Losses over Labels: Weakly Supervised Learning via Direct Loss Construction

Dylan Sam[1], **J. Zico Kolter**[1,2]

[1] Machine Learning Department, Carnegie Mellon University
[2] Bosch Center for Artificial Intelligence
dylansam@andrew.cmu.edu, zkolter@cs.cmu.edu

Abstract

Owing to the prohibitive costs of generating large amounts of labeled data, programmatic weak supervision is a growing paradigm within machine learning. In this setting, users design heuristics that provide noisy labels for subsets of the data. These weak labels are combined (typically via a graphical model) to form pseudolabels, which are then used to train a downstream model. In this work, we question a foundational premise of the typical weakly supervised learning pipeline: given that the heuristic provides all "label" information, why do we need to generate pseudolabels at all? Instead, we propose to directly transform the heuristics themselves into corresponding loss functions that penalize differences between our model and the heuristic. By constructing losses directly from the heuristics, we can incorporate more information than is used in the standard weakly supervised pipeline, such as *how* the heuristics make their decisions, which explicitly informs feature selection during training. We call our method Losses over Labels (LoL) as it creates losses directly from heuristics without going through the intermediate step of a label. We show that LoL improves upon existing weak supervision methods on several benchmark text and image classification tasks and further demonstrate that incorporating gradient information leads to better performance on almost every task.

1 Introduction

Recent advances in deep learning are enabled by the availability of large labeled datasets. However, expertly labeled data can be very costly to obtain, causing a bottleneck in many deep learning applications. Fortunately, in the absence of labeled data, we can leverage domain knowledge or auxiliary information for a given task. Many subfields of machine learning have tackled this problem, including semi-supervised learning, weakly supervised learning, and self-supervised learning. Although the specific methods employed by these approaches are quite distinct, many of them operate under a common theme. These approaches often work by generating **pseudolabels** (Lee 2013; Cascante-Bonilla et al. 2020; Pham et al. 2021), which can be plugged into a standard supervised learning pipeline to optimize a model on large amounts of unlabeled data. This paper focuses on their particular use in the paradigm of programmatic weak supervision (Zhang et al. 2022a), which we will refer to as weak supervision throughout this paper.

In the setting of weak supervision, we assume the presence of *weak labelers* that are commonly hand-engineered heuristics (we use the terms weak labeler and heuristic interchangeably, although weak labelers are more general and can have more varied structures). As an example, consider a sentiment classification task for restaurant reviews; a heuristic might associate the word "delicious" with a positive label, although this is clearly an imperfect label. Most recent advancements in weak supervision propose approaches that aggregate the outputs of multiple such heuristics to produce pseudolabels, typically through a graphical model (Ratner et al. 2017). These pseudolabels are used in combination with unlabeled data to optimize a downstream model, commonly referred to as an **end model**. This line of work has demonstrated its widespread applicability and efficiency (Bach et al. 2019) to learn from noisy information.

In this paper, we question this fundamental approach to weak supervision: specifically, why is there any need to produce pseudolabels at all? This standard approach distills the heuristics, which are rich sources of information, entirely into a single pseudolabel, to fit into a standard supervised learning pipeline. Instead, we propose to *directly transform heuristics into loss functions* and train a network to minimize a combination of these loss functions (Figure 1). We refer to our method as Losses over Labels (LoL), as we produce a combination of loss functions rather than intermediate pseudolabels. The simplest form of these losses is a smoothed variant of the number of times our model's prediction differs from the heuristic's prediction; this is similar to a form of "soft" pseudolabels. Building on this, our losses incorporate additional information from the heuristics, such as *which features are used in their decisions*. We add a penalty term to each loss, penalizing a model when its gradients do not match the heuristics' gradients on these important features. Incorporating this gradient information improves feature selection during optimization, which has not been previously considered in weak supervision and which indeed cannot be naturally performed with pseudolabels alone. Intuitively, this gradient information constrains our hypothesis space, removing lots of incorrect hypotheses that do not use correct features. Since our method uses the same heuristics as other weakly supervised algorithms, it does not require

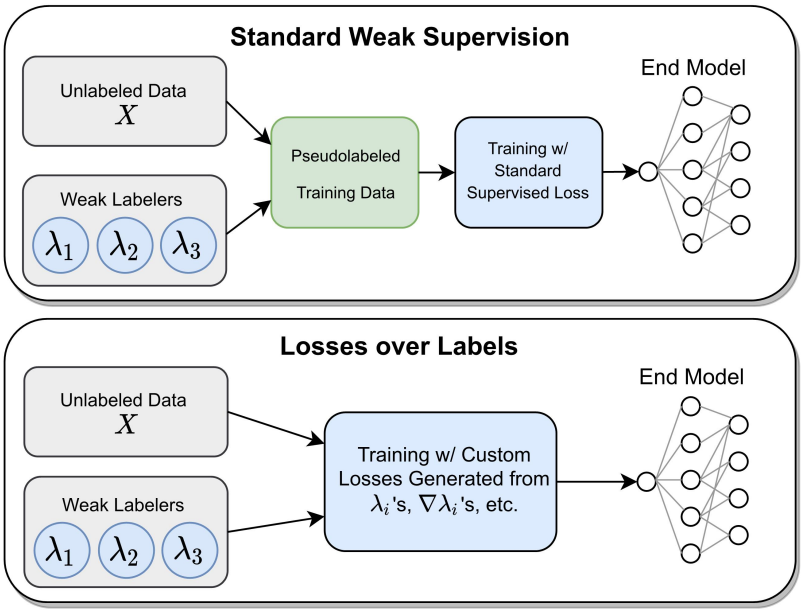

Figure 1: Depiction of our LoL method (bottom), compared to a standard weakly supervised pipeline (top). Our loss functions are generated by Equation 3.

any additional information over existing methods; it just accesses the heuristic's underlying decision mechanism.

We demonstrate that LoL performs favorably when compared to methods that create intermediate pseudolabels on five standard tasks from a weak supervision benchmark (Zhang et al. 2021). Our results support that our approach to incorporate gradient information especially helps in the regime when we do not have access to abundant unlabeled data. As other methods have to implicitly learn important features from data, LoL directly incorporates some of this information from the heuristics and, thus, requires less unlabeled data. Finally, we demonstrate that even when this gradient information is implicitly learnt (i.e., pretrained models on an auxiliary task), incorporating it during training still is beneficial to performance. As a whole, our results suggest that weakly supervised methods do not need to generate pseudolabels and can be further improved by directly translating heuristics into more informative loss functions.

2 Related Work

The problem of learning from weak or noisy sources of information has been studied in many different contexts.

Noisy Labels Various approaches have tackled the problem of learning from a single source of noisy labels. Many of these techniques assume that these labels are corrupted with random noise (Natarajan et al. 2013; Guan et al. 2018) and provide methods to address this in the optimization procedure or average over multiple splits of the dataset to denoise these corrupted labels. More recent techniques consider class-conditional noise or noise that is instance-dependent (Han et al. 2020) and provide techniques to correct such noise in the multiclass setting (Patrini et al. 2017). However, a fundamental distinction from this line of work is that weak supervision considers multiple sources of such noisy information.

Crowdsourcing Learning from multiple noisy sources has been classically studied in the field of crowdsourcing, where the goal is to combine human-annotated labels from multiple sources of varying quality. Many works in crowdsourcing combine these weak signals to recover the true labels. These approaches (Dawid and Skene 1979; Nitzan and Paroush 1982; Gao and Zhou 2013) make strong assumptions about the noisy annotators and derive optimal approaches to combine them under these assumptions. Prior work has extended this setting to allow abstentions (Dalvi et al. 2013). More recent crowdsourcing work has connected the field to modern machine learning applications by analyzing models trained on noisy crowdsourced labels (Zhang, Wu, and Sheng 2016). All of these works focus on aggregating crowdsourced annotators into a pseudolabel to potentially optimize a downstream model, while our approach proposes generating loss functions.

Ensemble Methods Aggregating multiple functions has also been studied in the field of ensemble methods. In boosting (Schapire 1990; Freund 1995; Chen and Guestrin 2016), methods aggregate multiple weak learners into a strong classifier. Bagging (Breiman 1996) aggregates multiple models to reduce variance from the individual (potentially overfitting) classifiers through bootstrapped samples. Ensemble methods have also been specifically studied in the semi-supervised setting (Balsubramani and Freund 2015b,a), creating a combination of classifiers via a minimax game. While these are all similar to combining weak labelers, their focus is to produce a weighted combination of multiple weak learners rather than training a downstream model.

Multi-task Learning Multi-task learning is a related

field that focuses on optimizing multiple loss functions, which are derived from different tasks. Here, approaches train a single model on multiple learning tasks simultaneously (Caruana 1993). Many approaches have focused on formalizing a notion of task similarity (Ben-David and Schuller 2003; Shui et al. 2019) to guide learning. Recent applications of multi-task learning to neural networks use some form of parameter sharing (Argyriou, Evgeniou, and Pontil 2007; Yang and Hospedales 2017; Ruder et al. 2017; Shui et al. 2019). We remark that multi-task learning is different from our setting as our weak labelers are designed for the *same task*, just with differing qualities and coverages. While a connection between weak supervision and multi-task learning has been made in previous work (Ratner et al. 2019), this approach still combines information from multiple tasks into a single pseudolabel via a graphical model, rather than considering the heuristics individually.

Weak Supervision The paradigm of weak supervision assumes the presence of weak labelers of varying quality that label subsets of the data. A seminal work (Ratner et al. 2016) first formulated combining these noisy sources of information via a graphical model, which produces pseudolabels used to train an end model. Most recent advances in weakly supervised learning have trended towards producing higher quality pseudolabels, frequently through more complex graphical models (Ratner et al. 2019; Fu et al. 2020). Additional works have developed variants of these graphical models to extend the weakly supervised setting to handle other objectives (Safranchik, Luo, and Bach 2020; Shin et al. 2022) and more general forms of weak supervision (Yu, Ding, and Bach 2022; Zhang et al. 2022b). Another line of weak supervision research has improved these methods by assuming additional information, i.e, a semi-supervised setting (Awasthi et al. 2020; Xu et al. 2021), an active learning setting (Biegel et al. 2021), or prior information on weak labeler accuracies (Arachie and Huang 2019; Mazzetto et al. 2021a,b; Arachie and Huang 2021, 2022). Finally, other works have developed more elaborate downstream training methods, primarily focused on deep learning models (Karamanolakis et al. 2021; Ruhling Cachay, Boecking, and Dubrawski 2021; Yu et al. 2021). Overall, recent advances in weak supervision have focused on improving pseudolabels and potentially assuming access to more information. We remark that our method looks to improve learning from weak supervision in an orthogonal direction by generating losses.

3 Preliminaries

Our classification task has some domain $\mathcal{X}$ and a set of multiple discrete labels $\mathcal{Y}$ where $|\mathcal{Y}| = k$. We assume that data is given by some underlying distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. Our goal is to learn a classifier $h : \mathcal{X} \to [0, 1]^k$ that maps examples to a probability distribution over labels $\mathcal{Y}$, which minimizes standard 0-1 risk (on the argmax of the outputs of h).

In the weakly supervised paradigm, we only observe training examples $X = \{x_1, ..., x_n\} \subset \mathcal{X}$. In the absence of labels for these examples, weak labelers are given as a set of m functions $\lambda = \{\lambda_1, ..., \lambda_m\}$, and each $\lambda_i : \mathcal{X} \to \mathcal{Y} \cup \{\emptyset\}$ where $\emptyset$ denotes an abstention. Abstentions allow more flexibility in the design of weak labelers, as users can create weak labelers that have high accuracy on specific regions of the data and abstain everywhere else. While weak labelers are permitted to abstain on data points, our classifier h is not able to abstain and must always make a prediction for all x.

4 Losses over Labels

Our main contribution is that, instead of producing pseudolabels, we propose to optimize a combination of loss functions that are derived directly from the weak labelers. This approach is arguably simpler than previous graphical models and still captures all "label" information used to generate these pseudolabels. In fact, this allows us to retain more information from these multiple sources (such as which features inform their decisions) during the optimization procedure. Intuitively, this improves the training process by aiding (and potentially correcting) feature selection with gradient information from the heuristics. We remark that this information is ignored by existing weak supervision methods that aggregate heuristics to produce intermediate pseudolabels.

4.1 Generating Losses from Heuristics

Formally, our approach looks to optimize an aggregation of loss functions that are derived from our heuristics. We can find some classifier $\hat{h}$

$$\hat{h} = \arg\min_h \sum_{i=1}^{m} \frac{1}{m(x_j)} \left(\sum_{j=1}^{n} \ell_i(x_j, h) \right), \qquad (1)$$

where ℓ_i corresponds to the loss function generated from weak labeler λ_i, and $m(x) = \sum_{i=1}^{m} \mathbf{1}\{\lambda_i(x) \neq \emptyset\}$ represents the number of labelers that do not abstain on point x. Perhaps the simplest loss function that we can consider is the loss ℓ_i on data x and classifier h as

$$\ell_i(x, h) = \mathbf{1}\{\lambda_i(x) \neq \emptyset\} \cdot \ell(h(x), \lambda_i(x)), \qquad (2)$$

where ℓ can be any arbitrary loss function. This simple loss function can be easily generated given access to the heuristic function outputs. A combination of these losses represents the average loss ℓ over weak labelers that do not abstain. This is reasonable as our learnt model should conflict the fewest possible number of times with the weak labels. For the remainder of this paper, we consider ℓ to be the cross-entropy loss, which serves as a smoothed version of the number of disagreements between the trained classifier h and weak labeler λ_i. We remark that this simple combination of losses is similar to optimizing over pseudolabels with a "soft" majority vote function. In fact, under the log loss and the square loss, we recover the same objective up to a constant factor.

Proposition 1. *An aggregation of log losses is equivalent to using a soft version of a majority vote as a pseudolabel.*

Proposition 2. *An aggregation of squared losses is equivalent to using a soft version of a majority vote as a pseudolabel, up to an additive constant (with respect to our model).*

The derivations of these propositions are shown in the extended version of this paper (Sam and Kolter 2022). While this combination of naive losses recovers the same objective as using pseudolabels, this overall framework paves the road for us to create more informative and powerful loss functions from heuristics by incorporating their gradient information.

4.2 Incorporating Additional Heuristic Information via Input Gradients

We now present our approach to incorporate gradient information from the heuristics, in order to produce more complex losses. In the typical weak supervision pipeline, the weak labeler aggregation model only looks at the *outputs of the weak labelers*. As the weak labelers are hand-engineered by experts, we frequently have access to the underlying mechanism of the heuristics. Therefore, we have information about which features are used to make their decisions.

The most common form of weak labelers (in a text classification setting) is that of a rule that checks for the presence of a set of words in a given example. For a bag of words data representation, we can determine exactly which dimensions of the data our weak labelers use, by finding the indices of the words in the vocabulary. This is a rich source of information, which is thrown away when generating pseudolabels. Therefore, we propose to leverage these additional properties of the heuristics through their input gradients. We remark that this form of weak labelers is a binary-valued function and is not differentiable. Therefore, we can incorporate its "gradient information" in a principled manner by creating a smoothed approximation of these heuristics via random inputs to the heuristics. This is similar to an idea used in adversarial robustness (Cohen, Rosenfeld, and Kolter 2019).

Formally, let $\mathcal{X} = \{0,1\}^n$ and $\mathcal{Y} = \{0,1\}$, although this easily generalizes to a multiclass setting. Then, we can consider a heuristic function that looks for the presence of a word at index j, or $\lambda_i(x) = \mathbf{1}_{\{x_j=1\}}$ and provides a positive label if it is present. We can define the smoothed approximation of this heuristic $\tilde{\lambda}_i : [0,1]^n \to [0,1]^{k+1}$ as

$$\tilde{\lambda_i}(\phi) = E_{x\sim Ber(\phi)}[\lambda_i(x)],$$

where $\phi = (\phi_1, ..., \phi_n) \in [0,1]^n$. This is now a (continuous-valued) smoothed approximation of the heuristic by considering its expectation over random independent Bernoulli inputs to the function. Then $x_j = 1$ with probability ϕ_j, so

$$\tilde{\lambda}_i(\phi_j) = \begin{pmatrix} 0 \\ \phi \\ 1-\phi \end{pmatrix},$$

where the first index corresponds to class 0, the second index corresponds to class 1, and the third index corresponds to the abstain vote $\emptyset$. We can now compute a gradient of the smoothed approximation of our weak labeler $\tilde{\lambda}_i$ with respect to the Bernoulli distribution parameter ϕ. The gradient of the smoothed heuristic i at index j is given by

$$\nabla_{\phi_j} \tilde{\lambda}_i(\phi_j) = \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}.$$

However, the binary nature of these heuristics only allows for gradients of the smoothed approximations to take values of -1 or 1. As our learnt classifier takes real-valued inputs and produces real-valued outputs, we penalize our model for having an input gradient less than some hyperparameter c times $\nabla_\phi \tilde{\lambda}_i$, which serves as a threshold for our model's desired gradient. In essence, this penalizes models that do not use the same features that are used by the heuristic.

While we can analytically compute the randomized approximation of weak labelers, we can only approximate the gradient of our classifier. We can similarly define a smoothed approximation of our classifier h on random Bernoulli inputs as

$$\tilde{h}(\phi) = E_{x\sim Ber(\phi)}[h(x)],$$

and we can estimate the input gradient empirically as $\nabla_\phi \tilde{h}(\phi) = \frac{1}{t}\sum_{i=1}^{t} \nabla_{z_i} h(z_i)$ where $z = \{z_1, ..., z_t\}$ is sampled iid from $Ber(\phi)$. Then, we can construct a gradient loss function ℓ_i^* induced by a weak labeler λ_i as

$$\begin{aligned} \ell_i^*(x,h) = \ & \mathbf{1}\{\lambda_i(x) \neq \emptyset\} \cdot \Big(\ell(h(x), \lambda_i(x)) + \\ & \alpha \cdot ||\Big[c \cdot \nabla_\phi \tilde{\lambda}_i(\phi)_{\mathcal{Y}} - \nabla_\phi \tilde{h}(\phi)\Big]^+ ||_2^2\Big), \end{aligned} \tag{3}$$

where $\tilde{\lambda}_i(\phi)_{\mathcal{Y}}$ denotes the gradient of the smoothed heuristic only along *dimensions that correspond to non-abstaining votes* and where $[x]^+ = \max(x, 0)$, $\alpha > 0$, and $c > 0$. We note that can only match the gradients of the heuristic on the non-abstaining dimensions as our model h cannot abstain.

At a high level, this loss function incorporates a squared penalty for the gradient of our model being less than c times the gradient of the heuristic (along non-abstained dimensions). α serves as a hyperparameter that determines the weighting or importance of the gradient matching term, similar to a weighting parameter for regularization. We provide ablations in the extended version (Sam and Kolter 2022) to compare with other choices of this gradient penalty (e.g., linear or exponential).

Finally, we can compute the empirical risk minimizer over the average of these gradient loss functions to produce a classifier $\hat{h}_g$:

$$\hat{h}_g = \arg\min_h \sum_{i=1}^{m} \left(\frac{1}{m(x_j)} \cdot \sum_{j=1}^{n} \ell_i^*(x_j, h) \right). \tag{4}$$

We remark that matching the input gradients of our learnt model to that of the heuristics only requires unlabeled data to compute. Therefore, our approach does not require any additional information, such as labeled data, to incorporate information about how the heuristics make their decisions. While this paper focuses on this prevalent type of heuristic, it generalizes to any weak labeler that has a gradient that we can compute. We demonstrate this applicability to other forms or weak labelers and types of data, such as pretrained models on image classification tasks, in Section 5.6.

5 Experiments

In our experiments, we compare LoL to existing weakly supervised algorithms on 5 text classification datasets from WRENCH (Zhang et al. 2021), which provides the weak supervision sources. For all of our text classification tasks, we use a bag of words data representation, so we can easily compute the gradients of a smoothed version of the heuristics. Most weak labelers in WRENCH check for the presence of a word or a set of words in a given sentence. We

	MV	Snorkel	T-Mean	T-Median	E2E	**LoL**
agnews	82.3 ± 1.1	82.7 ± 0.9	–	–	74.8 ± 0.1	$\mathbf{83.4 \pm 0.1}$
chemprot	51.4 ± 0.4	51.0 ± 0.4	–	–	50.1 ± 1.2	$\mathbf{52.9 \pm 0.3}$
IMDB	80.8 ± 0.2	$\mathbf{82.1 \pm 0.5}$	79.3 ± 1.1	80.4 ± 0.8	75.2 ± 0.6	81.8 ± 0.3
Yelp	77.1 ± 1.1	77.7 ± 0.5	79.3 ± 1.1	$\mathbf{81.0 \pm 1.0}$	79.7 ± 0.2	75.9 ± 0.7
YouTube	92.9 ± 0.9	91.0 ± 0.3	90.0 ± 0.7	90.6 ± 0.5	91.7 ± 0.5	$\mathbf{94.2 \pm 0.7}$

Table 1: Results on 5 datasets from the WRENCH benchmark, averaged over 5 random seeds. Accuracies are reported as mean $\pm$ standard deviation. We note the best-performing method in bold. We omit results from T-Mean and T-Median on multiclass tasks (agnews, chemprot) as they are binary classification methods.

	MV	Snorkel	T-Mean	T-Median	E2E	**LoL**
agnews	52.9 ± 1.8	54.7 ± 1.8	–	–	46.7 ± 2.5	$\mathbf{63.2 \pm 0.7}$
chemprot	38.2 ± 0.7	36.5 ± 1.9	–	–	37.1 ± 1.1	$\mathbf{39.8 \pm 1.2}$
IMDB	67.5 ± 0.6	66.0 ± 1.0	58.3 ± 2.5	52.4 ± 0.7	61.7 ± 1.0	$\mathbf{69.6 \pm 0.4}$
Yelp	67.8 ± 1.5	67.7 ± 2.2	70.0 ± 1.7	65.0 ± 2.6	65.9 ± 1.4	$\mathbf{71.2 \pm 2.4}$
YouTube	90.9 ± 1.01	87.5 ± 2.4	88.0 ± 1.3	79.8 ± 4.5	90.2 ± 0.7	$\mathbf{92.0 \pm 0.9}$

Table 2: Results when training on only 100 unlabeled data, averaged over 5 random seeds. Accuracies are reported as mean $\pm$ standard deviation. Again, we note the best-performing method in bold.

still include weak labelers that are of a different form (e.g., a pre-trained sentiment classifier) for all methods. However, we are unable to analytically compute the gradients for these other weak labelers, so we remove those terms from the LoL loss and use the simple loss given in Equation 2. In Section 5.6, we extend our setting to consider 3 image classification tasks from the Animals with Attributes 2 dataset (Xian et al. 2018) with weak labelers that are trained models. Code for our experiments can be found here[1].

5.1 Baselines

We compare our method against weakly supervised baselines that do not require any labeled data or estimates of weak labeler error rates. Our baselines include:

Majority Vote (MV): We create pseudolabels with a vanilla majority vote where ties are broken randomly. These pseudolabels are then used to train an end model.

Snorkel MeTaL (Snorkel): We generate soft pseudolabels via the graphical model approach in Snorkel MeTaL (Ratner et al. 2019), which are used to train an end model.

Triplet Methods (T-Mean and T-Median): We generate soft pseudolabels via the triplet method described in FlyingSquid (Fu et al. 2020). Mean denotes the original approach, and median denotes the extension later for estimating median parameters (Chen et al. 2021). Both of these produce pseudolabels that are used to train an end model. We remark that these models are defined for binary classification tasks, so we cannot report their scores on multiclass tasks.

[1]https://github.com/dsam99/LoL

End to End (E2E): We compare against the end-to-end approach that jointly learns both the aggregation of weak labelers and the end model (Ruhling Cachay, Boecking, and Dubrawski 2021). We remark that this approach uses an additional neural network as an encoder to generate pseudolabels. Therefore, it uses a more complex model as compared to LoL and other baselines.

5.2 Results

We provide the results of our methods and the baselines in Table 1. LoL achieves the best performance on 3 of the 5 datasets. In addition, our method is the second best-performing method on another dataset. While T-Median and E2E are the best-performing methods on the remaining dataset, we note that LoL outperforms both of these methods on all other tasks. Even when compared to the much more complex architecture of E2E, LoL still performs favorably in most cases.

These experiments illustrate that LoL achieves frequently better performance than existing weakly supervised approaches, without the need for intermediate pseudolabels. Furthermore, this demonstrates that using information about how the heuristics make their decisions can benefit weakly supervised methods, improving overall performance.

5.3 Performance with Limited Unlabeled Data

While weakly supervised learning usually assumes abundant unlabeled data, we demonstrate the ability of LoL on settings with limited unlabeled data. We remark that LoL outperforms other methods, especially when unlabeled data is

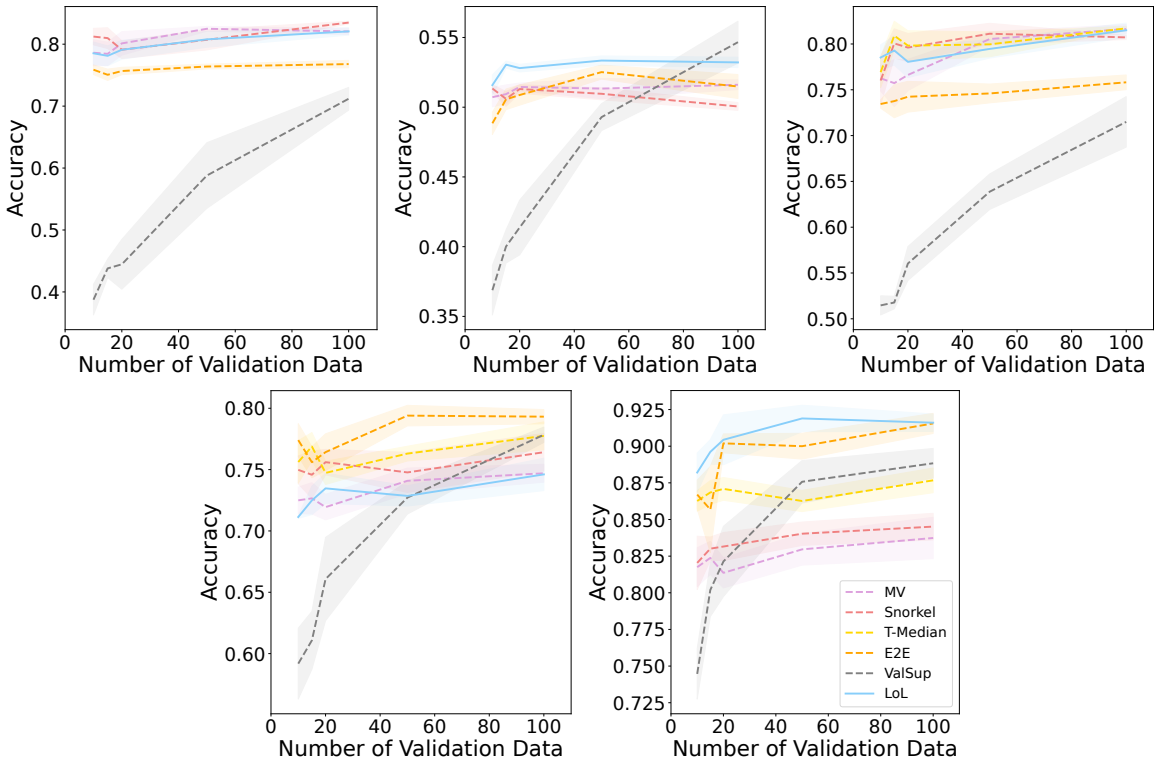

Figure 2: Results (left to right, top to bottom: agnews, chemprot, IMDB, Yelp, Youtube) as we vary validation set size. Baselines are given by dashed lines, and LoL is given by a solid line. We compute error bars (standard deviation) over 5 seeds.

also limited in quantity. This is due to the additional information from how the heuristics make their decisions; this additional gradient information helps with feature selection during the training process. On the other hand, other methods must learn this information from unlabeled data, failing when unlabeled data is limited.

We consider the same experimental setting as above in Table 2; however, we now only assume the presence of a few ($N = 100$) unlabeled training data from each task. We present these additional empirical results in Table 2. In this setting, LoL is now the best-performing method across all datasets. This is a notable result as LoL was previously outperformed on both IMDB and Yelp with large unlabeled data. In fact, we remark LoL with only 100 unlabeled data performs better than almost all other methods, using the full unlabeled training dataset (excluding MV).

The best performing baseline in most cases is MV, as other baselines (Snorkel, T, T-Median) likely produce incorrect accuracy estimates with limited unlabeled data. In addition, E2E is a more complex model, with an additional neural network as an encoder; thus, it struggles with learning from a small amount of data. These experiments demonstrate that our method especially improves performance specifically in the case when unlabeled data is limited.

5.4 Impact of Using Gradient Information

We provide an ablation study in Table 3 to directly compare using gradient information to the combination of naive losses (i.e, a soft majority vote pseudolabel), which we refer to as **LoL-simple**. This comparison looks to demonstrate the impacts of incorporating gradient information, which is the only difference between these two approaches.

	LoL	**LoL-simple**
agnews	**83.4 ± 0.1**	82.3 ± 1.2
chemprot	52.9 ± 0.3	**53.2 ± 0.2**
IMDB	**81.8 ± 0.3**	80.8 ± 0.8
Yelp	**75.9 ± 0.7**	74.8 ± 0.4
YouTube	**94.2 ± 0.7**	93.7 ± 0.8

Table 3: Comparison of LoL and LoL-simple, averaged over 5 random seeds. Accuracies are reported as mean ± standard deviation. We bold the best performing method.

We observe that in almost every dataset (except chemprot), incorporating gradient information improves upon the simple combination of labelers (that is equivalent to a soft majority vote). Only on the chemprot dataset, LoL-simple performs slightly better to LoL. This supports that gradient information is beneficial to the standard weakly supervised learning pipeline.

	MV	Snorkel	T-Mean	E2E	LoL-simple	**LoL**	**LoL-sw**
seal v. whale	90.2 ± 1.5	$\mathbf{97.1 \pm 0.6}$	93.1 ± 0.8	96.6 ± 0.6	91.8 ± 1.5	93.6 ± 1.3	95.5 ± 0.9
raccoon v. rat	75.3 ± 5.7	94.9 ± 0.9	85.1 ± 3.0	91.8 ± 2.1	69.9 ± 5.6	77.9 ± 4.1	$\mathbf{96.4 \pm 0.5}$
whale v. hippo	98.4 ± 0.4	$\mathbf{99.1 \pm 0.4}$	98.1 ± 0.8	98.4 ± 0.4	97.3 ± 0.5	98.3 ± 0.3	98.1 ± 0.4

Table 4: Results on binary classification tasks between test classes of the Animals with Attributes 2 dataset, averaged over 5 random seeds. We selected the 3 tasks with the fewest weak labelers (defined by differences in class attribute annotations). Accuracies are reported as mean $\pm$ standard deviation. $k = 20$ for LoL. The best-performing method is bolded.

5.5 Sensitivity to Amount of Validation Data

Finally, we provide experiments that demonstrate the performance of LoL as we vary the amount of validation data used for model selection. We note that LoL has two additional hyperparameters α, c; thus, it potentially requires more validation data. The WRENCH benchmark uses 10% of the entire data as validation data, which is frequently sufficient labeled data to even train a supervised model. Therefore, we provide experiments (Figure 2) that evaluate the sensitivities of our methods to the amount of validation data.

For each task, we split the dataset into 80% train and validation data and 20% test data. Then we further split training and validation data into N examples *per class* of labeled validation data. We report results for validation set sizes of $N \in \{10, 15, 20, 50, 100\}$. We additionally compare all methods against a supervised learning approach on the labeled validation data (**ValSup**). This baseline helps us assess the particular regime of our experiment; we primarily care about when traditional supervised learning fails. We note that the ValSup supervised learning baseline performs model selection on its own training set. In these experiments, we train methods for 10 epochs. Our results demonstrate that our model performs comparably to or better than other methods, with access to little validation data. On the YouTube dataset, we even outperform the supervised baseline.

5.6 Extension to Pretrained Labelers on Images

We extend our setting to consider image classification tasks with weak labelers that are trained models. We follow the standard procedure outlined in (Mazzetto et al. 2021b,a) to create weak labelers from the Animals with Attributes 2 dataset (Xian et al. 2018). These models learn to detect coarse-grained attributes from images (e.g., color, presence of water, etc.), which we use to produce noisy labels for new classes at test time. We follow the same experimental procedure as before, now using an underlying representation given by the pretrained ResNet18.

To generate our gradient constraints, we can directly compute the input gradients of the weak labelers, as they are functions defined on continuous input data. However, as these models are learnt, we do not have any explicit control over their gradients. Therefore, we only constrain our model to match the top-k gradient features of the weak labeler as they may be potentially noisy. Intuitively, we again benefit from additional feature selection as we add constraints that our models should explicitly use the most informative features for these auxiliary tasks, while not requiring the model to match less informative features.

We compare LoL against baselines on these image classification tasks in Table 4. We additionally provide a Snorkel-weighted variant of LoL, called **LoL-sw**. This essentially uses the same weights that Snorkel uses in generating pseudolabels. More discussion on different weightings of our losses is given in the extended version (Sam and Kolter 2022), which also demonstrates that optimal weighting schemes depend on the particular task of interest. We observe that our method is outperformed by Snorkel on two tasks but is the best-performing method on one task. As before, we also observe that incorporating constraints based on gradient information does lead to benefits over a simple combination of soft weights, as LoL improves over LoL-simple on all tasks. This demonstrates that incorporating gradient information from weak labelers, *even that which is implicitly learnt on auxiliary tasks*, can yield further improvements in the field of weak supervision.

6 Conclusion

In summary, we present a new and effective approach in the weakly supervised paradigm to optimize a model via a combination of loss functions that are generated directly from the weak labelers. Under this framework, these losses can leverage information about how the heuristics make decisions in our optimization, which has not been done previously by weakly supervised methods. We demonstrate that, on many text and image classification tasks, LoL achieves better performance than weakly supervised baselines and that incorporating (even learnt) gradient information almost always leads to better performance.

Our work has the potential to start a new direction of research in weakly supervised learning. While recent work in weak supervision has focused on improving the process of creating pseudolabels, our approach proposes a fundamentally different pipeline. When creating pseudolabels, we admittedly gain the benefit of using a plethora of standard supervised learning methods and models. However, we lose the ability to impose additional control over the model optimization process. Incorporating more information from the heuristics directly in our method gives experts more control over the training procedure of the model. In fact, this result has larger implications on the design of the weak labelers themselves; experts now have a greater influence on the weakly supervised pipeline, as they can induce constraints on the gradients of the learnt model via designing heuristics.

Acknowledgements

Dylan Sam was supported by funding from the Bosch Center for Artificial Intelligence and by the ARCS Foundation.

References

Arachie, C.; and Huang, B. 2019. Stochastic generalized adversarial label learning. *arXiv preprint arXiv:1906.00512*.

Arachie, C.; and Huang, B. 2021. Constrained Labeling for Weakly Supervised Learning. *ArXiv*, abs/2009.07360.

Arachie, C.; and Huang, B. 2022. Data Consistency for Weakly Supervised Learning. *ArXiv*, abs/2202.03987.

Argyriou, A.; Evgeniou, T.; and Pontil, M. 2007. Multi-Task Feature Learning. In Schölkopf, B.; Platt, J.; and Hoffman, T., eds., *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

Awasthi, A.; Ghosh, S.; Goyal, R.; and Sarawagi, S. 2020. Learning from Rules Generalizing Labeled Exemplars. In *International Conference on Learning Representations*.

Bach, S. H.; Rodriguez, D.; Liu, Y.; Luo, C.; Shao, H.; Xia, C.; Sen, S.; Ratner, A. J.; Hancock, B.; Alborzi, H.; Kuchhal, R.; Ré, C.; and Malkin, R. 2019. Snorkel DryBell: A Case Study in Deploying Weak Supervision at Industrial Scale. *Proceedings of the 2019 International Conference on Management of Data*.

Balsubramani, A.; and Freund, Y. 2015a. Optimally Combining Classifiers Using Unlabeled Data. In Grünwald, P.; Hazan, E.; and Kale, S., eds., *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, 211–225. Paris, France: PMLR.

Balsubramani, A.; and Freund, Y. 2015b. Scalable Semi-Supervised Aggregation of Classifiers. In *Neural Information Processing Systems (NeurIPS)*.

Ben-David, S.; and Schuller, R. 2003. Exploiting Task Relatedness for Multiple Task Learning. In Schölkopf, B.; and Warmuth, M. K., eds., *Learning Theory and Kernel Machines*, 567–580. Berlin, Heidelberg: Springer Berlin Heidelberg.

Biegel, S.; El-Khatib, R.; Oliveira, L. O. V. B.; Baak, M.; and Aben, N. 2021. Active WeaSuL: Improving Weak Supervision with Active Learning. arXiv:2104.14847.

Breiman, L. 1996. Bagging Predictors. *Machine Learning*, 24(2): 123–140.

Caruana, R. 1993. Multitask Learning: A Knowledge-Based Source of Inductive Bias. In *ICML*.

Cascante-Bonilla, P.; Tan, F.; Qi, Y.; and Ordonez, V. 2020. Curriculum Labeling: Self-paced Pseudo-Labeling for Semi-Supervised Learning. *ArXiv*, abs/2001.06001.

Chen, M. F.; Cohen-Wang, B.; Mussmann, S.; Sala, F.; and R'e, C. 2021. Comparing the Value of Labeled and Unlabeled Data in Method-of-Moments Latent Variable Estimation. In *AISTATS*.

Chen, T.; and Guestrin, C. 2016. XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Cohen, J. M.; Rosenfeld, E.; and Kolter, J. Z. 2019. Certified Adversarial Robustness via Randomized Smoothing. In *ICML*.

Dalvi, N.; Dasgupta, A.; Kumar, R.; and Rastogi, V. 2013. Aggregating Crowdsourced Binary Ratings. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, 285–294. New York, NY, USA: Association for Computing Machinery.

Dawid, A. P.; and Skene, A. M. 1979. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1): 20–28.

Freund, Y. 1995. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2): 256–285.

Fu, D. Y.; Chen, M. F.; Sala, F.; Hooper, S. M.; Fatahalian, K.; and Ré, C. 2020. Fast and Three-rious: Speeding Up Weak Supervision with Triplet Methods. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*.

Gao, C.; and Zhou, D. 2013. Minimax Optimal Convergence Rates for Estimating Ground Truth from Crowdsourced Labels. *CoRR*, abs/1207.0016.

Guan, D.; Wei, H.; Yuan, W.; Han, G.; Tian, Y.; Al-Dhelaan, M.; and Al-Dhelaan, A. 2018. Improving Label Noise Filtering by Exploiting Unlabeled Data. *IEEE Access*, 6: 11154–11165.

Han, B.; Yao, Q.; Liu, T.; Niu, G.; Tsang, I. W.; Kwok, J. T.; and Sugiyama, M. 2020. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406*.

Karamanolakis, G.; Mukherjee, S. S.; Zheng, G.; and Awadallah, A. H. 2021. Self-training with Weak Supervision. In *NAACL 2021*.

Lee, D.-H. 2013. Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. *Workshop on challenges in representation learning, ICML*.

Mazzetto, A.; Cousins, C.; Sam, D.; Bach, S. H.; and Upfal, E. 2021a. Adversarial Multi Class Learning under Weak Supervision with Performance Guarantees. In *International Conference on Machine Learning*, 7534–7543. PMLR.

Mazzetto, A.; Sam, D.; Park, A.; Upfal, E.; and Bach, S. 2021b. Semi-supervised aggregation of dependent weak supervision sources with performance guarantees. In *International Conference on Artificial Intelligence and Statistics*, 3196–3204. PMLR.

Natarajan, N.; Dhillon, I. S.; Ravikumar, P. K.; and Tewari, A. 2013. Learning with Noisy Labels. In Burges, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Nitzan, S.; and Paroush, J. 1982. Optimal Decision Rules in Uncertain Dichotomous Choice Situations. *International Economic Review*, 23(2): 289–97.

Patrini, G.; Rozza, A.; Krishna Menon, A.; Nock, R.; and Qu, L. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1944–1952.

Pham, H.; Xie, Q.; Dai, Z.; and Le, Q. V. 2021. Meta Pseudo Labels. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11552–11563.

Ratner, A.; Bach, S. H.; Ehrenberg, H.; Fries, J.; Wu, S.; and Ré, C. 2017. Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment*, 11(3): 269–282.

Ratner, A.; Hancock, B.; Dunnmon, J.; Sala, F.; Pandey, S.; and Ré, C. 2019. Training Complex Models with Multi-Task Weak Supervision. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 4763–4771.

Ratner, A. J.; De Sa, C. M.; Wu, S.; Selsam, D.; and Ré, C. 2016. Data programming: Creating large training sets, quickly. In *Neural Information Processing Systems (NeurIPS)*.

Ruder, S.; Bingel, J.; Augenstein, I.; and Søgaard, A. 2017. Sluice networks: Learning what to share between loosely related tasks. *arXiv preprint arXiv:1705.08142*.

Ruhling Cachay, S.; Boecking, B.; and Dubrawski, A. 2021. End-to-End Weak Supervision. *Advances in Neural Information Processing Systems*, 34.

Safranchik, E.; Luo, S.; and Bach, S. H. 2020. Weakly Supervised Sequence Tagging from Noisy Rules. In *AAAI*.

Sam, D.; and Kolter, J. Z. 2022. Losses over Labels: Weakly Supervised Learning via Direct Loss Construction. *arXiv preprint arXiv:2212.06921*.

Schapire, R. E. 1990. The strength of weak learnability. *Machine Learning*, 5(2): 197–227.

Shin, C.; Li, W.; Vishwakarma, H.; Roberts, N. C.; and Sala, F. 2022. Universalizing Weak Supervision. In *International Conference on Learning Representations*.

Shui, C.; Abbasi, M.; Robitaille, L.-É.; Wang, B.; and Gagné, C. 2019. A principled approach for learning task similarity in multitask learning. *International Joint Conferences on AI*.

Xian, Y.; Lampert, C. H.; Schiele, B.; and Akata, Z. 2018. Zero-Shot Learning—A Comprehensive Evaluation of the Good, the Bad and the Ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*.

Xu, Y.; Ding, J.; Zhang, L.; and Zhou, S. 2021. DP-SSL: Towards Robust Semi-supervised Learning with A Few Labeled Samples. In Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*.

Yang, Y.; and Hospedales, T. M. 2017. Trace Norm Regularised Deep Multi-Task Learning. *ArXiv*, abs/1606.04038.

Yu, P.; Ding, T.; and Bach, S. H. 2022. Learning from Multiple Noisy Partial Labelers. In *Artificial Intelligence and Statistics (AISTATS)*.

Yu, Y.; Zuo, S.; Jiang, H.; Ren, W.; Zhao, T.; and Zhang, C. 2021. Fine-Tuning Pre-trained Language Model with Weak Supervision: A Contrastive-Regularized Self-Training Approach. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1063–1077.

Zhang, J.; Hsieh, C.-Y.; Yu, Y.; Zhang, C.; and Ratner, A. 2022a. A survey on programmatic weak supervision. *arXiv preprint arXiv:2202.05433*.

Zhang, J.; Wang, B.; Song, X.; Wang, Y.; Yang, Y.; Bai, J.; and Ratner, A. 2022b. Creating Training Sets via Weak Indirect Supervision. In *International Conference on Learning Representations*.

Zhang, J.; Wu, X.; and Sheng, V. 2016. Learning from crowdsourced labeled data: a survey. *Artificial Intelligence Review*, 46.

Zhang, J.; Yu, Y.; Li, Y.; Wang, Y.; Yang, Y.; Yang, M.; and Ratner, A. 2021. WRENCH: A Comprehensive Benchmark for Weak Supervision. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.