional Metrics

This section provides additional metrics for the label model comparisons shown in Table 3.5 in the main paper. Again, results are averaged over 4 random runs. Table C.8 shows the weighted F1 score, an average over all classes weighted by the support of each class. Table C.7 shows weighted mean average precision, a metric that summarizes the precision-recall curve across all classes. The average precision is computed individually for each class (one vs. rest) and the scores are then aggregated by summing them weighted by the support of each class to produce the weighted mean average precision score.

Table C.8: Weighted F1 score of various label models on training samples with at least one LF vote. The F1 is computed separately for each class and then averaged weighted by the support of each class. The best result is highlighted in **blue** and the second best result in **bold**.

Dataset	MV	DawidSkene	MeTaL	FS	Snorkel	WSGAN-Vector	WSGAN-Encoder
AwA2 - A	0.641	**0.665**	0.62	0.619	0.636	0.637	**0.684**
AwA2 - B	0.604	**0.661**	0.58	0.597	0.593	**0.664**	**0.672**
DomainNet	0.603	**0.655**	0.443	0.622	0.468	**0.654**	**0.634**
MNIST	0.756	0.716	0.746	0.755	0.746	**0.764**	**0.795**
FashionMNIST	0.706	0.691	0.698	0.705	0.698	**0.710**	**0.715**
GTSRB	**0.802**	0.616	0.800	0.628	0.799	**0.801**	**0.811**
CIFAR10-A	0.824	**0.850**	0.797	0.796	0.798	**0.851**	**0.872**
CIFAR10-B	0.712	0.672	0.702	0.703	0.702	**0.720**	**0.727**
CIFAR10-C	0.726	**0.741**	0.723	0.713	0.718	**0.738**	**0.759**
CIFAR10-D	0.809	**0.839**	0.791	0.784	0.781	0.834	**0.844**
CIFAR10-E	0.864	**0.901**	0.843	0.825	0.839	**0.899**	**0.916**
CIFAR10-F	0.684	0.609	0.677	0.675	0.674	**0.688**	**0.699**

C.6 Theoretical Justification

This section provides additional setup details and proofs for the two theoretical claims of this work.

C.6.1 Claim (1)

Our goal is to derive a generalization bound; that is, an upper bound on $|\hat{\mathbb{R}}_{\hat{\mathcal{D}}} - \mathbb{R}_{\mathcal{D}}|$. In words, this is the gap between the loss on a sample drawn from the true distribution and the empirical loss we obtained by training on the weakly-supervised dataset with unlabeled data sampled from the generative model.

Mixture of Gaussians Recall that D is the joint distribution of the unlabeled and labeled points. Let us call the unlabeled data marginal distribution D_X. Then, we make the assumption that D_X is a mixture of k Gaussians. Here, there is some relationship between the mixtures and the two classes, but we need not further specify it. Using the result [9], we get that the number of samples needed to learn D_X up to ε in total variation distance is $\tilde{\Theta}(kd^2/\varepsilon^2)$.

Note that in fact this expression hides some polylogarithmic terms. However, for simplicity, we are going to ignore these terms and just pretend that the necessary bound is $c_G kd^2/\varepsilon^2$, where c_G is some constant for learning a density.

Based on this, we will make the following assumption. We perform density estimation on n_1 samples from D_X and obtain some model g such that distribution of g

(we will abuse notation and just refer to this as the model itself g) and D_X satisfies

$$d_{\text{TV}}(D_X, g) \leq d\sqrt{\frac{c_G k}{n_1}}. \tag{C.1}$$

So now we have control over one marginal (the unlabeled data). Let us work on the conditional term next.

Majority Vote For simplicity, let us assume that we use majority vote as the aggregation scheme for the m labeling functions. We make the following assumptions. The labeling functions have accuracy $1/2 + \alpha$, for some $\alpha \in (0, 1/2]$, in the following sense. For any datapoint (X, Y), the probability of a labeling function guessing the value of Y correctly is $1/2 + \alpha$, and the probability of any guessing wrong is $1/2 - \alpha$. This holds for all values of X. Note: these are very strong assumptions.

The probability that we make a mistake, e.g., that majority vote aggregates votes to 0 when $Y = 1$ or vice-versa is given by the binomial CDF $F(m/2, m, \alpha + 1/2)$, which has the following simple bound that follows from Hoeffding's inequality,

$$F(m/2, m, \alpha + 1/2) \leq \exp\left(-2m\left(\alpha + 1/2 - \frac{m/2}{m}\right)^2\right) = \exp(-2m\alpha^2).$$

With the above, as $D_{Y|X}$ is a Bernoulli random variable, we can directly upper bound the total variation distance between $D_{Y|X}$ and $D_{\hat{Y}|X}$:

$$d_{\text{TV}}(D_{Y|X}, D_{\hat{Y}|X}) \leq \exp(-2m\alpha^2). \tag{C.2}$$

Joint Distribution Now we have some control over the generative model's error (from the density estimation bound) and some control over the label recovery (from the above bound resulting from majority vote). Now we put it together. First, we write down some useful inequalities between the total variation distance and the Hellinger distance [84] (Prop 2.10). These are, for densities p, q,

$$D_{\text{hel}}(p, q) \leq \sqrt{2d_{\text{TV}}(p, q)} \tag{C.3}$$

and

$$d_{\text{TV}}(p, q) \leq D_{\text{hel}}(p, q)\sqrt{1 - D_{\text{hel}}(p, q)^2/4}. \tag{C.4}$$

We use p as the density for $\mathcal{D}$ and q as the density for $\hat{\mathcal{D}}$, and write $p = p_1(x)p_2(y|x)$, $q = q_1(x)q_2(y|x)$. First, using (C.1) and (C.3), we have that

$$D_{\text{hel}}(\mathcal{D}_X, g) \leq \sqrt{2d_{\text{TV}}(\mathcal{D}_X, g)} \leq \left(\frac{4c_G k d^2}{n_1}\right)^{\frac{1}{4}}. \tag{C.5}$$

Then, using (C.2) and (C.3), we get

$$D_{\text{hel}}(D_{Y|X}, D_{\hat{Y}|X}) \leq \sqrt{2d_{\text{TV}}(D_{Y|X}, D_{\hat{Y}|X})} \leq \sqrt{2\exp(-2m\alpha^2)} = \sqrt{2}\exp(-m\alpha^2). \tag{C.6}$$

Next,

$$\begin{aligned}
D_{\text{hel}}(\mathcal{D}, \hat{\mathcal{D}}) &= \int\int (\sqrt{p_1(x)p_2(y|x)} - \sqrt{q_1(x)Q_2(y|x)})^2 dydx \\
&= \int\int \left(p_1(x)p_2(y|x) + q_1(x)q_2(y|x) - 2\sqrt{p_1(x)p_2(y|x)q_1(x)q_2(y|x)}\right) dydx \\
&= 2 - 2\int\int \sqrt{p_1(x)p_2(y|x)q_1(x)q_2(y|x)} dydx \\
&= 2 - 2\int \sqrt{p_1(x)q_1(x)} \left(1 - \frac{1}{2}D_{\text{hel}}(p_2, q_2)\right) dx \\
&\leq 2 - 2\int \sqrt{p_1(x)q_1(x)} \left(1 - \frac{\sqrt{2}}{2}\exp(-m\alpha^2)\right).
\end{aligned}$$

Note that here, we use the fact that our bound holds for all conditional distributions regardless of x. Continuing,

$$\begin{aligned}
D_{\text{hel}}(\mathcal{D}, \hat{\mathcal{D}}) &\leq 2 - 2\int \sqrt{p_1(x)q_1(x)} \left(1 - \frac{\sqrt{2}}{2}\exp(-m\alpha^2)\right) \\
&= 2 - 2(1 - \frac{1}{2}D_{\text{hel}}(p_1, q_1) \left(1 - \frac{\sqrt{2}}{2}\exp(-m\alpha^2)\right) \\
&\leq 2 - 2\left(1 - \frac{1}{2}\left(\frac{4c_G kd^2}{n_1}\right)^{\frac{1}{4}}\right)\left(1 - \frac{\sqrt{2}}{2}\exp(-m\alpha^2)\right) \\
&= \left(\frac{4c_G kd^2}{n_1}\right)^{\frac{1}{4}} + \sqrt{2}\exp(-m\alpha^2) - \left(\frac{c_G kd^2}{n_1}\right)^{\frac{1}{4}}\exp(-m\alpha^2).
\end{aligned}$$

Now we apply (C.4) to get the bound back into the total variation distance setting. We have

$$\begin{aligned}
d_{\text{TV}}(\mathcal{D}, \hat{\mathcal{D}}) &\leq D_{\text{hel}}(\mathcal{D}, \hat{\mathcal{D}})\sqrt{1 - D_{\text{hel}}(\mathcal{D}, \hat{\mathcal{D}})^2/4} \leq D_{\text{hel}}(\mathcal{D}, \hat{\mathcal{D}}) \\
&\leq \left(\frac{4c_G kd^2}{n_1}\right)^{\frac{1}{4}} + \sqrt{2}\exp(-m\alpha^2) - \left(\frac{c_G kd^2}{n_1}\right)^{\frac{1}{4}}\exp(-m\alpha^2) \\
&\leq \left(\frac{4c_G kd^2}{n_1}\right)^{\frac{1}{4}} + \sqrt{2}\exp(-m\alpha^2).
\end{aligned} \tag{C.7}$$

Bounding the Risk The final task is to bound the risk. First, suppose we are training a classifier chosen from a function class $\mathcal{F}$, trained on n_2 independently-drawn data points. Then, a standard result is that with probability at least $1-\delta$,

$$\sup_{f\in\mathcal{F}} |\hat{\mathbb{R}}_{\mathcal{D}}(f) - \mathbb{R}_{\mathcal{D}}(f)| \leq 2\mathfrak{R} + \sqrt{\frac{\log(1/\delta)}{2n_2}}. \tag{C.8}$$

Here, $\mathfrak{R}$ is the Rademacher complexity of the function class. However, the above is for training on samples from the true distribution. Instead, we can write

$$\begin{aligned} |\hat{\mathbb{R}}_{\hat{\mathcal{D}}} - \mathbb{R}_{\mathcal{D}}| &= |\hat{\mathbb{R}}_{\hat{\mathcal{D}}} - \mathbb{R}_{\hat{\mathcal{D}}} + \mathbb{R}_{\hat{\mathcal{D}}} - \mathbb{R}_{\mathcal{D}}| \\ &\leq |\hat{\mathbb{R}}_{\hat{\mathcal{D}}} - \mathbb{R}_{\hat{\mathcal{D}}}| + |\mathbb{R}_{\hat{\mathcal{D}}} - \mathbb{R}_{\mathcal{D}}|. \end{aligned}$$

For the right-hand term, we have the following:

$$\begin{aligned} |\mathbb{R}_{\hat{\mathcal{D}}} - \mathbb{R}_{\mathcal{D}}| &= |\int \ell(f(x),y)|p(x,y) - q(x,y)|d\mu \\ &\leq B_\ell d_{\mathrm{TV}}(\hat{\mathcal{D}}, \mathcal{D}). \end{aligned}$$

Then, putting this together with the expression in (C.7) into (C.8), we get that, with probability at least $1-\delta$,

$$\begin{aligned} \sup_{f\in\mathcal{F}} |\hat{\mathbb{R}}_{\hat{\mathcal{D}}}(f) - \mathbb{R}_{\mathcal{D}}(f)| &\leq (\sup_{f\in\mathcal{F}} |\hat{\mathbb{R}}_{\hat{\mathcal{D}}} - \mathbb{R}_{\hat{\mathcal{D}}}| + |\mathbb{R}_{\hat{\mathcal{D}}} - \mathbb{R}_{\mathcal{D}}|) \\ &\leq 2\mathfrak{R} + \sqrt{\frac{\log(1/\delta)}{2n_2}} + B_\ell d_{\mathrm{TV}}(\hat{\mathcal{D}}, \mathcal{D}) \\ &\leq 2\mathfrak{R} + \sqrt{\frac{\log(1/\delta)}{2n_2}} + B_\ell \left(\frac{4c_G k d^2}{n_1}\right)^{\frac{1}{4}} + B_\ell\sqrt{2}\exp(-m\alpha^2). \end{aligned} \tag{C.9}$$

Interpreting the Bound In (C.9), we saw that

$$\sup_{f\in\mathcal{F}} |\hat{\mathbb{R}}_{\hat{\mathcal{D}}}(f) - \mathbb{R}_{\mathcal{D}}(f)| \leq 2\mathfrak{R} + \sqrt{\frac{\log(1/\delta)}{2n_2}} + B_\ell \left(\frac{4c_G k d^2}{n_1}\right)^{\frac{1}{4}} + B_\ell\sqrt{2}\exp(-m\alpha^2).$$

Now let's interpret this result piece-by-piece. The terms are the following

- The Rademacher complexity of the function class, which is present in the standard generalization bound.
- An estimation error term as a function of how much data we have to train our classifier n_2. It has the standard rate $1/\sqrt{n_2}$. Again, this is standard in any bound.

- A penalty term due to the generative model usage. It tells us how much we lose by training on generated data rather than (unlabeled) data from the true distribution. It scales as $n_1^{-1/4}$, where n_1 is the number of samples of unlabeled data used to train the generative model. Note also the dependence on the number of mixture components and dimension.

- A penalty term due to weak supervision. It tells us what we lose by using estimated (pseudo)labels rather than true labels; we note that the penalty scales exponentially in the number of labeling functions m, but is slowed down by small α, as our accuracies are α better than random.

C.6.2 Claim (2)

The proof of claim (2) uses the setting of [249], which introduces RCGAN. RCGAN is a conditional GAN architecture that corrupts the label before passing them to the discriminator by passing the true labels through a noisy channel. The authors provide a multiplicative approximation bound between the GAN loss under the unobserved true labels and the loss under the noisy labels. This noisy channel model acts as a nice model of the label generating process of weak supervision. Using this noisy channel model, we can control the amount of label corruption to match that of weak supervision.

Following the setup of [249], we define a function that multiplies a one-hot encoded true label vector by a right-stochastic matrix $C \in \mathbb{R}^{2\times 2}$ where $C_{i,j} = P(\tilde{y}_j|y_i)$—this is our noisy channel. This induces a joint distribution $\widetilde{P}_{X,\tilde{Y}}$ for the examples x and noisy labels $\tilde{y}$ from the conditional distribution defined by C. We restate the theorems of interest from [249] here, and proceed to adapt them to our problem setting.

Theorem 1. *(Multiplicative bound on the total variation distance from [249].) Let $P_{X,Y}$ and $Q_{X,Y}$ be two distributions over $\mathcal{X} \times \{0,1\}$ and let $\widetilde{P}_{X,\tilde{Y}}$ and $\widetilde{Q}_{X,\tilde{Y}}$ be the corresponding distributions with noisy labels from C. If C is full-rank, then*

$$d_{TV}(\widetilde{P},\widetilde{Q}) \leq d_{TV}(P,Q) \leq \|C^{-1}\|_\infty \, d_{TV}(\widetilde{P},\widetilde{Q}). \tag{C.10}$$

Theorem 1 says that the total variation distance between the true noisy distribution and the noisy generated distribution from RCGAN approximate its noiseless counterpart up to a factor of $\|C^{-1}\|_\infty$. The goal here is to construct $C_{\epsilon_{\text{MV}}}$ to model the noise from weak supervision (in particular, majority vote) and show that it leads to a tighter bound than when we directly plug in the labels from a single LF into Theorem 1. To begin, consider the following parameterization of C, with $\epsilon \in (0, 1/2)$:

$$C_\epsilon = I_2 + \begin{bmatrix} -\epsilon & \epsilon \\ \epsilon & -\epsilon \end{bmatrix} = \begin{bmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{bmatrix}. \tag{C.11}$$

Here, ϵ denotes the labeling error for each class. Given this parameterization, we obtain the following expression for $\|C_\epsilon^{-1}\|_\infty$.

$$\|C_\epsilon^{-1}\|_\infty = \left\| \begin{bmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{bmatrix}^{-1} \right\|_\infty \tag{C.12}$$

$$= |((1-\epsilon)^2 - \epsilon^2)^{-1}| \left\| \begin{bmatrix} 1-\epsilon & -\epsilon \\ -\epsilon & 1-\epsilon \end{bmatrix} \right\|_\infty \tag{C.13}$$

$$= (1-2\epsilon)^{-1} \tag{C.14}$$

Note that C_ϵ is full-rank as it has a finite inverse. It is also clear that $\|C_\epsilon^{-1}\|_\infty$ is a monotonically increasing function of ϵ. That is to say that if we do something to decrease the labeling error ϵ, then $\|C_\epsilon^{-1}\|_\infty$ also decreases and we obtain a tighter bound. We will go on to derive an expression for the labeling error under majority vote with m LFs, ϵ_{MV}, and show that it is smaller than the labeling error from a single LF, ϵ_λ. Namely, we want find a condition where $\epsilon_{\text{MV}} \leq \epsilon_\lambda$ holds and that majority vote leads to an improved Theorem 1 bound.

Proposition 1. *(Total variation version.) Let ϵ_{MV} be the labeling error from majority vote from m LFs, where $m \geq \frac{\log(1/\epsilon_\lambda)}{2\left(\frac{1}{2}-\epsilon_\lambda\right)^2}$, whose individual labeling errors are each ϵ_λ. Then the following holds*

$$d_{TV}(\widetilde{P}_{MV}, \widetilde{Q}_{MV}) \leq d_{TV}(P, Q) \leq \|C_{\epsilon_{MV}}^{-1}\|_\infty \; d_{TV}(\widetilde{P}_{MV}, \widetilde{Q}_{MV}) \leq \|C_{\epsilon_\lambda}^{-1}\|_\infty \; d_{TV}(\widetilde{P}_{MV}, \widetilde{Q}_{MV}).$$

Proof. We begin by deriving an upper bound on ϵ_{MV}. We have LFs $\{\lambda_i\}_{i=1}^m$ that each produce incorrect predictions with probability $\epsilon_\lambda = \frac{1}{2} - \alpha$, using α as defined in Claim (1). Now, we need to show that the probability of producing incorrect predictions using majority vote with more label functions, $\{\lambda_i\}_{i=1}^m$, has error $\epsilon_{\text{MV}} \leq \epsilon_\lambda$. Define the event that λ_i is incorrect as follows: $z_i = \mathbb{I}[\lambda_i \neq y]$, then $\mathbb{E}[z_i] = \epsilon_\lambda$. Using this, we apply Hoeffding's bound to ϵ_{MV}.

$$\epsilon_{\text{MV}} = P\left(\sum_{i=1}^m z_i - m\epsilon_\lambda \geq \frac{m}{2} - m\epsilon_\lambda\right) \tag{C.15}$$

$$\leq \exp\left(\frac{-2\left(\frac{m}{2} - m\epsilon_\lambda\right)^2}{m}\right) \tag{C.16}$$

$$= \exp\left(-2m\left(\frac{1}{2} - \epsilon_\lambda\right)^2\right). \tag{C.17}$$

Next, we plug the bound from (C.17) into (C.14) to obtain the following expression for $\|C_{\epsilon_{\text{MV}}}^{-1}\|_\infty$.

$$\|C_{\epsilon_{\text{MV}}}^{-1}\|_\infty = (1 - 2\epsilon_{\text{MV}})^{-1} \tag{C.18}$$

$$\leq \left(1 - 2\exp\left(-2m\left(\frac{1}{2} - \epsilon_\lambda\right)^2\right)\right)^{-1}. \tag{C.19}$$

To complete the proof, we need the following to hold: $\|C_{\epsilon_{\text{MV}}}^{-1}\|_\infty \leq \|C_{\epsilon_\lambda}^{-1}\|_\infty$, but due to the monotonicity of $\|C_{\epsilon}^{-1}\|_\infty$, it is sufficient to show that $\epsilon_{\text{MV}} \leq \epsilon_\lambda$.

Recall that $\epsilon_{\text{MV}} \leq \exp\left(-2m\left(\frac{1}{2}-\epsilon_\lambda\right)^2\right)$, so if we set $\exp\left(-2m\left(\frac{1}{2}-\epsilon_\lambda\right)^2\right) \leq \epsilon_\lambda$, we obtain the minimum number of label functions, m, required to ensure $\epsilon_{\text{MV}} \leq \epsilon_\lambda$.

$$\begin{aligned}\exp\left(-2m\left(\frac{1}{2}-\epsilon_\lambda\right)^2\right) &\leq \epsilon_\lambda \\ &\Rightarrow m \geq \frac{\log(1/\epsilon_\lambda)}{2\left(\frac{1}{2}-\epsilon_\lambda\right)^2}.\end{aligned}$$

Plugging (C.19) into Theorem 1, we obtain the following

$$\begin{aligned}d_{\text{TV}}(\widetilde{P}_{\text{MV}}, \widetilde{Q}_{\text{MV}}) \leq d_{\text{TV}}(P,Q) &\leq \|C_{\epsilon_{\text{MV}}}^{-1}\|_\infty \, d_{\text{TV}}(\widetilde{P}_{\text{MV}}, \widetilde{Q}_{\text{MV}}) \\ &\leq \left(1 - 2\exp\left(-2m\left(\frac{1}{2}-\epsilon_\lambda\right)^2\right)\right)^{-1} d_{\text{TV}}(\widetilde{P}_{\text{MV}}, \widetilde{Q}_{\text{MV}}) \\ &\leq (1-2\epsilon_\lambda)^{-1}\, d_{\text{TV}}(\widetilde{P}_{\text{MV}}, \widetilde{Q}_{\text{MV}}) \\ &= \|C_{\epsilon_\lambda}^{-1}\|_\infty \, d_{\text{TV}}(\widetilde{P}_{\text{MV}}, \widetilde{Q}_{\text{MV}})\end{aligned}$$

which completes the proof. □

Notice that the proof of Proposition 1 does not depend on total variation distance beyond the dependence on Theorem 1. As such, Proposition 1 can be stated more generally in terms of the Integral Probability Metric induced by the GAN discriminator $\mathcal{F}$ using Theorem 2 of [249]:

$$d_{\mathcal{F}}(\widetilde{P}_{\text{MV}}, \widetilde{Q}_{\text{MV}}) \leq d_{\mathcal{F}}(P,Q) \leq \|C_{\epsilon_{\text{MV}}}^{-1}\|_\infty \, d_{\mathcal{F}}(\widetilde{P}_{\text{MV}}, \widetilde{Q}_{\text{MV}}) \leq \|C_{\epsilon_\lambda}^{-1}\|_\infty \, d_{\mathcal{F}}(\widetilde{P}_{\text{MV}}, \widetilde{Q}_{\text{MV}}).$$

Finally, notice that Proposition 1 is made in terms of $d_{\mathcal{F}}(\widetilde{P}_{\text{MV}}, \widetilde{Q}_{\text{MV}})$ and not in terms of $d_{\mathcal{F}}(\widetilde{P}_\lambda, \widetilde{Q}_\lambda)$. We can show that as the number of LFs approach infinity, we recover the distance under the clean labels: $d_{\mathcal{F}}(P,Q)$. Applying Theorem 1 to majority vote and a single LF results in the following two expressions:

$$d_{\mathcal{F}}(\widetilde{P}_{\text{MV}}, \widetilde{Q}_{\text{MV}}) \leq d_{\mathcal{F}}(P,Q) \leq \|C_{\epsilon_{\text{MV}}}^{-1}\|_\infty \, d_{\mathcal{F}}(\widetilde{P}_{\text{MV}}, \widetilde{Q}_{\text{MV}}) \tag{C.20}$$

$$d_{\mathcal{F}}(\widetilde{P}_\lambda, \widetilde{Q}_\lambda) \leq d_{\mathcal{F}}(P,Q) \leq \|C_{\epsilon_\lambda}^{-1}\|_\infty \, d_{\mathcal{F}}(\widetilde{P}_\lambda, \widetilde{Q}_\lambda) \tag{C.21}$$

Rearranging terms, we obtain the following

$$1 \leq \frac{d_{\mathcal{F}}(P, Q)}{d_{\mathcal{F}}(\widetilde{P}_{\mathrm{MV}}, \widetilde{Q}_{\mathrm{MV}})} \leq \|C_{\epsilon_{\mathrm{MV}}}^{-1}\|_\infty \leq \|C_{\epsilon_\lambda}^{-1}\|_\infty$$

and

$$1 \leq \frac{d_{\mathcal{F}}(P, Q)}{d_{\mathcal{F}}(\widetilde{P}_{\lambda}, \widetilde{Q}_{\lambda})} \leq \|C_{\epsilon_\lambda}^{-1}\|_\infty.$$

Notice that $\|C_{\epsilon_\lambda}^{-1}\|_\infty$ has no dependence on m since it's a single LF, but $\|C_{\epsilon_{\mathrm{MV}}}^{-1}\|_\infty$ approaches 1 as $m \to \infty$:

$$\begin{aligned}
&\lim_{m\to\infty} 1 \leq \frac{d_{\mathcal{F}}(P, Q)}{d_{\mathcal{F}}(\widetilde{P}_{\mathrm{MV}}, \widetilde{Q}_{\mathrm{MV}})} \leq \|C_{\epsilon_{\mathrm{MV}}}^{-1}\|_\infty \leq \|C_{\epsilon_\lambda}^{-1}\|_\infty \\
&\Rightarrow 1 \leq \frac{d_{\mathcal{F}}(P, Q)}{d_{\mathcal{F}}(\widetilde{P}_{\mathrm{MV}}, \widetilde{Q}_{\mathrm{MV}})} \leq 1 \leq \|C_{\epsilon_\lambda}^{-1}\|_\infty \\
&\Rightarrow d_{\mathcal{F}}(P, Q) = d_{\mathcal{F}}(\widetilde{P}_{\mathrm{MV}}, \widetilde{Q}_{\mathrm{MV}}).
\end{aligned}$$

Hence we obtain a stronger bound as the number of LFs increases.

C.6.3 Extensions

For the sake of clarity, several simplifying assumptions are made in Claims (1) and (2). Both claims use the simplest possible aggregation strategy for weak supervision—majority vote, and our analysis in Claim (1) involves the use of a Gaussian mixture model—a less complex object of study compared to a GAN. Both analyses can directly be extend to use more sophisticated weak supervision label models instead of majority vote, and different generative models, which should lead to improved bounds at the expense of a more complex claim statement. Note, additionally, that neither of the claims attempt to provide deep insight into the benefits of *jointly* learning the generative model and the label model—but this may be done with a slightly more careful analysis.

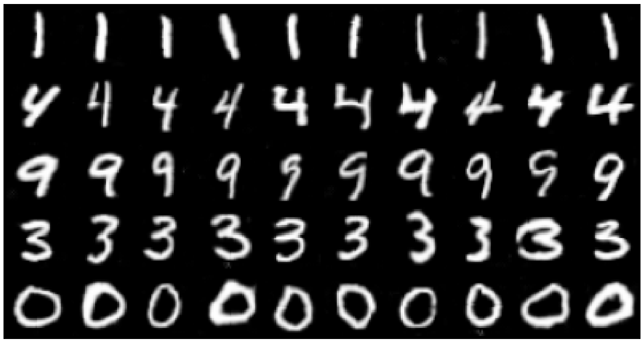


Figure C.5: A random set of MNIST images generated by WSGAN-E (using a DCGAN), with the discrete latent random variable kept fix for each row of images.

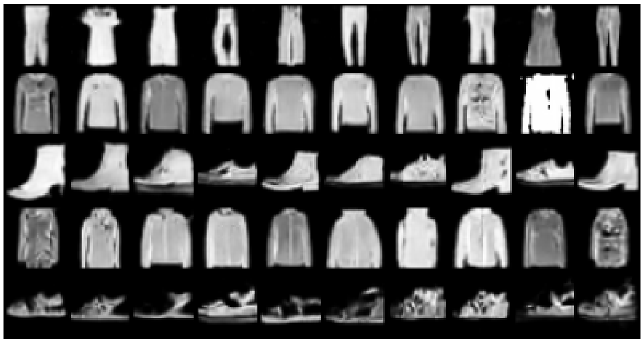


Figure C.6: A random set of FashionMNIST images generated by WSGAN-E (using a DCGAN), with the discrete latent random variable kept fix for each row of images.

C.7 Additional Images

This section provides additional generated images in Figures C.5, C.6, C.7, and C.8. These random images are generated by WSGAN with a DCGAN base architecture, where the discrete latent variable d passed to the generator is kept the same in each row.

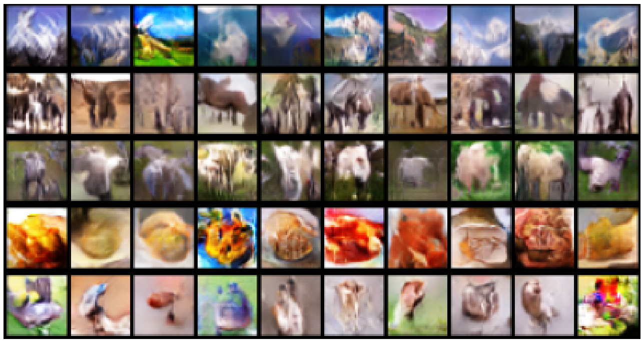


Figure C.7: A random set of Domainnet images generated by WSGAN-E (using a DCGAN), with the discrete latent random variable kept fix for each row of images. Note that this dataset is particularly challenging for a GAN as the subset used that is used has few images, resulting in considerably lower quality of synthetic images compared to GTSRB for example.

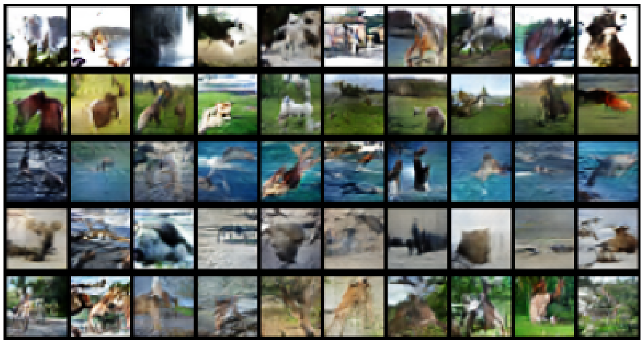


Figure C.8: A random set of AwA2 images generated by WSGAN-E (using a DC-GAN), with the discrete latent random variable kept fix for each row of images. Note that this dataset is particularly challenging for a GAN, as the weakly supervised subset that is used has very few images, resulting in considerably lower quality of synthetic images compared to GTSRB for example.

Appendix D

Interactive Weak Supervision

D.1 LF Accuracy and Coverage Trade-off

This section analyzes how LF accuracy and LF propensity (i.e. non-abstain behavior) influence the estimate of the true latent class label Y^*. For simplicity, the analysis will focus on the binary classification case. Assume each data point $x \in \mathcal{X}$ has a latent class label $y^* \in \mathcal{Y} = \{-1, 1\}$. Given n unlabeled, i.i.d. data points $X = \{x_i\}_{i=1}^n$, the goal is to train a classifier $f : \mathcal{X} \to \mathcal{Y}$ such that $f(x) = y^*$. As in [222] a user provides m LFs $\{\lambda_j\}_{j=1}^m$, where $\lambda_j : \mathcal{X} \to \mathcal{Y} \cup \{0\}$ noisily label the data with $\lambda_j(x) \in \{-1, 1\}$ or abstain with $\lambda_j(x) = 0$. The corresponding LF output matrix is $\Lambda \in \{-1, 0, 1\}^{n \times m}$, where $\Lambda_{i,j} = \lambda_j(x_i)$.

A factor graph is defined as proposed in [222, 220] to obtain probabilistic labels by modeling the LF accuracies via factor $\phi_{i,j}^{Acc}(\Lambda, Y) \triangleq \mathbb{1}\{\Lambda_{ij} = y_i\}$ and labeling propensity by factor $\phi_{i,j}^{Lab}(\Lambda, Y) \triangleq \mathbb{1}\{\Lambda_{ij} \neq 0\}$. Assume LFs are independent conditional on Y. The label model is defined as

$$p_\theta(Y, \Lambda) \triangleq Z_\theta^{-1} \exp\left(\sum_{i=1}^n \theta^\top \phi_i(\Lambda_i, y_i)\right), \tag{D.1}$$

where Z_θ is a normalizing constant and $\phi_i(\Lambda_i, y_i)$ defined to be the concatenation of the factors for all LFs $j = 1, \ldots, m$ for sample i. Also, let $\theta = (\theta^{(1)}, \theta^{(2)})$ where $\theta^{(1)}, \theta^{(2)} \in \mathbb{R}^m$. Here, $\theta^{(1)}$ are the canonical parameters for the LF accuracies, and $\theta^{(2)}$ the canonical parameters for the LF propensities.

To estimate the label model parameters, one generally obtains the maximum marginal likelihood estimate via the (scaled) log likelihood

$$l(\theta) = 1/n \sum_{i=1}^n \log\left(\sum_{y \in \mathcal{Y}} p(\Lambda_i, y | \theta)\right).$$

Let finite $\hat{\theta} \in \mathbb{R}^{2m}$ be such an estimate. One uses $p_{\hat{\theta}}(y|\Lambda_i)$ to obtain probabilistic labels:

$$p_{\hat{\theta}}(y_i = k|\Lambda_i) = \frac{p_{\hat{\theta}}(y_i = k, \Lambda_i)}{p_{\hat{\theta}}(\Lambda_i)} \tag{D.2}$$

$$= \frac{\exp(\sum_{j=1}^{m} \hat{\theta}_j^{(1)} \phi^{Acc}(\lambda_j(x_i), k))}{\sum_{\tilde{y}\in\mathcal{Y}} \exp(\sum_{j=1}^{m} \hat{\theta}_j^{(1)} \phi^{Acc}(\lambda_j(x_i), \tilde{y}))}. \tag{D.3}$$

Note that the label estimate does not directly depend on $\theta^{(2)}$. Further, note that the denominator is the same over different label possibilities. Finally, note that even in a case where we include correlation factors $\phi_{i,j,k}^{Corr}(\Lambda, Y) = \mathbb{1}\{\Lambda_{ij} = \Lambda_{ik}\}, (j,k) \in C$ in the model above with C as a set of potential dependencies, the probabilistic label will only directly depend on the estimated canonical accuracy parameters $\theta^{(1)}$. In the binary classification case, which is assumed here, the expression simplifies further. For $k \in \{-1, 1\}$:

$$p_{\hat{\theta}}(y_i = k|\Lambda_i) = \frac{\exp(\sum_{j=1}^{m} \hat{\theta}_j^{(1)} \phi^{Acc}(\lambda_j(x_i), k))}{\sum_{\tilde{y}\in\{-1,1\}} \exp(\sum_{j=1}^{m} \hat{\theta}_j^{(1)} \phi^{Acc}(\lambda_j(x_i), \tilde{y}))} \tag{D.4}$$

$$= \frac{1}{1 + \exp(\sum_{j=1}^{m} \hat{\theta}_j^{(1)} (\phi^{Acc}(\lambda_j(x_i), -k) - \phi^{Acc}(\lambda_j(x_i), k)))} \tag{D.5}$$

$$= \sigma(\sum_{j=1}^{m} \hat{\theta}_j^{(1)} (\phi^{Acc}(\lambda_j(x_i), k) - \phi^{Acc}(\lambda_j(x_i), -k))), \tag{D.6}$$

where σ denotes the sigmoid function. The probabilistic labels are a softmax in the multi-class classification case and, as shown above, simplify to a sigmoid in the binary case. An absolute label prediction $\hat{y} \in \{-1, 1\}$ is therefore simply a function of

$$\hat{y}_i = \underset{\tilde{y}\in\mathcal{Y}}{\operatorname{argmax}}\, p_{\hat{\theta}}(y_i = \tilde{y}|\Lambda_i) = \underset{\tilde{y}\in\mathcal{Y}}{\operatorname{argmax}} \sum_{j=1}^{m} \hat{\theta}_j^{(1)} \phi^{Acc}(\lambda_j(x_i), \tilde{y}).$$

Some assumptions on the accuracy and error probabilities of labeling functions will now be introduced, similar to the Homogenous Dawid-Skene model [72, 161] in crowd sourcing, where label source accuracy is the same across classes and errors are evenly divided with probability mass *independent* of the true class.

Under these assumptions, denote by $\alpha_j = P(\lambda_j(x) = y^*|\lambda_j(x) \neq 0)$ the accuracy of LF j. Further, denote by $l_j = P(\lambda_j(x) \neq 0)$ the labeling propensity of j, i.e. how frequently LF j does not abstain. Note that the observed LF propensity is also referred to as LF coverage in the related literature. Recall Theorem 4.1.1:

Theorem 4.1.1. *Assume a binary classification setting, m independent labeling functions with accuracy $\alpha_j \in [0, 1]$ and labeling propensity $l_j \in [0, 1]$. For a label model as* *in Eq. (4.1) with given label model parameters* $\hat{\theta} \in \mathbb{R}^{2m}$*, and for any* $i \in \{1, \dots, n\}$*,*

$$P(\hat{y}_i = y_i^*) \geq 1 - \exp\left(- \frac{(\sum_{j=1}^m \hat{\theta}_j^{(1)} (2\alpha_j - 1) l_j)^2}{2||\hat{\theta}^{(1)}||^2} \right)$$

where $\hat{\theta}^{(1)}$ *are the* m *weights of* ϕ^{Acc}*, and* $\hat{y}_i \in \{-1, 1\}$ *is the label model estimate for* y_i^**.*

Proof. Assume that we use the label model to obtain a label estimate $\hat{y}_i \in \{-1, 1\}$. As shown in Eq. (D.3), the prediction rule in that case is

$$\hat{y}_i = \underset{\tilde{y} \in \{-1,1\}}{\operatorname{argmax}} \sum_{j=1}^m \hat{\theta}_j^{(1)} \phi^{Acc}(\lambda_j(x_i), \tilde{y}).$$

Define by $\lambda(x) = (\lambda_1(x), \dots, \lambda_m(x))$ the vector of the $j = 1, \dots, m$ LF outputs on x. Further, define for $k \in \{-1, 1\}$:

$$\begin{aligned} V_{\hat{\theta}}(\lambda(x), k) &= \sum_{j=1}^m \hat{\theta}_j^{(1)} (\phi^{Acc}(\lambda_j(x), k) - \phi^{Acc}(\lambda_j(x), -k)) \\ &= \sum_{j=1}^m \hat{\theta}_j^{(1)} \left(\mathbb{1}\{\lambda_j(x) = k\} - \mathbb{1}\{\lambda_j(x) = -k\} \right). \end{aligned}$$

For the two label options $k \in \{-1, 1\}$, we have

$$V_{\hat{\theta}}(\lambda(x), 1) = \sum_{j=1}^m \hat{\theta}_j^{(1)} \left(\mathbb{1}\{\lambda_j(x) = 1\} - \mathbb{1}\{\lambda_j(x) = -1\} \right) = \sum_{j=1}^m \hat{\theta}_j^{(1)} \lambda_j(x)$$

and

$$V_{\hat{\theta}}(\lambda(x), -1) = \sum_{j=1}^m \hat{\theta}_j^{(1)} \left(\mathbb{1}\{\lambda_j(x) = -1\} - \mathbb{1}\{\lambda_j(x) = 1\} \right) = -\sum_{j=1}^m \hat{\theta}_j^{(1)} \lambda_j(x).$$

Now, we want to obtain a bound on the probability that the label estimate $\hat{y}_i$ is equal to the true label. We have

$$\begin{aligned} P(\hat{y}_i = y_i^*) &= P(y_i^* = 1) P(\hat{y}_i = 1 | y_i^* = 1) + P(y_i^* = -1) P(\hat{y}_i = -1 | y_i^* = -1) \\ &= P(y_i^* = 1) P(\hat{y}_i = 1 | y_i^* = 1) + (1 - P(y_i^* = 1)) P(\hat{y}_i = -1 | y_i^* = -1). \end{aligned}$$

Note that

$$P(\hat{y}_i = 1|y_i^* = 1) = P(V_{\hat{\theta}}(\lambda(x_i), 1) > 0|y_i^* = 1) = P(\sum_{j=1}^{m} \hat{\theta}_j^{(1)} \lambda_j(x) > 0|y_i^* = 1),$$

and that

$$P(\hat{y}_i = -1|y_i^* = -1) = P(V_{\hat{\theta}}(\lambda(x_i), -1) > 0|y_i^* = -1) = P(\sum_{j=1}^{m} \hat{\theta}_j^{(1)} \lambda_j(x_i) < 0|y_i^* = -1).$$

We therefore have

$$P(\hat{y}_i = y_i^*) = P(y_i^* = 1)P(\sum_{j=1}^{m} \hat{\theta}_j^{(1)} \lambda_j(x_i) > 0|y_i^* = 1) + (1 - P(y_i^* = 1))P(\sum_{j=1}^{m} \hat{\theta}_j^{(1)} \lambda_j(x_i) < 0|y_i^* = -1$$

Now we define $\xi_{ij} = \hat{\theta}_j^{(1)} \lambda_j(x_i)$ and we know that $\xi_j \in [-|\hat{\theta}_j^{(1)}|, |\hat{\theta}_j^{(1)}|]$. Given the Dawid-Skene model assumptions stated previously, we have

$$\mathbb{E}[\sum_{j=1}^{m} \xi_{ij}|y_i^* = 1] = \sum_{j=1}^{m} \mathbb{E}[\xi_{ij}|y_i^* = 1] = \sum_{j=1}^{m} \hat{\theta}_j^{(1)} l_j(2 * \alpha_j - 1),$$

and

$$\mathbb{E}[\sum_{j=1}^{m} \xi_{ij}|y_i^* = -1] = \sum_{j=1}^{m} \mathbb{E}[\xi_{ij}|y_i^* = -1] = -\sum_{j=1}^{m} \hat{\theta}_j^{(1)} l_j(2 * \alpha_j - 1).$$

Now, using Hoeffding's inequality and assuming independent LFs, we can bound $P(\hat{y}_i = 1|y_i^* = 1)$ and $P(\hat{y}_i = -1|y_i^* = -1)$ from below:

$$\begin{aligned}
P(\sum_{j=1}^{m} \hat{\theta}_j^{(1)} \lambda_j(x_i) > 0|y_i^* = 1) &= P(\sum_{j=1}^{m} \xi_{ij} > 0|y_i^* = 1) \\
&= P(\sum_{j=1}^{m} \xi_{ij} - \mathbb{E}[\sum_{j=1}^{m} \xi_{ij}|y_i^* = 1] > -\sum_{j=1}^{m} \hat{\theta}_j^{(1)} l_j(2 * \alpha_j - 1) \ |y_i^* = 1) \\
&\geq 1 - \exp\left(-\frac{(\sum_{j=1}^{m} \hat{\theta}_j^{(1)}(2\alpha_j - 1)l_j)^2}{2||\hat{\theta}^{(1)}||^2}\right),
\end{aligned}$$

and

$$\begin{aligned}
P(\sum_{j=1}^{m} \hat{\theta}_j^{(1)} \lambda_j(x_i) < 0|y_i^* = -1) &= P(\sum_{j=1}^{m} \xi_{ij} < 0|y_i^* = -1) \\
&= P(\sum_{j=1}^{m} \xi_{ij} - \mathbb{E}[\sum_{j=1}^{m} \xi_{ij}|y_i^* = -1] < \sum_{j=1}^{m} \hat{\theta}_j^{(1)} l_j(2 * \alpha_j - 1) \ |y_i^* = -1) \\
&\geq 1 - \exp\left(-\frac{(\sum_{j=1}^{m} \hat{\theta}_j^{(1)}(2\alpha_j - 1)l_j)^2}{2||\hat{\theta}^{(1)}||^2}\right).
\end{aligned}$$

Finally we have

$$
\begin{aligned}
P(\hat{y}_i = y_i^*) &= P(y_i^* = 1)P(\hat{y}_i = 1|y_i^* = 1) + (1 - P(y_i^* = 1))P(\hat{y}_i = -1|y_i^* = -1) \\
&\geq 1 - \exp\left(-\frac{(\sum_{j=1}^{m} \hat{\theta}_j^{(1)}(2\alpha_j - 1)l_j)^2}{2||\hat{\theta}^{(1)}||^2}\right).
\end{aligned}
$$

□

What do the theorem and the quantities analyzed in this section indicate?

- The trade-off between LF accuracy and LF propensity (also referred to as LF coverage) is captured by $(2\alpha_j - 1)l_j$ which allows us to rank LFs if we know the accuracy α_j or can estimate it and use the observed, empirical coverage as an estimate of l_j.

- Not surprising, the relation between sign(θ_j) and α_j is important. A better than random LF j should have a positive θ_j. This indicates that a gap to randomness is important if we cannot guarantee that we learn θ_j well, to reduce the chance of obtaining a negative θ_j for better than random LF j, or vice versa.

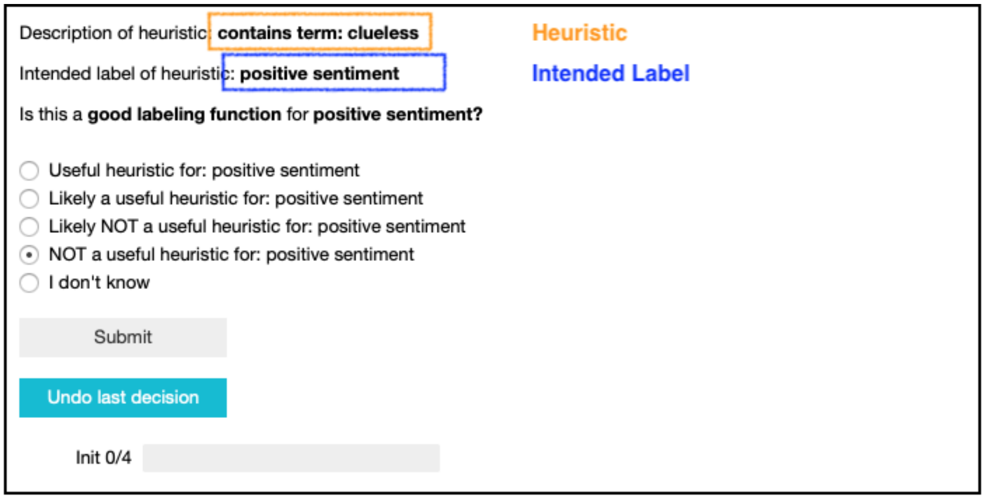
- Note how the label estimates are obtained in Eq. (D.3). Increasing the θ_j of am LF also effectively means reducing the impact other LFs have on a prediction. In particular when θ estimates are imperfect, a gap to random accuracy of α_j is important to obtain good label estimates. Intuitively, we do not want to add excessive noise by including LFs close to random unless we can guarantee that their parameter estimate is appropriately low and has the correct sign.

D.2 Interactive Weak Supervision User Experiments

D.3 Interface and experiment prompt

Fig. D.1 shows an example of the prompt that was shown to users at each iteration of the IWS user experiments. Before the experiment started, users were first instructed on the interface they would see and the task they would be given, i.e. to label a heuristic as good if they would expect it to label samples at better than random accuracy and as bad otherwise. Users were also instructed about the response options, including the option to not answer a query if they were unsure (‘I don’t know’).

Users were given a description of the classification task and domain of the documents for which heuristics were being acquired. Users were also provided with a description of the heuristic generated which labeled samples with a target label if a document contained a certain term. Finally, users were given two examples of a better than random heuristic, and two examples of an arbitrary heuristic.



Description of heuristic: **contains term: clueless**

Intended label of heuristic: **positive sentiment**

Is this a **good labeling function** for **positive sentiment?**

○ Useful heuristic for: positive sentiment
○ Likely a useful heuristic for: positive sentiment
○ Likely NOT a useful heuristic for: positive sentiment
● NOT a useful heuristic for: positive sentiment
○ I don't know

Figure D.1: An example of the prompt and answer options that users were shown during the user study. Before starting the experiment, users were provided with a description of the task and the labeling function family.

During the experiments, users were also provided with 4 random examples of documents documents where the queried LF applied. Users were instructed to first consider the LF without inspecting these random samples, and to only consider the examples if necessary.

While LFs receive binary labels, users were allowed to express uncertainty about their decision, which was used as a sample weight (1 if certain else 0.5) of LFs during training of the probabilistic model of user feedback.

D.4 Additional statistics of user experiments

Fig. D.2 provides more details about the IWS user experiments. The top row displays the test set performance of downstream model f for each individual user. The middle row shows how the number of LFs determined by the user to be useful $u = 1$ increases with the number of iterations. The bottom row displays the maximum positive correlation between a new LF with $u = 1$ at iteration t and all previously accepted LFs with $u = 1$ up to iteration t. Note that abstains are taken into account by computing the correlation between and LF i and j only on entries where at least one of them is nonzero.

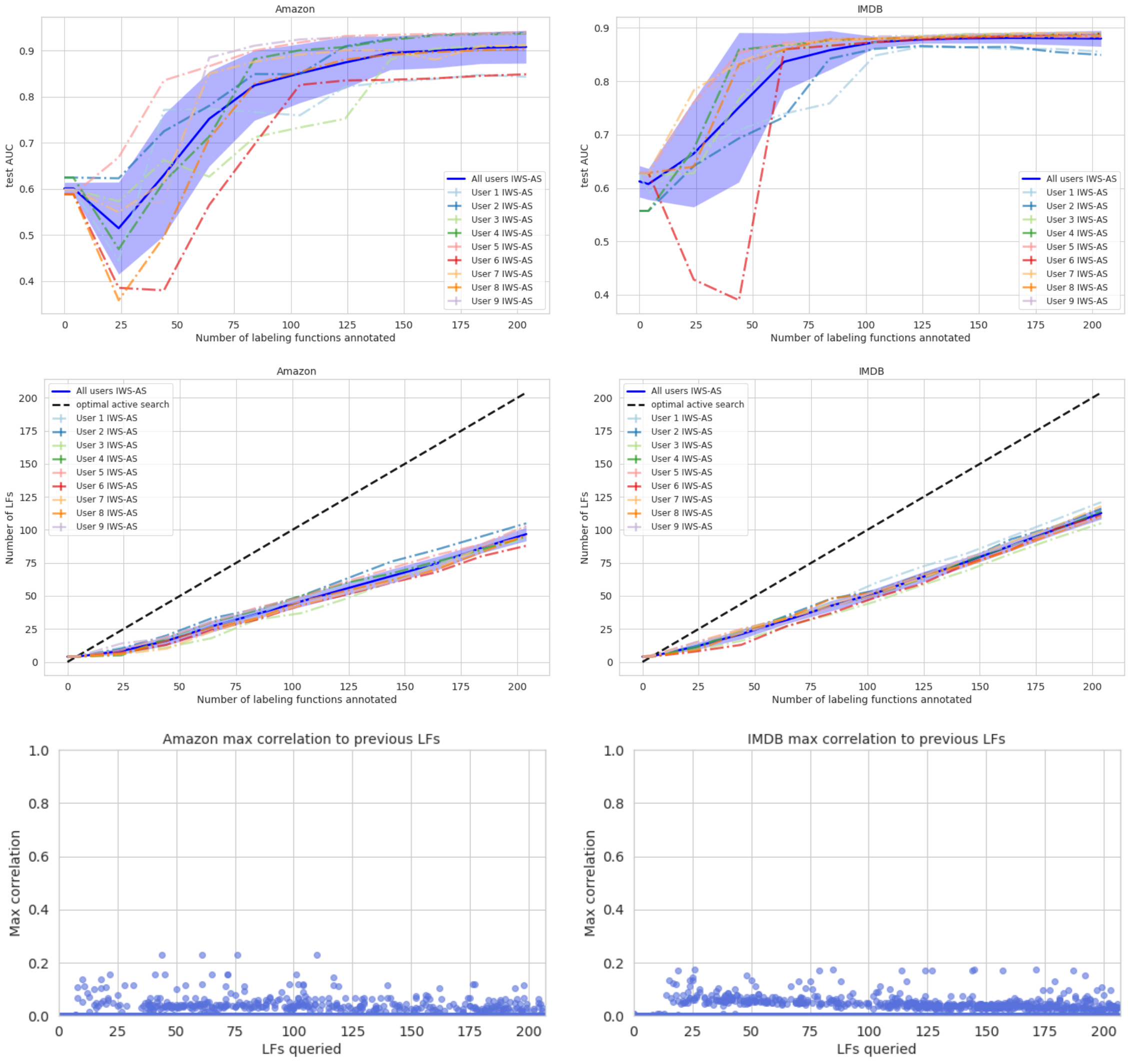


Figure D.2: Test AUC by IWS iteration shown for individual user experiments with IWS-AS (*top*). Number of LFs labeled as useful by IWS iterations (*middle*). Maximum correlation to previously accepted LFs by number of iterations (*bottom*).

Appendix E

Weak Supervision as Paired Multi-Modal data: Vision–Language Processing in Biomedicine

E.1 Additional Experiments

E.1.1 Zero-shot Text-prompt Sensitivity Analysis

Vision-language pretraining aligns image and text data in a joint representation space, which enables impressive zero-shot downstream image classification performance via input text prompts. However, some recent work [128, 292] has shown that downstream task performance can heavily depend on the choice of text prompts. Constructing good text prompts (prompt engineering) may require expert domain knowledge and can be costly and time-consuming. Table E.1 studies RSNA pneumonia zero-shot classification performance using different text prompt combinations. Compared to the baseline, BioViL demonstrates much lower sensitivity to prompt choices selected from the data distribution. BioViL maintains its high performance even when faced with relatively long queries, which is not the case for the baseline model. These observations suggest that the improved text encoder CXR-BERT is more robust to prompt variations, and makes prompt engineering easier and less of a requirement to achieve high zero-shot classification performance.

E.1.2 Qualitative Results – Phrase Grounding

Fig. E.1 describes some phrase grounding examples obtained with different models on the `MS-CXR` dataset. From left to right, the figure shows the ClinicalBERT baseline, ConVIRT, GLoRIA, and BioViL similarity maps. While the figure only illustrates

Table E.1: Text prompt sensitivity analysis on the RSNA pneumonia zero-shot classification task. Image-text models trained without the proposed text modelling improvements (Table 4.5) show higher sensitivity to different input text prompts as the latent text embeddings are inconsistent for synonym phrases. For this reason, baseline methods often require post-hoc text prompt engineering heuristics (e.g. [128]).

Method	Pos. Query	Neg. Query	F1 Score	ROC-AUC	$\lvert\Delta AUC\rvert$
BioViL	"Findings suggesting pneumonia"	"There is no evidence of acute pneumonia"	0.657	0.822	-
ClinicalBert	"Findings suggesting pneumonia"	"There is no evidence of acute pneumonia"	0.581	0.731	-
BioViL	"Findings suggesting pneumonia"	"No evidence of pneumonia"	0.665	0.831	-
BioViL	"Consistent with the diagnosis of pneumonia"	"There is no evidence of acute pneumonia"	0.669	0.839	0.008
ClinicalBert	"Findings suggesting pneumonia"	"No evidence of pneumonia"	0.614	0.815	-
ClinicalBert	"Consistent with the diagnosis of pneumonia"	"There is no evidence of acute pneumonia"	0.621	0.694	0.121
BioViL	"Findings consistent with pneumonia"	"No evidence of pneumonia"	0.672	0.838	-
BioViL	"Findings consistent with pneumonia"	"There is no pneumonia"	0.679	0.847	0.009
ClinicalBert	"Findings consistent with pneumonia"	"No evidence of pneumonia"	0.640	0.782	-
ClinicalBert	"Findings consistent with pneumonia"	"There is no pneumonia"	0.586	0.724	0.058

Table E.2: An extension of Table 4.7 to include Sensitivity and Specificity for the RSNA Pneumonia zero-shot and fine-tuned classification. Results are compared to GLoRIA scores reported in [128], which outperforms ConVIRT [292] (see [128]). Training size: GLoRIA ($N = 186k$, private dataset), BioViL ($N = 146.7k$ of MIMIC-CXR).

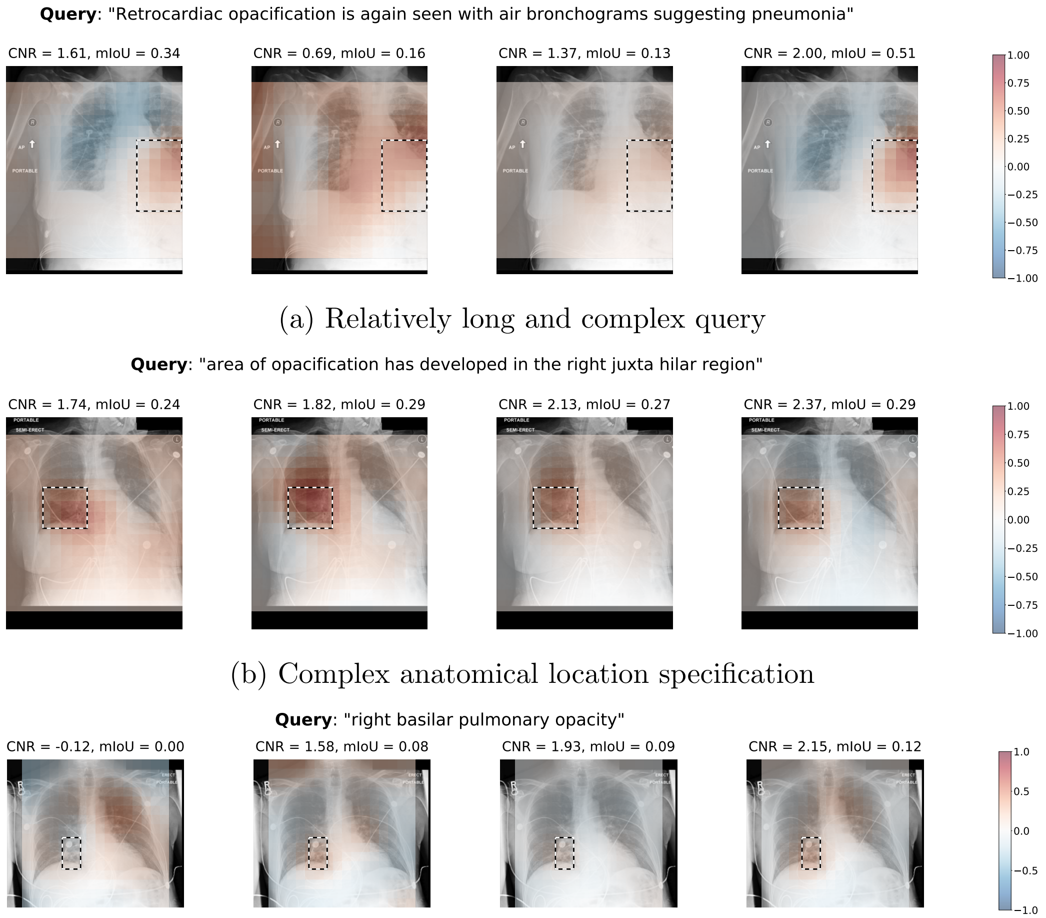
Method	Type	Text Model	Loss	% of labels	Acc.	Sens.	Spec.	F1	AUROC
SimCLR [45]	Image only	-	Global	1%	0.545	0.776	0.436	0.522	0.701
				10%	0.760	0.663	0.806	0.639	0.802
				100%	0.788	0.685	0.837	0.675	0.849
GLoRIA [128]	Joint	ClinicalBERT	Global & local	Zero-shot	0.70	0.89	0.65	0.58	-
				1%	0.72	0.82	0.69	0.63	0.861
				10%	0.78	0.78	0.79	0.63	0.880
				100%	0.79	0.87	0.76	0.65	0.886
Baseline	Joint	ClinicalBERT	Global	Zero-shot	0.719	0.648	0.781	0.614	0.812
BioViL	Joint	CXR-BERT	Global	Zero-shot	0.732	0.831	0.685	0.665	0.831
				1%	0.805	0.791	0.812	0.723	0.881
				10%	0.812	0.781	0.826	0.727	0.884
				100%	0.822	0.755	0.856	0.733	0.891

a few examples, the results demonstrate that phrase grounding performance can be significantly enhanced by leveraging improved text modelling (BioViL). The examples include clinical findings that differ in size, type, and anatomical location.

Additionally, Fig. E.3 describes some failure cases of BioViL on the `MS-CXR` dataset to motivate any further research on this topic. In particular, the models show limitations in grounding the descriptions relating to smaller structures (e.g., rib fracture, pneumothorax), and in a few cases the location modifier is not disassociated from the entities corresponding to abnormalities, see (a) in Fig. E.3.

E.1.3 Additional Evaluation Metrics

In Table E.2, an extension of Table 4.7 is provided to include the sensitivity and specificity metrics for the zero-shot and fine-tuned classification experiments presented in Section 4.2.3. The classification thresholds are set to maximize the F1 scores for each method. Further, Table E.4 provides mean IoU scores for the phrase grounding experiments presented in Section 4.2.3, which evaluates the pretrained BioViL model on the `MS-CXR` dataset. It is observed that the distribution of similarity scores is different for GLoRIA and BioViL-L due to the different temperature parameter used in the local loss term in [128]. To provide a fair comparison, the similarity scores are adjusted via min-max scaling to the full $[-1, 1]$ range. The same scaling strategy is utilized in the implementation of the baseline method [128]. Note that the CNR scores are not affected by this linear re-scaling.

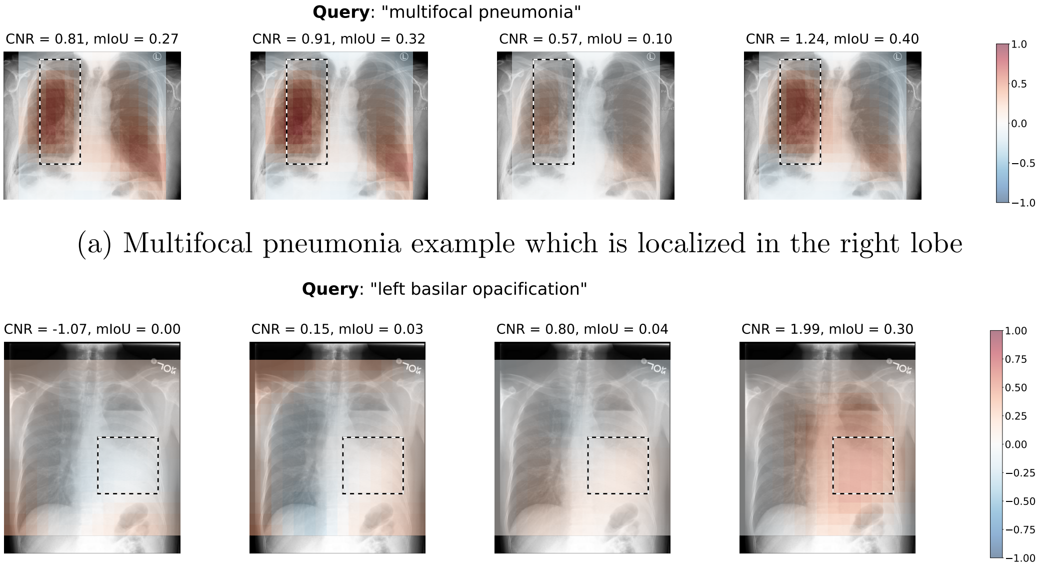


(a) Relatively long and complex query

(b) Complex anatomical location specification

(c) Small ground-truth bounding box

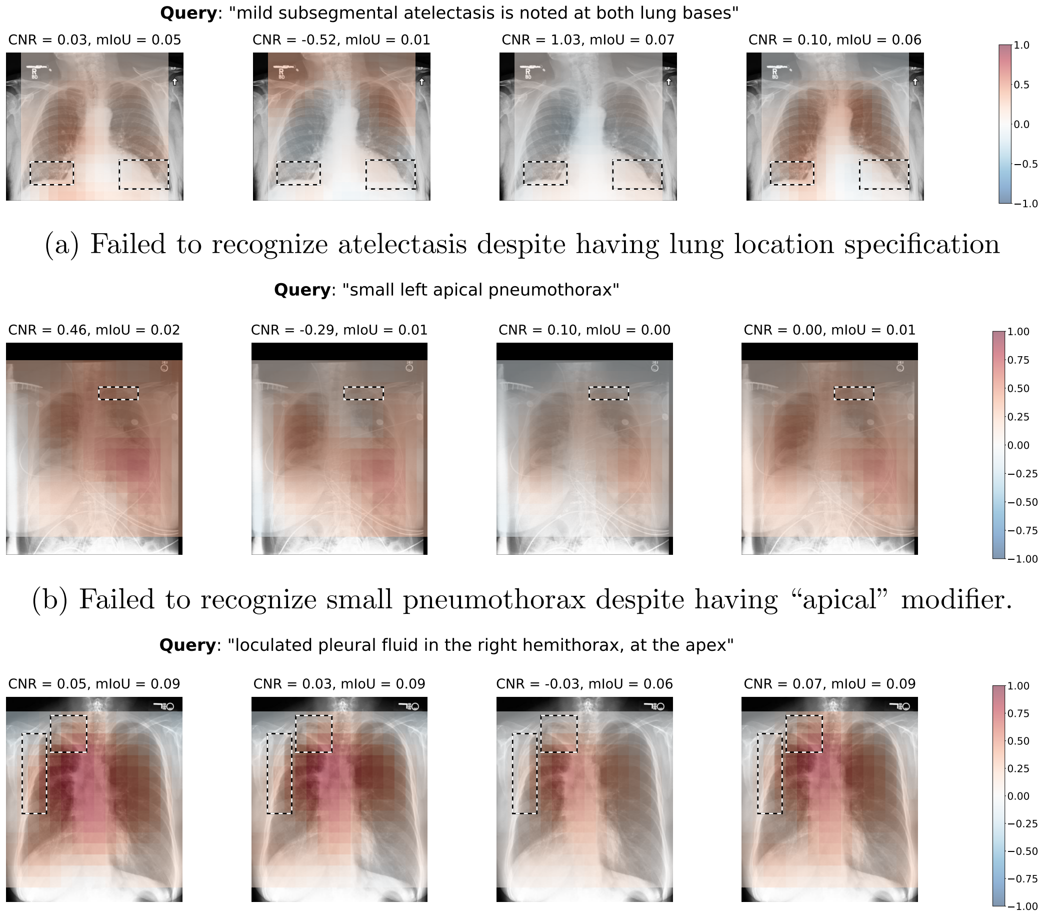
Figure E.1: Qualitative examples from MS-CXR phrase grounding benchmark. Model outputs (latent vector similarity) are compared (from left, ClinicalBERT baseline, ConVIRT, GLoRIA, and BioViL). See Fig. E.2 for more examples.



(a) Multifocal pneumonia example which is localized in the right lobe

(b) Location modifier “left basilar”

Figure E.2: More qualitative examples from MS-CXR phrase grounding benchmark. Model outputs (latent vector similarity) are compared (from left, ClinicalBERT baseline, ConVIRT, GLoRIA, and BioViL)

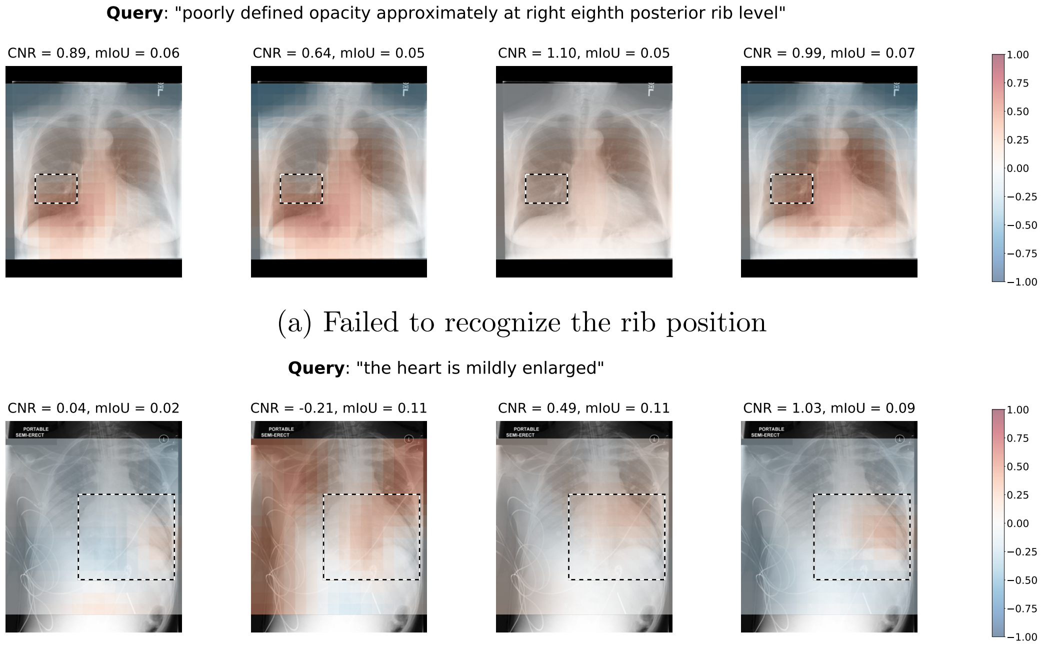


(a) Failed to recognize atelectasis despite having lung location specification

(b) Failed to recognize small pneumothorax despite having “apical” modifier.

(c) Loculated pleural fluid not recognized despite “apical” and “right hemithorax” information.

Figure E.3: `MS-CXR` benchmark failure cases. Latent vector similarity is compared (from left, ClinicalBERT baseline, ConVIRT, GLoRIA, and BioViL). See Fig. E.4 for more failure cases.



(a) Failed to recognize the rib position

(b) Mismatch between bounding box and salient region: Models attend to the enlarged area to identify the abnormality instead of the entire heart.

Figure E.4: `MS-CXR` benchmark failure cases. Latent vector similarity is compared (from left, ClinicalBERT baseline, ConVIRT, GLoRIA, and BioViL).

Table E.3: Ablations on BioViL – Increasing training set size and use of raw DICOM images instead of compressed JPEG images. The approaches are compared in terms of contrast-to-noise ratio (CNR) obtained on the newly released `MS-CXR` dataset, averaged over four runs with different seeds.

Method	Training	Atelectasis	Cardiomegaly	Consolidation	Lung opacity	Edema	Pneumonia	Pneumothorax	Pl. effusion	Avg.
BioViL	146.7k	1.02±.06	0.63±.08	1.42±.02	1.05±.06	0.93±.03	1.27±.04	0.48±.06	1.40±.06	1.03±.02
+ More data	176.0k	1.01±.07	0.70±.03	1.45±.01	1.04±.04	0.94±.01	1.27±.05	0.54±.05	1.43±.04	1.05±.02
+ Raw images	176.0k	1.03±.06	0.64±.09	1.51±.02	1.12±.06	1.00±.07	1.39±.04	0.56±.05	1.46±.05	1.09±.02

Table E.4: Mean IoU scores obtained on the newly released `MS-CXR` dataset, averaged over four runs with different seeds. The results are collected using different text encoder and training objectives (e.g., G&L: Global and local loss).

Method	Objective	Text encoder	Atelectasis	Cardiomegaly	Consolidation	Lung opacity	Edema	Pneumonia	Pneumothorax	Pl. effusion	Avg.
Baseline	Global	ClinicalBERT	0.228	0.269	0.293	0.173	0.268	0.249	0.084	0.232	0.224
Baseline	Global	PubMedBERT	0.225	0.293	0.297	0.167	0.266	0.286	0.077	0.222	0.225
ConVIRT [292]	Global	ClinicalBERT	0.257	0.281	0.313	0.177	0.272	0.238	0.091	0.227	0.238
GLoRIA [128]	G&L	ClinicalBERT	0.261	0.273	0.324	0.198	0.251	0.246	0.100	0.254	0.246
BioViL	Global	CXR-BERT	0.296	0.292	0.338	0.202	0.281	0.323	0.109	0.290	0.266
BioViL-L	G&L	CXR-BERT	0.302	0.375	0.346	0.209	0.275	0.315	0.135	0.315	0.284

E.1.4 Ablations on Training Dataset Size & Use of Raw Input Images

An additional set of experiments are conducted to test the impact of (I) training dataset size and (II) the use of raw DICOM images instead of JPEG images on phrase grounding performance. In the former case, the number of training pairs is increased from 146.7k to 176k, where all available studies with IMPRESSION section and AP/PA scans are used after excluding the test set. In the latter ablation, the JPEG images are replaced with the raw DICOM images to reduce image artifacts due to compression. Table E.3 shows that further performance gains can be achieved by utilizing the DICOM data and matching the training set size to related methods (e.g., GLoRIA [128]), where the raw data is empirically observed to contribute more. These improved results and pretraining models are neither reported nor used in the experiments presented in the main body of this paper. The findings can provide useful insights for future research on this topic.

E.2 Background in Chest Radiology

Chest X-rays are the most commonly performed diagnostic X-ray examination, and a typical text report for such an exam consists of three sections: a "Background" section describing the reason for examination and the exam type, a "Findings" section describing abnormalities as well as normal clinical findings in the scan, and an "Impression" section which summarizes the findings and offers interpretation with possible recommendations. Multiple large Chest X-ray datasets have been released to the public (see [248] for an overview of CXR image datasets), including multi-modal ones of images and text such as MIMIC-CXR [136], some also accompanied by small sets of expert-verified ground-truth annotations of various nature, making the application a popular candidate for exploring self-supervised VLP on biomedical data.

The application area also possesses a strong clinical motivation. Globally, there is a shortage of qualified trained radiologists and a constantly increasing number of examinations in healthcare systems, workflows are hampered by issues such as a lack of standardization in report writing, and fatigue-based errors occur too frequently. Thus, decision-support systems that can analyze incoming images or image-report pairs in order to provide real-time feedback to radiologists are a promising avenue towards improving workflow efficiency and the quality of medical image readings. In practice, the existing radiology workflow can for example be augmented via machine learning models by providing feedback on any incorrect or missing information in reports, and by standardizing the reports' structure and terminology.

E.2.1 Key NLP and Dataset Challenges in Radiology

This work focuses on developing text and image models to enable clinical decision-support systems for biomedical applications via self-supervised VLP, without ground-truth annotations, and the work conducts experiments in CXR applications. Image and text understanding in the biomedical domain is distinct from general-domain applications and requires careful consideration. Medical images are elaborately structured, which is reflected in the corresponding notes. To be able to harness the dense information captured in text notes for free-text natural language supervision, it becomes imperative to obtain finely tuned text models.

Complex Sentence Structure. Linguistic characteristics in radiology reports, many shared with related clinical text settings, decidedly differ from general domain text and thus require carefully tuned text models to acquire the best possible free-text natural language supervision in self-supervised VLP. For one, negations are frequently used to indicate the absence of findings, in particular to make references as to how a patient's health has evolved, e.g. "there are no new areas of consolidation to suggest the presence of pneumonia". This sentence is for example falsely captured as positive for pneumonia by the automated CheXpert labeler [132]. Furthermore, as exemplified in this example, long-range dependencies are common, which makes understanding of relations within sentences challenging.

Use of Modifiers. Another characteristic is the use of highly specialized spatial language in radiology, which is crucial for correct diagnosis, often describing the positioning of radiographic findings or medical devices with respect to anatomical structures, see e.g. [66, 67]. The use of words like "medial", "apical", "bilateral" or "basilar" as spatial modifiers is unlikely to appear in the general domain but very common in CXR radiology reports. In addition to spatial modifiers, severity modifiers such as “mild”, “moderate” or “severe” are also commonly attached to an identified disorder or abnormality [81].

Expressions of Uncertainty. Another interesting difference to most general domain VLP applications and datasets such as Internet image captions, are expressions of uncertainty that one frequently encounters in radiology reports. One would rarely expect to find an image caption to read “We see a person petting an animal, it is likely a dog but it could also be a cat”. In contrast, consider the following real radiology example: “New abnormality in the right lower chest could be either consolidation in the lower lobe due to rapid pneumonia or collapse, and/or moderate right pleural effusion, more likely abnormality in the lung because of absent contralateral mediastinal shift.” It is an extremely long description expressing uncertainty and containing long range dependencies.

Class Imbalance. Finally, a challenge for many domain-specific VLP applications that is far less pronounced in the general domain setting is that of imbalanced latent entities. An example of such entities are the normal and anomalous findings in radiology images that doctors will describe in their report. In the CXR application, reports can roughly be divided into normal and abnormal scans, where abnormal ones reveal signs or findings observed during the exam [61]. Normal scans that do not show any signs of disease are far more common than any other findings, which leads to a larger number of false negatives in contrastive objectives compared to the general domain. An important detail is that normal scans tend to be expressed in specific forms and doctors frequently use templates to produce reports with no abnormalities.

E.3 MS-CXR Dataset Details

General Overview. This new benchmark dataset provides bounding box and sentence pair annotations describing clinical findings visible in a given chest X-ray image. `MS-CXR` consists of 1047 images, with a total of 1153 bounding box and sentence pairs. Each sentence describes a single pathology present in the image, and there could be multiple manually annotated bounding boxes corresponding to the description of the single radiological finding. Additionally, an image may have more than one pathology present, and the dataset provides separate sets of bounding boxes for each phrase describing a unique pathology associated with an image. The annotations were collected on a subset of MIMIC-CXR images, which additionally contains labels across eight different pathologies: atelectasis, cardiomegaly, consolidation, edema, lung opacity, pleural effusion, pneumonia and pneumothorax. These pathologies were chosen based on the overlap between pathology classes present in the existing datasets and the CheXbert classifier [242]. In Fig. E.5 and Table E.6, some representative image and

Table E.5: Distribution of the annotation pairs (image bounding-box and sentence) across different clinical findings. The demographic statistics (e.g., gender, age) of the subjects are collected from MIMIC-IV dataset for `MS-CXR` and all MIMIC-CXR.

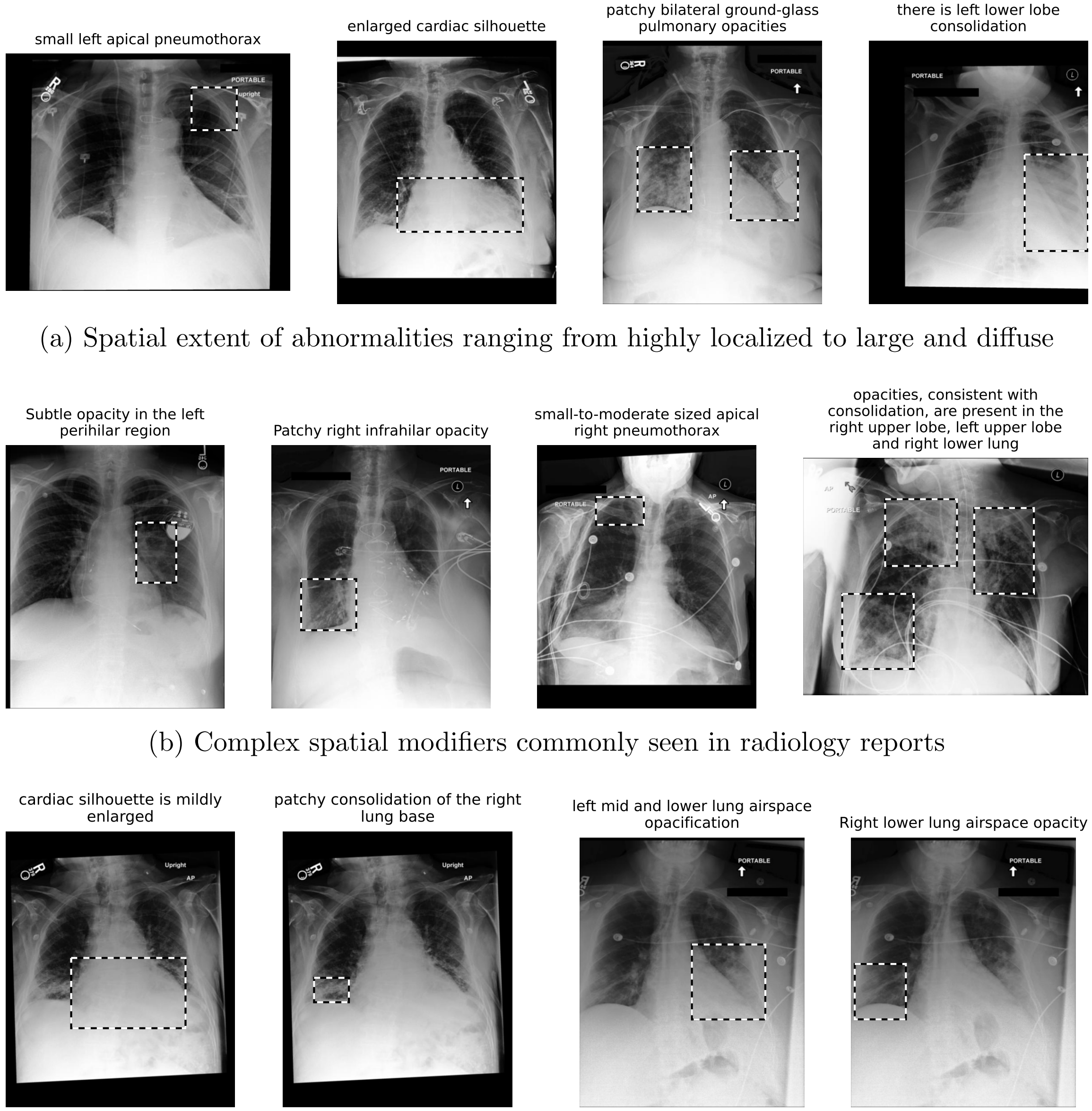
Findings	# of annotation pairs	# of subjects	Gender - F (%)	Avg Age (std)
Atelectasis	61	61	28 (45.90%)	64.52 (15.95)
Cardiomegaly	333	282	135 (47.87%)	68.10 (14.81)
Consolidation	117	109	40 (36.70%)	60.08 (17.67)
Edema	46	42	18 (42.86%)	68.79 (14.04)
Lung opacity	81	81	33 (40.24%)	62.07 (17.20)
Pleural effusion	96	95	41 (43.16%)	66.36 (15.29)
Pneumonia	182	146	65 (44.52%)	64.32 (17.17)
Pneumothorax	237	151	66 (43.71%)	60.71 (18.04)
Total	1153	851	382 (44.89%)	64.37 (16.61)
Background (all MIMIC-CXR)	-	65379	34134.0 (52.39%)	56.85 (19.47)

text examples from `MS-CXR` are shown. Additionally, the distribution of samples across the pathology classes is shown in Table E.5 together with demographics across subjects in `MS-CXR`.

Differences to Existing Annotations. The proposed benchmark builds on top of publicly available bounding-box/ellipse annotations in REFLACX [20] and MIMIC-CXR-Annotations [248], where the latter also contains simplified text phrases for pneumonia and pneumothorax. `MS-CXR` extends and curates these annotation sets by (I) adding a new set of studies to cover a wider range of clinical findings and pathologies, (II) reviewing the clinical correctness and suitability of the existing annotations for the grounding task (see Section 4.2.2), (III) creating, verifying, and correcting bounding boxes where necessary, and (IV) pairing them up with real clinical descriptions extracted from MIMIC-CXR reports if none were present. Most importantly, the textual descriptions paired with dense image region annotations are sampled from the original distribution of word tokens, which capture dense text semantics and are better aligned with real-world clinical applications that build on good local alignment.

E.3.1 Label Collection and Review

Original MIMIC reports and REFLACX [20] radiology transcripts are first parsed by extracting sentences to form a large pool of text descriptions of pathologies. These candidates are later filtered by deploying the CheXbert [242] text classifier, in order to keep only the phrases associated with the target pathologies whilst ensuring the following two criteria: (I) For a given study, there is only one sentence describing the target pathology, and (II) the sentence does not mention more than one findings that are irrelevant to each other. After extracting the text descriptions, they are paired with image annotations on a study level. At the final stage, a review process is


small left apical pneumothorax
enlarged cardiac silhouette
patchy bilateral ground-glass pulmonary opacities
there is left lower lobe consolidation
Subtle opacity in the left perihilar region
Patchy right infrahilar opacity
small-to-moderate sized apical right pneumothorax
opacities, consistent with consolidation, are present in the right upper lobe, left upper lobe and right lower lung
cardiac silhouette is mildly enlarged
patchy consolidation of the right lung base
left mid and lower lung airspace opacification
Right lower lung airspace opacity

(a) Spatial extent of abnormalities ranging from highly localized to large and diffuse

(b) Complex spatial modifiers commonly seen in radiology reports

(c) Multiple pathologies reported for the same study

(d) Findings with multiple spatial locations reported separately

Figure E.5: Some examples illustrating important axes of variability present in the MS-CXR dataset. Text descriptions include clinical findings of varying spatial extent (a) and a range of different spatial modifiers (b). Additionally, a subset of studies contain multiple bounding-box and sentence annotations per image (c–d).

conducted with two board certified radiologists mainly to verify the match between the text and bounding box candidates. Moreover, in this review process, the suitability of the annotation pairs for the grounding task were also assessed, whilst ensuring clinical accuracy. In detail, the phrase-image samples are filtered out if at least one of following conditions is met:

1. Text describing a finding not present in the image.
2. Phrase/sentence does not describe a clinical finding or describes multiple unrelated abnormalities that appear in different lung regions.
3. There is a mismatch between the bounding box and phrase, such as image annotations are placed incorrectly or do not capture the true extent of the abnormality.
4. High uncertainty is expressed regarding reported findings, e.g. "there is questionable right lower lobe opacity".
5. Chest X-ray is not suitable for assessment of the finding or has poor image quality.
6. Text contains differential diagnosis or longitudinal information that prohibits correct grounding via the single paired image.
7. Sentences longer than 30 tokens, which often contain patient meta-information that is not shared between the two modalities (e.g., de-identified tokens).

Note that only phrases containing multiple findings are filtered out, not images with multiple findings. For instance, if an image contains both pneumonia and atelectasis, with separate descriptions for each in the report, then two instances of phrase-bounding box pairs are created. Among those candidate annotations automatically extracted from radiology reports [136] or dictated transcripts [20], 222 of out 817 were rejected and not included in `MS-CXR`. Here the raw text data were first processed with an algorithm to extract caption candidates for the review process. The same review process is applied to adjudicate the annotation pairs released in [248], and 53 out of 367 pairs were rejected and not included in `MS-CXR`.

To further increase the size of the dataset, and to balance samples across classes, additional CXR studies are sampled at random, conditioned on the underrepresented pathologies. The following procedure is applied to create the pairs of image and text annotations for these selected studies: Text descriptions are extracted using the same methodology outlined above, using MIMIC-CXR and ImaGenome datasets [271], where the latter provides sentence extracts from a subset of MIMIC-CXR dataset for clinical findings. However, differently from the initial step, the corresponding bounding box annotations (either one or more per sentence) are created from scratch by radiologists for the finding described in the text, and the same filtering as above is

Table E.6: Example findings in MS-CXR with complex syntactic structures. Please note how radiological sentences are most often not just a simple statement of the form "[class1, class2, ...]" that can be parsed with a simple bag-of-words approach, as in typical natural image captioning benchmarks (e.g., "A couple getting married" retrieved from Flickr30k [211]).

Sentence	Difficulty	Class
"Abnormal opacity in the basilar right hemithorax is likely atelectasis involving the right lower and middle lobes"	Complex syntactic structure	Atelectasis
"Multisegmental lower lobe opacities are present, consistent with areas of consolidated and atelectatic lung"	Complex syntactic structure	Atelectasis
"Parenchymal opacification in the mid and lower lung"	Less common expression	Pneumonia
"Air bronchograms extending from the left hilum throughout the left lung which has the appearance of infection"	Complex location description	Pneumonia
"Persistent focal bibasilar opacities, most consistent with infection"	Domain-specific modifier	Pneumonia
"Widespread infection, less severe on the left"	Location partially specified	Pneumonia
"Airspace consolidation in the right upper, right middle and lower lobes"	Multiple locations	Pneumonia
"Subsegmental-sized opacities are present in the bilateral infrahilar lungs"	Domain specific modifiers	Lung opacity
"There continues to be a diffuse bilateral predominantly interstitial abnormality in the lungs with more focal vague opacity in the left upper peripheral lung"	Complex syntactic structure	Lung opacity
"Left apical pneumothorax"	Domain-specific modifier	Pneumothorax
"Fluid level posteriorly, which represents a loculated hydropneumothorax"	Domain-specific language	Pneumothorax
"Mild-to-moderate left pneumothorax"	Severity modifier	Pneumothorax
"There is no pulmonary edema or pneumothorax, but small pleural effusions are still present"	Negated disease entities	Pleural effusion
"Pleural effusions are presumed but impossible to quantify, except say they are not large"	Complex sentence structure	Pleural effusion

applied by the annotator to discard candidates if the image and/or sentence is found unsuitable for the grounding task.

Analysis of Average Phrase Length. The average number of tokens (inc. full-stop) across all phrases is calculated for each benchmark dataset to better understand the characteristics of the dataset and domain. In that regard, the phrases released in [248] has an average of 6.76 tokens per sample and MS-CXR has an average of 7.49 of tokens per sample. The auto-extracted radiology sentences from transcriptions [20] whereas has an average of 8.49 tokens per sample. Relatively long sentences, auto-extracted from transcripts [20], were rejected more often in the review process, as

Table E.7: Example findings in ImaGenome which would make grounding of phrases difficult.

Sentence	Difficulty	Annotated Finding
"Even though Mediastinal veins are more distended, previous pulmonary vascular congestion has improved slightly, but there is more peribronchial opacification and consolidation in both lower lobes which could be atelectasis or alternatively results of recent aspiration, possibly progressing to pneumonia."	Multiple findings, uncertainty, different subparts of lung	Pneumonia
"Moderate right pleural effusion and bilateral heterogenous airplace opacities, concerning for pneumonia."	Multiple findings, differing laterality	Pneumonia
"It could be an early infection"	Region unclear	Pneumonia
"There is also a new small left-sided pleural effusion."	Differential diagnosis, there could be another effusion	Effusion

they often describe multiple clinical findings located in different image regions. This observation further emphasizes the importance of a review process of annotation pairs by the domain experts.

Patient Demographics. As shown in Table E.5, the average age of subjects in `MS-CXR` is higher than the average for all subjects in MIMIC-CXR. This observation can be explained with the fact that studies from healthy subjects that do not display any anomalous findings are not sampled for `MS-CXR` , and these are statistically likely to be younger. Similarly, it is not expected that gender bias is present due to the sampling strategy, as none of the pathologies that were sample are gender-specific. Overall `MS-CXR` does not deviate far from the MIMIC-CXR distribution.

E.4 Additional Related Work

This section provides additional related work to complement the related work of the main document, which focused on weak supervision.

Biomedical VLP Representation Learning. Several studies [124, 128, 166, 194, 292] have explored joint representation learning for paired image and text data in the medical domain. Contrastive VIsual Representation Learning from Text (ConVIRT) [292] uses a contrastive learning formulation for instance-level representation learning from paired medical images and text. The authors uniformly sample sentences and maximize their similarity to true augmented paired images via the InfoNCE contrastive loss [205], while reducing similarity between negative pairs in the same batch. [128, 194] both introduce approaches that combine instance-level image–report contrastive learning with local contrastive learning for medical data. In contrast, [166] use a local-only objective in an approach that approximates the mutual information between grid-like local features of images and sentence-level text features of medical data. The formulation learns image and text encoders as well as a discriminator trained to distinguish positive and negative pairs. While most related approaches use no ground truth, [40] study a semi-supervised edema severity classification setting, and [116] assume sets of seen and unseen labels towards zero-shot classification on CXR data. [165] evaluate pretrained joint embedding models—general domain VLP representation learning models that use a transformer to learn a joint embedding—by fine-tuning the models on CXR data.

Multiple CXR datasets exist that enable a partial evaluation of phrase grounding, but all come with some limitations which the `MS-CXR` dataset (see Section 4.2.2) aims to mitigate. VinDr [199], RSNA Pneumonia [237], and the NIH Chest X-ray Dataset [266] are datasets that provide bounding-box image annotations, but lack accompanying free-text descriptions. REFLACX [20] provides gaze locations captured with an eye tracker, dictated reports and some ground truth annotations for gaze locations, but no full phrase matches to image regions. Phrase annotations for MIMIC-CXR data released in [248] are of small size (350 studies), only contain two abnormalities, and for some samples have shortened phrases that were adapted to simplify the task. ImaGenome [271] provides a large number of weak local labels for CXR images and reports, with a focus on anatomical regions. However, its ground-truth set is smaller (500 studies), bounding-box regions annotate anatomical regions rather than radiological findings. Furthermore, ImaGenome sentence annotations are not curated, see Table E.7 for some examples. Sentences often contain multiple diseases as well as uncertain findings, making an accurate, largely noiseless grounding evaluation difficult. Some sentences also contain differential diagnosis and temporal change information, which cannot be grounded without access to prior scans.

Language Modeling in Radiology. Most recent general domain VLP work relies on transformer based contextual word embedding models, in particular BERT [79], pretrained on general domain data from newswire and web domains such as Wikipedia. But specific domains often exhibit differences in linguistic characteristics from general text and even related domains, such as between clinical and non-clinical biomedical text as noted in [4], motivating the use of more specialized language models in most related work with a focus on the medical domain. Here, related multi-modal work commonly uses publicly available models including BioBERT [157], ClinicalBERT [4], BioClinicalBERT [4], or PubMedBERT [107], which are either trained from scratch or fine-tuned via continual pretraining using a Masked Language Modeling (MLM) objective. Sometimes additional objectives are added such as adversarial losses [173] or Next Sentence Prediction. [107] provide evidence that training language models from scratch for specialized domains with abundant amounts of unlabeled text can result in substantial gains over continual pretraining of models first fit to general domain text. The specialized corpora these biomedical and clinical domain models use include PubMed abstracts and PubMed Central full texts, and de-identified clinical notes from MIMIC-III [137]. All the aforementioned language models have a pre-specified vocabulary size consisting of words and subwords, usually 30,000 words in standard BERT. The in-domain vocabulary plays a particularly important role in representative power for a specialized domain. A vocabulary that is not adapted will break up more words into subwords and additionally contain word pieces that have no specific relevance in the specialized domain, hindering downstream learning (see e.g. [107]). As [107] highlight, BERT models that use continual pretraining are stuck with the original vocabulary from the general-domain corpora.

Other closely related tasks in the CXR domain that share similar NLP challenges include report summarization [61, 291], automatic report generation [51, 170, 189], and natural language inference for radiology reports [189]. Finally, while the name implies close similarity to CXR-BERT, CheXbert [242] is a BERT based sentence classification model developed for improving the CheXpert [132] labeler, and the model does not have a domain-specific vocabulary like CXR-BERT or PubMedBERT.

Note that most related work on self-supervised multi-modal learning on CXR data neither explores text augmentation, nor maintains text losses such as MLM during multi-modal training. An exception is found in [194], who use the Findings and Impression/Assessment sections of radiology reports, and randomly change the sentence order by swapping pairs of them.

E.5 Model Details

E.5.1 CXR-BERT Pretraining Details

The CXR-BERT text encoder is based on the BERT (base size) architecture [256]. An implementation available via the Huggingface transformers library [269] is adopted for this purpose. The model weights are randomly initialized and pretrained from scratch. As described in Section 4.2.1, CXR-BERT is pretrained in three phases before the joint pretraining phase. For Phase (I), the Huggingface tokenizer library[1] is used to generate a custom WordPiece vocabulary of 30k tokens. For Phase (II), the AdamW [178] optimizer with a batch size of 2048 sequences and a linear learning rate schedule over 250k training steps with a 5% warm up period is used. A base learning rate of 4e-4 is set. Following RoBERTa [174], multiple sentences are packed into one input sequence of up to 512 tokens, and dynamic whole-word masking is employed. In Phase (III), pretraining of the model is continued using only MIMIC-CXR text reports. In addition to the MLM loss, the RSM loss is added to pretrain the projection layer. The projection layer P_{txt} is used to project the 768-dimensional feature vector $\tilde{\mathbf{t}}$ to a 128-dimensional report representation $\mathbf{t}$. The AdamW optimizer

[1] `https://github.com/huggingface/tokenizers`

Table E.8: Hyper-parameter values used for image data augmentations.

	Image-Text Pretraining	Image-only Pretraining	Fine-tuning for Downstream Tasks
Affine transform – shear	15°	40°	25°
Affine transform – angle	30°	180°	45°
Color jitter – brightness	0.2	0.2	0.2
Color jitter – contrast	0.2	0.2	0.2
Horizontal flip probability	-	0.5	0.5
Random crop scale	-	(0.75, 1.0)	-
Occlusion scale	-	(0.15, 0.4)	-
Occlusion ratio	-	(0.33, 0.3)	-
Elastic transform (σ, α) [240]	-	(4, 34)	-
Elastic transform probability	-	0.4	-
Gaussian noise	-	0.05	-

with a batch size of 256 sequences and a linear learning rate schedule over 100 epochs with a 3% warm up period is used. The base learning rate is set to 2e-5.

E.5.2 Image Encoder

Pretraining Details. For the image encoder, the ResNet50 [117] architecture is chosen. The 2048-dimensional feature maps $\tilde{\mathbf{V}}$ of the ResNet50 are projected to 128-dimensional feature maps $\mathbf{V}$ using a two-layer perceptron P_{img} implemented with 1×1 convolutional layers and batch-normalization [131]. The global image representation $\mathbf{v}$ is obtained by average-pooling the projected local features $\mathbf{V}$. Prior to image-text joint training, the model weights are randomly initialized and pretrained on MIMIC-CXR images using SimCLR [45] — an image-only self-supervised learning approach. A large-batch optimization (LARS) technique [281] is used on top of ADAM with a batch size of 256 and a linear learning rate scheduler over 100 epochs with a 3% warm up period. The base learning rate is set to 1e-3.

Augmentations. For each training stage, a different set of custom image augmentations is applied in order to have a better control over the learned feature invariances (e.g., laterality). During the image-text joint pretraining stage, affine transformations (random rotation and shearing) and contrast and brightness color jitter are used. Unlike ConVIRT [292] and GLoRIA [128], horizontal flips are not applied during the joint training in order to preserve location information (e.g. "pneumonia in the left lung"). During the image-only SSL (SimCLR) pretraining phase, additional image augmentations are used including random occlusion, additive Gaussian noise, and elastic spatial transforms [240]. Implementations available in the torchvision library[2] are used for this purpose. The image augmentation parameters and their corresponding values are listed in Table E.8. Before applying these transformations, the input image intensities are normalized by re-scaling each color channel values to the $[0, 255]$ range. During inference, only center cropping and resizing is applied.

[2] `https://pytorch.org/vision/stable/transforms.html`