Mitigating the Impact of Labeling Errors on Training via Rockafellian Relaxation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Labeling errors in datasets are common, if not systematic, in practice. They naturally arise in a variety of contexts—human labeling, noisy labeling, and weak labeling (i.e., image classification), for example. This presents a persistent and pervasive stress on machine learning practice. In particular, neural network (NN) architectures can withstand minor amounts of dataset imperfection with traditional countermeasures such as regularization, data augmentation, and batch normalization. However, major dataset imperfections often prove insurmountable. We propose and study the implementation of Rockafellian Relaxation (RR), a new loss reweighting, architecture-independent methodology, for neural network training. Experiments indicate RR can enhance standard neural network methods to achieve robust performance across classification tasks in computer vision and natural language processing (sentiment analysis). We find that RR can mitigate the effects of dataset corruption due to both (heavy) labeling error and/or adversarial perturbation, demonstrating effectiveness across a variety of data domains and machine learning tasks.

1 Introduction

Labeling errors are systematic in practice, stemming from various sources. For example, the reliability of human-generated labels can be negatively impacted by incomplete information, or the subjectivity of the labeling task - as is commonly seen in medical contexts, in which experts can often disagree on matters such as the location of electrocardiogram signal boundaries [8], prostate tumor region delineation, and tumor grading [20]. As well, labeling systems, such as Mechanical Turk[1] often find expert labelers being replaced with unreliable non-experts [27]. For all these reasons, it would be advisable for any practitioner to operate under the assumption that their dataset is corrupted with labeling errors, and possibly to a large degree.

In this paper, we propose a loss-reweighting methodology for the task of training a classifier on data having higher levels of labeling errors. We show that our method relates to optimistic and robust distributional optimization formulations aimed at addressing adversarial training (AT). These findings underscore our numerical experiments on NNs that suggest this method of training can provide test performance robust to high levels of labeling error, and to some extent, feature perturbation. Overall, we tackle the prevalent challenges of label corruption and class imbalance in training datasets, which are critical obstacles for deploying robust machine learning models. Our proposed approach implements Rockafellian Relaxations [23] to address corrupted labels and automatically manage class imbalances without the need for clean validation sets or sophisticated hyper-parameters - common constraints of current methodologies. This distinct capability represents our key contribution, making our approach more practical for handling large industrial datasets.

[1] http://mturk.com

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

We proceed to discuss related works in section 2, and our specific contributions to the literature. In section 3 we discuss our methodology in detail and provide some theoretical justifications that motivate the effectiveness of our methodology. The datasets and NN model architectures upon which our experimental results are based are discussed in sections 4 and 5, respectively. We then conclude with numerical experiments and results in section 6.

2 Related Work

Corrupted datasets are of concern, as they potentially pose severe threats to classification performance of numerous machine-learning approaches [36], including, most notably, NNs [15, 33]. Naturally, there have been numerous efforts to mitigate this effect [28, 8]. These efforts can be categorized into robust architectures, robust regularization, robust loss function, *loss adjustment*, and sample selection [28]. Robust architecture methods focus on developing custom NN layers and dedicated NN architectures. This differs from our approach, which is architecture agnostic and could potentially "wrap around" these methods. While robust regularization methods like data augmentation [26], weight decay [16], dropout [29], and batch normalization [14] can help to bolster performance, they generally do so under lower levels of dataset corruption. Our approach, on the other hand, is capable of handling high levels of corruption, and can seamlessly incorporate methods such as these. In label corruption settings, it has been shown that loss functions, such as robust mean absolute error (MAE) [10] and generalized cross entropy (GCE) [35] are more robust than categorical cross entropy (CCE). Again, our method is not dependent on a particular loss function, and it is possible that arbitrary loss functions, including robust MAE and GCE, can be swapped into our methodology with ease. Our approach resembles the loss adjustment methods most closely, where the overall loss is adjusted based on a (re)weighting scheme applied to training examples.

In *loss adjustment* methods, individual training example losses are typically adjusted multiple times throughout the training process prior to NN updates. These methods can be further grouped into loss correction, loss reweighting, label refurbishment, and meta-learning [28]. Our approach most closely resembles the *loss reweighting* methods. Under this scheme each training example is assigned a unique weight, where smaller weights are assigned to examples that have likely been corrupted. This reduces the influence of corrupted examples. A training example can be completely removed if its corresponding weight becomes zero. Indeed, a number of loss reweighting methods are similar to our approach. For example, Ren et al., [22] learn sample weights through the use of a noise-free validation set. Chang et al. [5] assign sample weights based on prediction variances, and Zhang et al. [34] examine the structural relationship among labels to assign sample weights. However, we view the need for a clean dataset, or at least one with sufficient class balance, by these methods as a shortcoming, and our method, in contrast, makes no assumption on the availability of such a dataset.

Satoshi et al. [12] propose a two-phased approach to noise cleaning. The first phase trains a standard neural network to determine the top-m most influential training instances that influence the decision boundary; these are subsequently removed from the training set to create a cleaner dataset. In the second phase, the neural network is retrained using the cleansed training set. Their method demonstrates superior validation accuracy for various values of m on MNIST and CIFAR-10. Although impressive, their method does not address the fact that most industrial datasets have a reasonably large amount of label corruption [28] which, upon complete cleansing, could also remove informative examples that lie close to the decision boundary. Additionally, the value of m is an additional hyper-parameter that could require significant tuning on different datasets and sources.

Mengye et al. [22] propose dealing with label noise and class imbalance by learning exemplar weights automatically. They propose doing so in the following steps: a) Create a pristine noise-free validation set. b) Initially train on a large, noisy training dataset, compute the training loss on the training set, train on the clean validation set, and compute the training loss on the validation set. c) Finally, compute the exemplar weights that temper the training loss computed in step two with validation loss. This approach is algorithmically the most similar to ours, with some key differences. The major difference is that it treats noise and class imbalance similarly. Our approach deals with noisy labels explicitly and can cope with almost any amount of class imbalance automatically, as tested in our experiments with the open-source Hate-Speech dataset, where we experimented with different prevalence levels of Hate-Speech text. The biggest drawback of the method proposed by Mengye et al. is that it requires a clean validation set, which in practice is almost impossible

to obtain; if it were possible, it would not be very prohibitive to clean the entire dataset. Noise, typically, is an artifact of the generative distribution which cannot be cherry-picked as easily in practice. Our approach does not require a clean dataset to be operational or effective.

3 Methodology

3.1 Mislabeling

Let $\mathcal{X}$ denote a *feature* space, with $\mathcal{Y}$ a corresponding *label* space. Then $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ will be a collection of feature-label pairs, with an unknown probability distribution D. Throughout the forthcoming discussions, $\{(x_i, y_i)\}_{i=1}^N$ will denote a sample of N feature-label pairs, for which some pairs will have a mislabeling. More precisely, we begin with a collection $(x_i, \tilde{y}_i)$ drawn i.i.d. from D, but there is some unknown set $C \subsetneq \{1, \ldots, N\}$ denoting (corrupted) indices for which $y_i = \tilde{y}_i$ if and only if $i \notin C$. For those $i \in C$, y_i is some incorrect label, selected uniformly at random, following the Noise Completely at Random (NCAR) model [8] also known as *uniform label noise*.

3.2 Rockafellian Relaxation Method (RRM)

We adopt the empirical risk minimization (ERM) [31] problem formulation:

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} J(\theta; x_i, y_i) + r(\theta) \tag{1}$$

as a baseline against which our method is measured. Given an NN architecture with (learned) parameter setting θ that takes as input any feature x and outputs a prediction $\hat{y}$, $J(\theta; x, y)$ is the loss with which we evaluate the prediction $\hat{y}$ with respect to y. Finally, $r(\theta)$ denotes a regularization term.

In ERM it is common practice to assign each training observation i a probability $p_i = 1/N$. However, when given a corrupted dataset, we may desire to remove those samples that are affected; in other words, if $C \subsetneq \{1, ..., N\}$ is the set of corrupted training observations, then we would desire to set the probabilities in the following alternative way:

$$p = (p_1, ..., p_N) \text{ with } p_i = \begin{cases} 0, & \text{if } i \in C \\ \frac{1}{N - |C|}, & \text{if } i \in \{1, ..., N\} \setminus C, \end{cases} \tag{2}$$

where $|C|$ is the cardinality of the unknown set C. In this work, we provide a procedure - the *Rockafellian Relaxation Method* (RRM) - with the intention of aligning the p_i values closer to the desired (but unknown) p of (2) in self-guided, automated fashion. It does so by adopting the Rockafellian Relaxation approach of [23]. More precisely, we consider the problem

$$\min_{\theta} \Big[v(\theta) := \min_{u \in U} \sum_{i=1}^{N} (\frac{1}{N} + u_i) \cdot J(\theta; x_i, y_i) + \gamma \|u\|_1 \Big], \tag{3}$$

where $U := \{u \in \mathbb{R}^N : \sum_{i=1}^N u_i = 0, \frac{1}{N} + u_i \geq 0 \ \ \forall i = 1, \ldots, N\}$, and some $\gamma > 0$.

We proceed to comment on this problem that is nonconvex in general, before providing an algorithm.

3.3 Analysis and Interpretation of Rockafellian Relaxation

Although problem (3) is nonconvex in general, the computation of $v(\theta)$ for any fixed θ amounts to a linear program. The following result characterizes the complete set of solutions to this linear program, and in doing so, provides an interpretation of the role that γ plays in the loss-reweighting action of RRM.

Theorem 3.1. *Let $\gamma > 0$ and $c = (c_1, \ldots, c_N) \in \mathbb{R}^N$, with $c_{min} := \min_i c_i$, and $c_{max} := \max_i c_i$. Write $I_{min} := \{i : c_i = c_{min}\}$, $I_{big} := \{i : c_i = c_{min} + 2\gamma\}$, and for any $S_1 \subseteq I_{min}, S_2 \subseteq I_{big}$,*

define the polytope $U^*_{S_1,S_2} := \left\{ u^* \in U : \begin{array}{c} u^*_i \geq 0 \ \ \forall i : c_i = c_{min} \\ u^*_i = 0 \ \ \forall i : c_i \in (c_{min}, c_{min} + 2\gamma) \\ u^*_i = -\frac{1}{N} \ \ \forall i \in I_{big} \setminus S_2 \\ u^*_i = -\frac{1}{N} \ \ \forall i : c_i > c_{min} + 2\gamma \\ u^*_i = 0 \ \ \forall i \in S_1 \cup S_2 \end{array} \right\}$. *Then*

$$conv\left(\cup_{S_1,S_2} U^*_{S_1,S_2}\right) = \arg\min_{u \in U} \sum_{i=1}^{N} (\frac{1}{N} + u_i) \cdot c_i + \gamma \|u\|_1. \tag{4}$$

The theorem explains that the construction of any optimal solution u^* essentially reduces to categorizing each of the losses among $\{c_i = J(\theta; x_i, y_i)\}_{i=1}^N$ as "small" or "big", according to their position in the partitioning of $[c_{min}, \infty) = [c_{min}, c_{min} + 2\gamma) \cup [c_{min} + 2\gamma, \infty)$. For losses that occur at the break points of c_{min} and $c_{min} + 2\gamma$, this classification can be arbitrary - hence, the use of S_1 and S_2 set configurations to capture this degree of freedom.

In particular, those points with losses c_i exceeding $c_{min} + 2\gamma$ are down-weighted to zero and effectively removed from the dataset. And in the event that $c_{max} - c_{min} < 2\gamma$, no loss reweighting occurs. In this manner, while lasso produces sparse solutions in the model parameter space, RRM produces sparse weight vectors by assigning zero weight to data points with high losses.

Consequently, if $\chi := \{i : c_i \in (c_{min} + 2\gamma, \infty)\}$ converges over the course of any algorithmic scheme, e.g., Algorithm 1, to some set C, then we can conclude that these data points are effectively removed from the dataset even if the training of θ might proceed. This convergence was observed in the experiments of Section 6. It is hence of possible consideration to tune γ for consistency with an estimate $\alpha \in [0, 1]$ of labeling error in the dataset $\{(x_i, y_i)\}_{i=1}^N$. More precisely, we may tune γ so that $\frac{|\chi|}{N} \approx \alpha$.

3.4 RRM and Optimistic Wasserstein Distributionally Robust Optimization

In this section, we discuss RRM's relation to distributionally robust and optimistic optimization formulations. Indeed, (3)'s formulation as a min-min problem bears resemblance to optimistic formulations of recent works, e.g., [19]. We will see as well that the minimization in u, as considered in Theorem 3.1, relates to an approximation of a data-driven Wasserstein Distributionally Robust Optimization (DRO) formulation [30].

3.4.1 Loss-reweighting via Data-Driven Wasserstein Formulation

For this discussion, as it relates to reweighting, we will lift the feature-label space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. More precisely, we let $\mathcal{W} := \mathbb{R}_+$ denote a space of *weights*. Next, we say $\mathcal{W} \times \mathcal{Z}$ has an unknown probability distribution $\mathcal{D}$ such that $\pi_{\mathcal{Z}}\mathcal{D} = D$ and $\Pi_{\mathcal{W}}\mathcal{D}(\{1\}) = 1$. In words, all possible (w.r.t. D) feature-label pairs have a weight of 1. Finally, we define an *auxiliary loss* $\ell : \mathcal{W} \times \mathcal{Z} \times \Theta$ by $\ell(w, z; \theta) := w \cdot J(x, y; \theta)$, for any $z = (x, y) \in \mathcal{Z}$.

Given a sample $\{(1, x_i, y_i)\}_{i=1}^N$, just as in Section 3.2, we can opt not to take as granted the resulting empirical distribution $\mathcal{D}_N$ because of the possibility that $|C|$-many have incorrect labels (i.e., $y_i \neq \tilde{y}_i$). Instead, we will admit alternative distributions obtained by shifting the $\mathcal{D}_N$'s probability mass off "corrupted" tuples $(1, x_i, y_i)_{i \in C}$ to possibly $(0, x_i, y_i)$, $(1, x_i, \tilde{y}_i)$, or even some other tuple $(1, x_j, \tilde{y}_j)$ with $j \notin C$ for example - equivalently, eliminating, correcting, or replacing the sample, respectively. In order to admit such favorable corrections to $\mathcal{D}_N$, we can consider the optimistic [19, 30] data-driven problem

$$\min_{\theta} \left(v_N(\theta) := \min_{\tilde{\mathcal{D}} : W_1(\mathcal{D}_N, \tilde{\mathcal{D}}) \leq \epsilon} \mathbb{E}_{\tilde{\mathcal{D}}} \left[\ell(w, z; \theta) \right] \right), \tag{5}$$

in which for each parameter tuning θ, $v_N(\theta)$ measures the expected auxiliary loss with respect to the most favorable distribution within an ϵ - prescribed W_1 (1- Wasserstein) distance of $\mathcal{D}_N$. It turns out that a budgeted deviation of the weights alone (and not the feature-label pairs) can approximate (up to an error diminishing in N) $v_N(\theta)$. More precisely, we derive the following approximation along similar lines to [30].

Proposition 3.2. *Let $\epsilon > 0$, and suppose for any θ, $\max_{(x,y)\in\mathcal{Z}} |J(\theta; x, y)| < \infty$. Then there exists $\kappa \geq 0$ such that for any θ, the following problem*

$$v_N^{MIX}(\theta) := \min_{u_1,\ldots,u_N} \sum_{i=1}^{N} (\frac{1}{N} + u_i) \cdot J(\theta; x_i, y_i) + \gamma_\theta \sum_{i=1}^{N} |u_i|$$
$$\text{s.t. } u_i + \frac{1}{N} \geq 0 \;\; i = 1, \ldots, N$$

satisfies $v_N(\theta) + \frac{\kappa}{N} \geq v_N^{MIX}(\theta) \geq v_N(\theta)$.

In particular, $-\gamma_\theta \leq \min_i J(\theta; x_i, y_i)$, and $\{i : J(\theta; x_i, y_i) > \gamma_\theta\}$ are all down-weighted to zero, i.e., $u_i^* = -\frac{1}{N}$ for any u^* solving $v_N^{MIX}(\theta)$.

In summary, while the optimistic Wasserstein formulation would permit correction to $\mathcal{D}_N$ with a combination of reweighting and/or feature-label revision, the above indicates that a process focused on reweighting alone could accomplish a reasonable approximation; further, upon comparison to (3), we see that RRM is a constrained version of this approximating problem, that is,

$$v(\theta) \geq v_N^{MIX}(\theta) \geq v_N(\theta).$$

Hence, in some sense, we can confirm that RRM is an optimistic methodology but that it is less optimistic than the data-driven Wasserstein approach.

3.5 RRM Algorithm

Towards solving problem (3) in the two decisions θ and u, we proceed iteratively with a block-coordinate descent heuristic outlined in Algorithm 1, whereby we update the two separately in cyclical fashion. In other words, we update θ while holding u fixed, and we update u whilst holding θ fixed. The update of θ is an SGD step on a batch of $s-$ many samples. The update of u reduces to a linear program. In light of the discussion in 3.4, we also outline an Adversarial Rockafellian Relaxation method (A-RRM), an execution of RRM that includes a perturbation (parameterized by $\epsilon \geq 0$) to the feature x of a sample (x, y), for the purposes of adversarial training.

Algorithm 1 (Adversarial) Rockafellian Relaxation Algorithm (A-RRM/RRM)

Require: Perturbation Multiplier $\epsilon \in [0, 1]$, Number of epochs σ, Batch size $s \geq 1$, learning rate $\eta > 0$, regularization parameter $\gamma > 0$, reweighting step $\mu \in (0, 1)$.
$u \leftarrow 0 \in \mathbb{R}^N$
repeat
  for $e = 1, \ldots, \sigma$ **do**
    for $b = 1, \ldots, \lceil \frac{N}{s} \rceil$ **do**
      $\{(x_i^b, y_i^b)\}_{i=1}^s \leftarrow$ Draw Batch of size s from $\{(x_i, y_i)\}_{i=1}^N$
      for i = 1, . . . , s **do**
        $x_i^b \leftarrow x_i^b + \epsilon \cdot sign\left(\nabla_x J(\theta; (x_i^b, y_i^b))\right)$
      end for
      $\theta \leftarrow \theta - \eta \sum_{i=1}^s \left(\frac{1}{N} + u_i\right) \cdot \nabla_\theta J(\theta; (x_i^b, y_i^b))$
    end for
  end for
  $u^* \leftarrow \min_{u\in U} \sum_{i=1}^N \left(\frac{1}{N} + u_i\right) \cdot J(\theta; x_i, y_i) + \gamma \|u\|_1$
  $u \leftarrow \mu u^* + (1 - \mu)u$
until Desired Validation Accuracy or Loss

The stepsize parameters μ, η and the regularization parameter γ are hyper-parameters that may be tuned, or guided by the general discussions above in Section 3.3.

The RRM algorithm, in which $\epsilon = 0$, is meant for contexts in which only label corruption and no feature corruption occurs. The A-RRM algorithm, for which $\epsilon > 0$, is intended for contexts in which both label and feature corruption is anticipated.

4 Datasets

We select several datasets to evaluate RRM. In some cases, the selected dataset is nearly pristine. In these cases we perturb the dataset to achieve various types and levels of corruption. Other datasets consist of weakly labeled examples, which we maintain unaltered. The varied data domains and regimes of corruption enable a robust evaluation of RRM.

MNIST [17]: A multi-class classification dataset consisting of 70000 images of digits zero through nine. 60000 digits are set aside for training and 10000 for testing. 0%, 5%, 10%, 20%, and 30% of the training labels are swapped for different, randomly selected digits. The test set labels are unmodified.

Toxic Comments [6]: A multi-label classification problem from JIGSAW that consists of Wikipedia comments labeled by humans for toxic behavior. Comments can be any number (including zero) of six categories: toxic, severe toxic, obscene, threat, insult, and identity hate. We convert this into a binary classification problem by treating the label as either none of the six categories or at least one of the six categories. This dataset is a public dataset used as part of the Kaggle Toxic Comment Classification Challenge.

IMDb [18]: A binary classification dataset consisting of 50000 movie reviews each assigned a positive or negative sentiment label. 25000 reviews are selected randomly for training and the remaining are used for testing. 25%, 30%, 40%, and 45% of the labels of the training reviews are randomly selected and swapped from positive sentiment to negative sentiment, and vice versa, to achieve four training datasets of desired levels of label corruption. The test set labels are unmodified.

Tissue Necrosis: A binary classification dataset consisting of 7874 256x256-pixel hematoxylin and eosin (H&E) stained RGB images derived from [2]. The training dataset consists of 3156 images labeled non-necrotic, as well as 3156 images labeled necrotic. The training images labeled non-necrotic contain no necrosis. However, only 25% of the images labeled necrotic contain necrotic tissue. This type of label error can be expected in cases of weakly-labeled Whole Slide Imagery (WSI). Here, an expert pathologist will provide a slide-level label for a potentially massive slide consisting of gigapixels, but they lack time or resources to provide granular, segmentation-level annotations of the location of the pathology in question. Also, the diseased tissue often occupies a small portion of the WSI, with the remainder consisting of normal tissue. When the gigapixel-sized WSI is subsequently divided into sub-images of manageable size for typical machine-learning workflows, many of the sub-images will contain no disease, but will be assigned the "weak" label chosen by the expert for the WSI. The test dataset consists of 718 necrosis and 781 non-necrosis 256x256-pixel H&E images, which were also derived from [2]. For both the training and test images, [2] provide segmentation-level necrosis annotations, so we are able to ensure a pristine test set, and, in the case of the training set, we were able to identify the corrupted images for the purpose of algorithm evaluation.

5 Architectures

We do not strive to develop a novel NN architectures capable of defeating current state-of-the-art (SOA) performance in each data domain. Nor do we focus on developing *robust architectures* as described in [28]. Rather, we select a reasonable NN architecture and measure model performance with and without the application of RRM. This approach enables us to demonstrate the general superiority of RRM under varied data domains and NN architectures. We discuss the underlying NN architectures that we employ in this section.

MNIST: The MNIST dataset has been studied extensively and harnessed to investigate novel machine-learning methods, including CNNs [4]. We adopt a basic CNN architecture with a few convolutional layers. The first layer has a depth of 32, and the next two layers have a depth of 64. Each convolutional layer employs a kernel of size three and the ReLU activation function followed by a max-pooling layer employing a kernal of size 2. The last convolutional layer is connected to a classification head consisting of a 100-unit dense layer with ReLU activation, followed by a 10-unit dense layer with softmax activation. In total, there are 159254 trainable parameters. Categorical cross-entropy is employed for the loss function.

Toxic Comments: We use a simple model with only a single convolutional layer. A pretrained embedding from FastText is first used to map the comments into a 300 dimension embedding space, followed by a single convolutional layer with a kernel size of two with a ReLU activation layer followed by a max-pooling layer. We then apply a 36-unit dense layer, followed by a 6 unit dense layer with sigmoid activation. Binary cross-entropy is used for the loss function.

IMDb: Transformer architectures have achieved SOA performance on the IMDb dataset sentiment analysis task [7, 32]. As such, we a adopt a reasonable transformer architecture to assess RRM. We utilize the DistilBERT [25] architecture with low-rank adaptation (LoRA) [13] for large language models, which reduces the number of trainable weights from 67584004 to 628994. In this manner, we reduce the computational burden, while maintaining excellent sentiment analysis performance. Binary cross-entropy is employed for the loss function.

Tissue Necrosis: Consistent with the computational histopathology literature [21], we employ a convolutional neural network (CNN) architecture for this classification task. In particular, a ResNet-50 architecture with pre-trained ImageNet weights is harnessed. The classification head is removed and replaced with a dense layer of 512 units and ReLU activation function, followed by an output layer with a single unit using a sigmoid activation function. All weights, with the exception of the new classification head are frozen, resulting in 1050114 trainable parameters out of 24637826. Binary cross-entropy is employed for the loss function.

6 Experiments and Results

In this work, we have discussed errors/perturbations/corruption to features and labels. We now perform experiments to see how RRM performs under one or the other, or both. The MNIST experiments are performed under a setting of both adversarial perturbation, as well as label corruption. The Toxic Comments experiments are performed under settings of label corruption only. All experiments are performed using a combination of GPU resources, both cloud-base, as well as access to an on-premise high-performance computing (HPC) facility. We refer the reader to the Appendix (Sections 6.3 and 6.4) for the experiments on IMDb and Tissue Necrosis.

6.1 MNIST

Twenty percent of the training data is set aside for validation purposes. Using Tensorflow 2.10 [1], 50 iterations of RRM are executed with $\sigma = 10$ epochs per iteration for a total of 500 epochs for a given hyperparameter setting. For RRM, the hyperparameter settings of μ and γ at 0.5 and 2.0, respectively, are based on a search to optimize validation set accuracy. For contrast, we perform a comparable 500 epochs using ERM. Both ERM and RRM employ stochastic gradient descent (SGD) with a learning rate (η) of 0.1. Each time a batch is drawn, each training image is perturbed using the Fast Gradient Sign Method (FGSM) [11] adversarial attack: $adv_x = x + \epsilon \cdot sign(\nabla_x J(\theta, x, y))$, where adv_x is the resulting perturbed image, x is the original image, y is the image label, ϵ is a multiplier controlling the magnitude of the image perturbation, θ are the model parameters, and J is the loss. An $\epsilon = 1.0$ is used for all training image perturbations.

For each of the 0%, 5%, 10%, 20%, and 30% training label corruption levels, we compare adversarial training (AT) and adversarial RRM (A-RRM) performance under varios regimes of test set perturbation ($\epsilon_{test} \in 0.0, 0.1, 0.25, 0.5, 1.0$). In Table 1 we show the test set accuracy achieved when validation set accuracy peaks. We can see that training with an $\epsilon_{train} = 1.0$ and testing with lower ϵ_{test} levels of $0.00, 0.10$, and 0.25, results in a drastic degradation in accuracy for AT for corruption levels greater than 0%. This performance collapse is not observed when using A-RRM. Given that it may be difficult to anticipate the adversarial regime in production environments, A-RRM seems to confer a greater benefit than AT.

We examine the u_i-value associated with each training observation, i, from iteration-to-iteration of the heuristic algorithm. Table 2 summarizes the progression of the u_i-vector across its 49 updates for the dataset corruption level of 20%. Column "1. iteration" shows the distribution of u_i-values following the first u-optimization for both the 9600 corrupted training observations and the 38400 clean training observations. Initially, all u_i-values are approximately equal to 0.0. It is once again observed that, over the course of iterations, the u_i-values noticeably change. In column "10. iteration" it can be seen that a significant number of the u_i-values of the corrupted training observations

Table 1: Test accuracy (%) for AT and A-RRM on MNIST under different levels of corruption C and test-set adversarial perturbation ϵ_{test}.

	Percentage Corrupted Training Data									
C	0%		5%		10%		20%		30%	
ϵ_{test}	AT	A-RRM	AT	A-RRM	AT	A-RRM	AT	A-RRM	AT	A-RRM
0.00	**97**	96	63	**95**	57	**97**	58	**96**	26	**86**
0.10	**95**	93	64	**92**	71	**94**	61	**93**	20	**82**
0.25	**93**	90	83	**91**	88	**92**	84	**90**	74	**81**
50	**91**	88	**94**	91	**94**	90	**90**	88	**97**	80
1.00	**86**	83	**95**	90	**94**	86	**88**	83	**98**	77

achieve negative values, while a large majority of the u_i-values for the clean training observations remain close to 0.0. Finally, column "49. iteration" displays the final u_i-values. 9286 out of 9600 of the corrupted training observations have achieved a $u_i \in (-2.08, -1.56] \cdot 10 - 5$. This means these training observations are removed, or nearly-so, from consideration because this value cancels the nominal probability $1/N$ = 2.08 · 10-5. It is observed that a large majority (35246/38400) clean training observations remain with their nominal probability. This helps explain the performance benefit of A-RRM over AT. A-RRM "removes" the corrupted data points in-situ, whereas AT does not. It appears that under adversarial training regimes with corrupted training data, it is essential to identify and "remove" the corrupted examples, especially if the level adversarial perturbation encountered in the test set is unknown, or possibly lower than the level of adversarial perturbation applied to the training set.

Table 2: Evolution of u-vector across 9600 corrupted data points and 38400 clean data points. Note that 1/(9600 + 38400) = 2.08 · 10-5.

	1. iteration		10. iteration		49. iteration	
u_i value	corrupted data points	clean data points	corrupted data points	clean data points	corrupted data points	clean data points
$\gg 0$	0	1	0	4.	0	25
≈ 0	8844	38385	2058	37524	91	35246
(-0.52, 0.00) · 10-5	0	0	7	36	146	1655
(-1.04, -0.52] · 10-5	0	0	41	45	43	155
(-1.56, -1.04] · 10-5	756	14	415	174	34	168
(-2.08, -1.56] · 10-5	0	0	7079	617	9286	1151

6.2 Toxic Comment

We use the Toxic Comment dataset to test the efficacy of RRM on low prevalence text data. The positive (toxic) comments consist of only 3% of the data and we corrupt anywhere from 1% to 20% of the labels. There are a total of 148,000 samples, and we set aside 80% for training and 20% for test. $\sigma = 2$ with 3 iterations of the heuristic algorithm results in a total of 6 epochs, and ERM is run for a total of 6 epochs to make the results comparable. Since the data is highly imbalanced, we look at the area under the curve of the precision/recall curve to assess the performance of the models. Unsurprisingly, as the noise increase, the model performance decreases. We note that RRM outperforms ERM across all noise levels tested, though as the noise increase, the gap between RRM and ERM decreases.

Table 3: Comparison of training and test area under the precision/recall curve for ERM and RRM at noise levels ranging from 1% to 20%.

Method	Percentage Corrupted Training Data					
	1%	5%	7%	10%	15%	20%
ERM (train)	0.2904	0.2006	0.1589	0.1302	0.1073	0.0920
RRM (train)	0.6875	0.4458	0.3805	0.3087	0.2438	0.1966
ERM (test)	0.5861	0.3970	0.3246	0.2550	0.2013	0.1717
RRM (test)	**0.6705**	**0.4338**	**0.3619**	**0.2824**	**0.2208**	**0.1861**

6.3 IMDb

Twenty percent of the training data is set aside for validation purposes. Using Pytorch 2.1.0 [3], 30 iterations of RRM are executed, with $\sigma = 10$ epochs per iteration for a total of 300 epochs for a given hyperparameter setting. For RRM, the hyperparameter settings of μ and γ at 0.5 and 0.4, respectively, are based on a search to optimize validation set accuracy. For contrast, we perform a comparable 300 epochs using ERM. Both ERM and RRM employ stochastic gradient descent (SGD) with a learning rate (η) of 0.001. In Table 4 we record both the test set accuracy achieved when validation set accuracy peaks, as well as the maximum test set accuracy. At these high levels of corruption RRM consistently achieves a better maximum test set accuracy.

Table 4: Test accuracy (%) for ERM and RRM on IMDb under different levels of corruption. Test set accuracy at peak validation accuracy and maximum test set accuracy are recorded.

Method	Percentage Corrupted Training Data			
	25%	30%	40%	45%
ERM	*90.2*, 90.2	89.5, 89.6	86.4, 86.6	*80.7*, 81.1
RRM	90.1, **90.4**	*90.2*, **90.4**	*88.4*, **88.7**	76.9, **82.6**

6.4 Tissue Necrosis

Twenty percent of the training data is set aside for validation purposes, including hyperparameter selection. 60 iterations of RRM are executed, with $\sigma = 10$ epochs per iteration, for a total of 600 epochs for a given hyperparameter setting. For RRM, the hyperparameter settings of μ and γ at 0.5 and 0.016, respectively, are based on a search to optimize validation set accuracy. For contrast, we perform a comparable 600 epochs using ERM. Both ERM and RRM employ stochastic gradient descent (SGD) with a learning rate (η) of 5.0 and 1.0, respectively. RRM achieves a test set accuracy at peak validation accuracy of **74.6**, and a maximum test set accuracy **77.2**, whereas ERM achieves 71.7 and 73.2, respectively. RRM appears to confer a performance benefit under this regime of weakly labeled data.

7 Conclusion

In this study, we demonstrate the robustness of the A-RRM algorithm in a variety of data domains, data corruption schemes, model architectures and machine learning applications. In the MNIST example we show that conducting training in preparation for deployment environments with varied levels of adversarial attacks, one can benefit from implementation of the A-RRM algorithm. This can lead to a model more robust across levels of both feature perturbation and high levels of label corruption. We also demonstrate the mechanism by which A-RRM operates and confers superior results: by automatically identifying and removing the corrupted training observations at training time execution.

The Toxic Comment example presents another challenging classification problem, characterized by a low prevalence target class amidst label noise. Our experiments demonstrate that as the amount of label noise increases, standard methods become increasingly ineffective. However, RRM remains reasonably robust under varying degrees of label corruption. Therefore, RRM could be a valuable addition to the set of tools being developed to enhance the robustness of AI-based decision engines.

In the IMDb example we demonstrate that RRM can confer benefits to the sentiment analysis classification task using pre-trained large models under conditions of high label corruption. The success of fine-tuning in LLMs depends, in large part, on access to high quality training examples. We have shown that RRM can mitigate this need by allowing effective training in scenarios of high training data corruption. As such, resource allocation dedicated to dataset curation may be lessened by the usage of RRM.

In the Tissue Necrosis example, we demonstrate that RRM also confers accuracy benefits to the necrosis identification task provided weakly labeled WSIs. Again, RRM can mitigate the need for expert-curated, detailed pathology annotations, which are costly and time-consuming to generate.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Mohamed Amgad, Habiba Elfandy, Hagar Hussein, Lamees A Atteya, Mai A T Elsebaie, Lamia S Abo Elnasr, Rokia A Sakr, Hazem S E Salem, Ahmed F Ismail, Anas M Saad, Joumana Ahmed, Maha A T Elsebaie, Mustafijur Rahman, Inas A Ruhban, Nada M Elgazar, Yahya Alagha, Mohamed H Osman, Ahmed M Alhusseiny, Mariam M Khalaf, Abo-Alela F Younes, Ali Abdulkarim, Duaa M Younes, Ahmed M Gadallah, Ahmad M Elkashash, Salma Y Fala, Basma M Zaki, Jonathan Beezley, Deepak R Chittajallu, David Manthey, David A Gutman, and Lee A D Cooper. Structured crowdsourcing enables convolutional segmentation of histology images. *Bioinformatics*, 35(18):3461–3467, 02 2019.

[3] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, 4 2024.

[4] Alejandro Baldominos, Yago Saez, and Pedro Isasi. A survey of handwritten character recognition with mnist and emnist. *Applied Sciences*, 9(15), 2019.

[5] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples, 2018.

[6] cjadams, Jeffrey Sorensen, Julia Elliott, Lucas Dixon, Mark McDonald, nithum, and Will Cukierski. Toxic comment classification challenge, 2017.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] Benoit Frenay and Michel Verleysen. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, 2014.

[9] Rui Gao and Anton Kleywegt. Distributionally robust stochastic optimization with wasserstein distance. *Mathematics of Operations Research*, 48(2):603–655, 2023.

[10] Aritra Ghosh, Himanshu Kumar, and P. S. Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 1919–1925. AAAI Press, 2017.

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations*, 2015.

[12] Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. *Data cleansing for models trained with SGD*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[13] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

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

[15] Jonathan Krause, Benjamin Sapp, Andrew Howard, Howard Zhou, Alexander Toshev, Tom Duerig, James Philbin, and Li Fei-Fei. The unreasonable effectiveness of noisy data for fine-grained recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 301–320. Springer, 2016.

[16] Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

[17] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[18] 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*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[19] Viet Anh Nguyen, Soroosh Shafieezadeh Abadeh, Man-Chung Yue, Daniel Kuhn, and Wolfram Wiesemann. Optimistic distributionally robust optimization for nonparametric likelihood approximation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[20] Guy Nir, Soheil Hor, Davood Karimi, Ladan Fazli, Brian F. Skinnider, Peyman Tavassoli, Dmitry Turbin, Carlos F. Villamil, Gang Wang, R. Storey Wilson, Kenneth A. Iczkowski, M. Scott Lucia, Peter C. Black, Purang Abolmaesumi, S. Larry Goldenberg, and Septimiu E. Salcudean. Automatic grading of prostate cancer in digitized histopathology images: Learning from multiple experts. *Medical Image Analysis*, 50:167–180, 2018.

[21] Dominika Petríková and Ivan Cimrák. Survey of recent deep neural networks with strong annotated supervision in histopathology. *Computation*, 11(4), 2023.

[22] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, 2018.

[23] Johannes O. Royset, Louis L. Chen, and Eric Eckstrand. Rockafellian relaxation and stochastic optimization under perturbations. *Mathematics of Operations Research (to appear)*, 2023.

[24] Johannes O. Royset and Roger J-B Wets. *An Optimization Primer*. Springer, 2021.

[25] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[26] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

[27] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In Mirella Lapata and Hwee Tou Ng, editors, *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 254–263, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.

[28] Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *CoRR*, abs/2007.08199, 2020.

[29] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[30] Matthew Staib and Stefanie Jegelka. Distributionally robust deep learning as a generalization of adversarial training. In *NIPS workshop on Machine Learning and Computer Security*, volume 3, page 4, 2017.

[31] V. Vapnik. Principles of risk minimization for learning theory. In *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91, page 831–838, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

[32] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019.

[33] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

[34] HaiYang Zhang, XiMing Xing, and Liang Liu. Dualgraph: A graph-based method for reasoning about label noise. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9649–9658, 2021.

[35] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 8792–8802, Red Hook, NY, USA, 2018. Curran Associates Inc.

[36] Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise: A quantitative study. *Artificial intelligence review*, 22:177–210, 2004.

A Appendix / supplemental material

A.1 Section 3 Proofs

Theorem 3.1. *Let $\gamma > 0$ and $c = (c_1, \ldots, c_N) \in \mathbb{R}^N$, with $c_{min} := \min_i c_i$, and $c_{max} := \max_i c_i$. Write $I_{min} := \{i : c_i = c_{min}\}$, $I_{big} := \{i : c_i = c_{min} + 2\gamma\}$, and for any $S_1 \subseteq I_{min}, S_2 \subseteq I_{big}$, define the polytope* $U^*_{S_1,S_2} := \left\{ u^* \in U : \begin{array}{c} u^*_i \geq 0 \ \ \forall i : c_i = c_{min} \\ u^*_i = 0 \ \ \forall i : c_i \in (c_{min}, c_{min} + 2\gamma) \\ u^*_i = -\frac{1}{N} \ \ \forall i \in I_{big} \setminus S_2 \\ u^*_i = -\frac{1}{N} \ \ \forall i : c_i > c_{min} + 2\gamma \\ u^*_i = 0 \ \ \forall i \in S_1 \cup S_2 \end{array} \right\}$. *Then*

$$conv\left(\cup_{S_1,S_2} U^*_{S_1,S_2}\right) = \arg\min_{u \in U} \sum_{i=1}^{N} (\frac{1}{N} + u_i) \cdot c_i + \gamma \|u\|_1. \tag{4}$$

Proof. For any set C, let $\iota_C(x) = 0$ and $\iota_C(x) = \infty$ otherwise. We recognize that $u^\star$ is a solution of the minimization problem if and only if it is a minimizer of the function h given by

$$h(u) = \sum_{i=1}^{N} \Big(c_i/N + u_i c_i + \gamma |u_i| + \iota_{[0,\infty)}(1/N + u_i) \Big) + \iota_{\{0\}}\Big(\sum_{i=1}^{N} u_i \Big)$$

Thus, because $h(u) > -\infty$ for all $u \in \mathbb{R}^N$ and h is convex, $u^\star$ is a solution of the minimization problem if and only if $0 \in \partial h(u^\star)$ by Theorem 2.19 in [24]. We proceed by characterizing ∂h.

Consider the univariate function h_i given by

$$h_i(u_i) = c_i/N + u_i c_i + \gamma |u_i| + \iota_{[0,\infty)}(1/N + u_i).$$

For $u_i \geq -1/N$, the Moreau-Rockafellar sum rule (see, e.g, [24, Theorem 2.26]) gives that

$$\partial h_i(u_i) = c_i + \begin{cases} \{\gamma\} & \text{if } u_i > 0 \\ [-\gamma, \gamma] & \text{if } u_i = 0 \\ \{-\gamma\} & \text{if } -1/N < u_i < 0 \\ (-\infty, -\gamma] & \text{if } u_i = -1/N. \end{cases}$$

For $u = (u_1, \ldots, u_N) \in [-1/N, \infty)^N$, we obtain by Proposition 4.63 in [24] that

$$\partial\Big(\sum_{i=1}^N h_i\Big)(u) = \partial h_1(u_1) \times \cdots \times \partial h_N(u_N).$$

Let h_0 be the function given by $h_0(u) = \iota_{\{0\}}(\sum_{i=1}^N u_i)$. Again invoking the Moreau-Rockafellar sum rule while recognizing that the interior of the domain of $\sum_{i=1}^N h_i$ intersects with the domain of h_0, we obtain

$$\partial h(u) = \partial\Big(\sum_{i=1}^N h_i\Big)(u) + \partial h_0(u) = \partial h_1(u_1) \times \cdots \times \partial h_N(u_N) + \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \mathbb{R}$$

for any $u = (u_1, \ldots, u_N)$ with $u_i \geq -1/N$, $i = 1, \ldots, N$, and $\sum_{i=1}^N u_i = 0$. Hence, $u^* \in U$ is optimal if and only if for some $\lambda \in \mathbb{R}$,

$$\lambda \in \begin{cases} \{c_i + \gamma\} & \text{if } u_i^\star > 0 \\ [c_i - \gamma, c_i + \gamma] & \text{if } u_i^\star = 0 \\ \{c_i - \gamma\} & \text{if } u_i^\star \in (-1/N, 0) \\ (-\infty, c_i - \gamma] & \text{if } u_i^\star = -1/N. \end{cases}$$

It follows that $\lambda = c_{min} + \gamma$ can accompany any optimal u^* in satisfying the above; hence, the result follows.

□

Proposition A.1. *Let $\epsilon > 0$, and suppose for any θ, $\max_{(x,y)\in\mathcal{Z}} |J(\theta; x, y)| < \infty$. Then there exists $\kappa \geq 0$ such that for any θ, the following problem*

$$v_N^{MIX}(\theta) := \min_{u_1,\ldots,u_N} \sum_{i=1}^N (\frac{1}{N} + u_i) \cdot J(\theta; x_i, y_i) + \gamma_\theta \sum_{i=1}^N |u_i|$$
$$\text{s.t. } u_i + \frac{1}{N} \geq 0 \;\; i = 1, \ldots, N$$

satisfies $v_N(\theta) + \frac{\kappa}{N} \geq v_N^{MIX}(\theta) \geq v_N(\theta)$.

In particular, $-\gamma_\theta \leq \min_i J(\theta; x_i, y_i)$, and $\{i : J(\theta; x_i, y_i) > \gamma_\theta\}$ are all down-weighted to zero, i.e., $u_i^* = -\frac{1}{N}$ for any u^* solving $v_N^{MIX}(\theta)$.

Proof. Fix θ. Then for any $z = (x, y) \in \mathcal{Z}$, the function $\ell(\cdot, z, \theta)$ is linear, and hence Lipschitz with constant $\ell(1, z, \theta) = J(\theta; x, y) \leq \max_{(x,y)\in\mathcal{Z}} |J(\theta; x, y)| < \infty$.

By Lemma 3.1 of [30] and/or Corollary 2 of [9],

$$v_N^{MIX}(\theta) := \min_{\tilde{w}^1,\ldots,\tilde{w}^N \geq 0} \frac{1}{N} \sum_{i=1}^N \ell(\tilde{w}^i, z^i; \theta)$$
$$\text{s.t. } \frac{1}{N} \sum_{i=1}^N |\tilde{w}^i - w^i| \leq \epsilon$$

provides the stated approximation of $v(\theta)$.

Upon introducing the change of variable $u_i = \frac{\tilde{w}^i}{N} - \frac{1}{N}$, and applying a Lagrange multiplier γ_θ to the $\epsilon-$ budget constraint (any convex dual optimal multiplier), we recover

$$\min_{u_1,\ldots,u_N} \sum_{i=1}^{N} \ell(u_i + \frac{1}{N}, z^i; \theta) + \gamma_\theta \sum_{i=1}^{N} |u_i|$$
$$\text{s.t. } u_i + \frac{1}{N} \geq 0 \;\; i = 1, \ldots, N$$

□

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Sections 6.1, 6.2, 6.3, 6.4

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper has focused more on label corruption, rather than feature perturbation settings.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Section 3

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sections 3.2, 4, 5, 6

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The datasets are open-source, and the code will be made available pending conference review of this work

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Sections 4, 5, 6

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Error bars are not reported because it would be too computationally expensive.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See section 6

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper conforms with the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The work in the paper is foundational research and is not tied to a particular application or deployment.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: No models are released as part of this work, and the datasets are publicly available.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Citations for publicly available datasets and code are provided.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: No new assets are introduced in the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No crowdsourcing experiments or research with human subjects was conducted.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.