~~ve~~ data มี mis labeled
ง่าย มากในปัจจุบัน ขึ้น Amazon M-turk
(Crowdsourcing)

noisy ~~ex~~ example

ประเด็นที่เกิดขึ้นใน NN

1. NN จะ memorizing noisy example → worse generalization
2. NN are pretty good at memorizing

วิธีแก้ก่อนหน้านี้

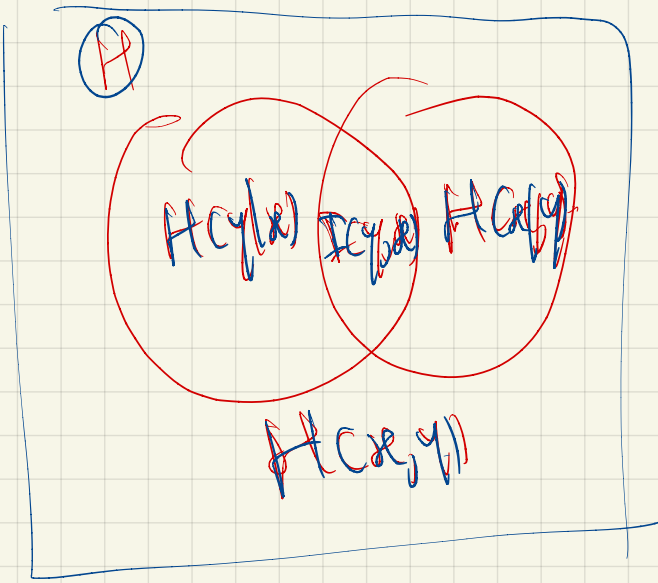
- → Design Robust loss fn (
- → loss correction approache (confusion matrix)
- → Detecting mislabeled example (Mentor-Net, small loss ~~trick~~ trick
  vari
- → Co-training two Network (ex. teacher-student)
- → Standard regularization techniques.

What do we do

1. Understand the source of memorization
2. Measure label memorization and Control it

$$H(x) = -\int p(x) \log p(x)\, dx$$

$$\therefore I(x,y) = H(x) - H(x|y)$$
$$= H(y) - H(y|x)$$



⇓

ค่าที่ใช้วัดความแตกต่าง KL-divergence

Setup $x = \{x^{(i)}\}_{i=1}^{n}$ label $y = \{y^{(i)}\}_{i=1}^{n}$; $D = (x,y)$

$$H(y|x) > 0$$

Classifier $f(y|x,w)$ → we train use algo $A(w|D)$

$$H_{p,f}(y|x,w) = E_{D=(x,y)}\left[E \right.$$

popular Algo

random variable — ค่าที่ w ที่ทำให้ delta ที่สุด

$$A_{ERM}(w|D) = \delta(w^*) \quad ; \quad w^* \in \text{argmin}_w \sum_{i=1}^{N} -\log f(y^{(i)}|x^{(i)},w)$$

(ERM ⇒ Empirical risk minimization)

$H_{p,f}(y|x,w) = \underbrace{H(y|x)}_{\text{intrinsic error}}$

$+ \underbrace{E_{x,w}\, D_{KL}\left[p(y|x) \,\|\, f(y|x,w)\right]}_{\text{how good is the classifier}}$

$\underbrace{- I(w;y|x)}_{\text{memorization}}$ $\Rightarrow$ $I \uparrow$ ทำให้ $H \downarrow$ → model ไม่ได้เรียนรู้ pattern แต่จำมากขึ้น

∴ Loss fn ที่ดี ควรมี $I(w;y|x)$ ต่ำๆ เพื่อลด memorization

$$\boxed{\text{Loss fn} \triangleq H_{p,f}(y|x,w) + I(w;y|x)} \Rightarrow \text{bound obj.}$$

r => the expected rate of mistake

$$r \geq \frac{H(y^{(1)}|x^{(1)}) - I(w;y|x)/n - H(r)}{\log(|y|-1)}$$

$|y| \Rightarrow$ ตัวแปร class

ใช้ graph ดู

$I(w;D) = I(w;x) + I(w;y|x)$ minimize memorization

$$E[\text{gen gap}] \leq \sqrt{\frac{2C}{n} I(w;D)}$$

train vs test performance

Derivation sketch

① From information in weight to information in gradient

$$I(w; y \mid x) \le I(g_{1:T}; y \mid x) = \sum_{t=1}^{T} I(g_t; y \mid x, g_{<t})$$

chain rule

② upper bound information in gradient

$$I(g_t; y \mid x, g_{<t}) \le E_{g(t)}[q_\phi(g_t \mid x, g_{<t})] + C$$

③ use predicted gradients to train the classifier

network $z \to$ $\to$ predict gradients (q_ϕ)

④ penalize the L2 Norm of predicting gradients

Improving Generalization by Controlling Label-Noise Information in Neural Network Weights

Hrayr Harutyunyan [1] **Kyle Reing** [1] **Greg Ver Steeg** [1] **Aram Galstyan** [1]

Abstract

In the presence of noisy or incorrect labels, neural networks have the undesirable tendency to memorize information about the noise. Standard regularization techniques such as dropout, weight decay or data augmentation sometimes help, but do not prevent this behavior. If one considers neural network weights as random variables that depend on the data and stochasticity of training, the amount of memorized information can be quantified with the Shannon mutual information between weights and the vector of all training labels given inputs, $I(w; \mathbf{y} \mid \mathbf{x})$. We show that for any training algorithm, low values of this term correspond to reduction in memorization of label-noise and better generalization bounds. To obtain these low values, we propose training algorithms that employ an auxiliary network that predicts gradients in the final layers of a classifier without accessing labels. We illustrate the effectiveness of our approach on versions of MNIST, CIFAR-10, and CIFAR-100 corrupted with various noise models, and on a large-scale dataset Clothing1M that has noisy labels.

1. Introduction

Supervised learning with deep neural networks has shown great success in the last decade. Despite having millions of parameters, modern neural networks generalize surprisingly well. However, their training is particularly susceptible to noisy labels, as shown by Zhang et al. (2016) in their analysis of generalization error. In the presence of noisy or incorrect labels, networks start to memorize the training labels, which degrades the generalization performance (Chen et al., 2019). At the extreme, standard architectures have the capacity to achieve 100% classification accuracy on training data, even when labels are assigned at random (Zhang et al., 2016). Furthermore, standard explicit or implicit regularization techniques such as dropout, weight decay or data augmentation do not directly address nor completely prevent label memorization (Zhang et al., 2016; Arpit et al., 2017).

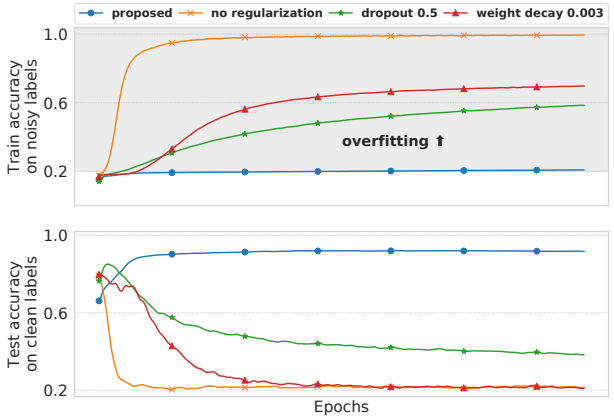


Figure 1: Neural networks tend to memorize labels when trained with noisy labels (80% noise in this case), even when dropout or weight decay are applied. Our training approach limits label-noise information in neural network weights, avoiding memorization of labels and improving generalization. Please refer to Sec. 2.1 for more details.

Poor generalization due to label memorization is a significant problem because many large, real-world datasets are imperfectly labeled. Label noise may be introduced when building datasets from unreliable sources of information or using crowd-sourcing resources like Amazon Mechanical Turk. A practical solution to the memorization problem is likely to be algorithmic as sanitizing labels in large datasets is costly and time consuming. Existing approaches for addressing the problem of label-noise and generalization performance include deriving robust loss functions (Natarajan et al., 2013; Ghosh et al., 2017; Zhang & Sabuncu, 2018; Xu et al., 2019), loss correction techniques (Sukhbaatar et al., 2014; Tong Xiao et al., 2015; Goldberger & Ben-Reuven, 2017; Patrini et al., 2017), re-weighting samples (Jiang et al., 2017; Ren et al., 2018), detecting incorrect samples and re-

[1]Information Sciences Institute, University of Southern California, Marina del Rey, CA 90292. Correspondence to: Hrayr Harutyunyan <hrayrh@isi.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

labeling them (Reed et al., 2014; Tanaka et al., 2018; Ma et al., 2018), and employing two networks that select training examples for each other (Han et al., 2018; Yu et al., 2019). We propose an information-theoretic approach that directly addresses the root of the problem. If a classifier is able to correctly predict a training label that is actually random, it must have somehow stored information about this label in the parameters of the model. To quantify this information, Achille & Soatto (2018) consider weights as a random variable, w, that depends on stochasticity in training data and parameter initialization. The entire training dataset is considered a random variable consisting of a vector of inputs, $\mathbf{x}$, and a vector of labels for each input, $\mathbf{y}$. The amount of label memorization is then given by the Shannon mutual information between weights and labels conditioned on inputs, $I(w; \mathbf{y} \mid \mathbf{x})$. Achille & Soatto (2018) show that this term appears in a decomposition of the commonly used expected cross-entropy loss, along with three other individually meaningful terms. Surprisingly, cross-entropy rewards large values of $I(w; \mathbf{y} \mid \mathbf{x})$, which may promote memorization if labels contain information beyond what can be inferred from $\mathbf{x}$. Such a result highlights that in addition to the network's representational capabilities, the loss function – or more generally, the learning algorithm – plays an important role in memorization. To this end, we wish to study the utility of limiting $I(w; \mathbf{y} \mid \mathbf{x})$, and how it can be used to modify training algorithms to reduce memorization.

Our main contributions towards this goal are as follows: 1) We show that low values of $I(w; \mathbf{y} \mid \mathbf{x})$ correspond to reduction in memorization of label-noise, and lead to better generalization gap bounds. 2) We propose training methods that control memorization by regularizing label-noise information in weights. When the training algorithm is a variant of stochastic gradient descent, one can achieve this by controlling label-noise information in gradients. A promising way of doing this is through an additional network that tries to predict the classifier gradients without using label information. We experiment with two training procedures that incorporate gradient prediction in different ways: one which uses the auxiliary network to penalize the classifier, and another which uses predicted gradients to train it. In both approaches, we employ a regularization that penalizes the L2 norm of predicted gradients to control their capacity. The latter approach can be viewed as a search over training algorithms, as it implicitly looks for a loss function that balances training performance with label memorization. 3) Finally, we show that the auxiliary network can be used to detect incorrect or misleading labels. To illustrate the effectiveness of the proposed approaches, we apply them on corrupted versions of MNIST, CIFAR-10, CIFAR-100 with various label noise models, and on the Clothing1M dataset, which already contains noisy labels. We show that methods based on gradient prediction yield drastic improvements over standard training algorithms (like cross-entropy loss), and outperform competitive approaches designed for learning with noisy labels.

2. Label-Noise Information in Weights

We begin by formally introducing a measure of label-noise information in weights, and discuss its connections to memorization and generalization. Throughout the paper we use several information-theoretic quantities such as entropy: $H(X) = -\mathbb{E}\big[\log p(x)\big]$, mutual information: $I(X;Y) = H(X) + H(Y) - H(X,Y)$, Kullback–Leibler divergence: $\mathrm{KL}(p(x)||q(x)) = \mathbb{E}_{x\sim p(x)}\left[\log(p(x)/q(x))\right]$ and their conditional variants (Cover & Thomas, 2006).

Consider a setup in which a labeled dataset, $S = (\mathbf{x}, \mathbf{y})$, for data $\mathbf{x} = \{x^{(i)}\}_{i=1}^n$ and categorical labels $\mathbf{y} = \{y^{(i)}\}_{i=1}^n$, is generated from a distribution $p_\theta(x, y)$. A training algorithm for learning weights w of a fixed probabilistic classifier $f(y \mid x, w)$ can be denoted as a conditional distribution $\mathcal{A}(w \mid S)$. Given any training algorithm $\mathcal{A}$, its training performance can be measured using the expected cross-entropy:

$$H_{p,f}(\mathbf{y} \mid \mathbf{x}, w) = \mathbb{E}_S \mathbb{E}_{w|S}\left[\sum_{i=1}^{n} -\log f(y^{(i)} \mid x^{(i)}, w)\right].$$

Achille & Soatto (2018) present a decomposition of this expected cross-entropy, which reduces to the following when the data generating process is fixed:

$$\begin{aligned} H_{p,f}(\mathbf{y} \mid \mathbf{x}, w) =& H(\mathbf{y} \mid \mathbf{x}) - \overbrace{I(w; \mathbf{y} \mid \mathbf{x})}^{\text{memorizing label-noise}} \\ &+ \mathbb{E}_{\mathbf{x},w}\mathrm{KL}\big[p(\mathbf{y} \mid \mathbf{x})||f(\mathbf{y} \mid \mathbf{x}, w)\big]. \end{aligned} \tag{1}$$

The problem of minimizing this expected cross-entropy is equivalent to selecting an appropriate training algorithm. If the labels contain information beyond what can be inferred from inputs (meaning non-zero $H(\mathbf{y} \mid \mathbf{x})$), such an algorithm may do well by memorizing the labels through the second term of (1). Indeed, minimizing the empirical cross-entropy loss $\mathcal{A}^{\text{ERM}}(w \mid S) = \delta(w^*)$, where $w^* \in \arg\min_w \sum_{i=1}^n -\log f(y^{(i)} \mid x^{(i)}, w)$, does exactly that (Zhang et al., 2016).

2.1. Decreasing $I(w; \mathbf{y} \mid \mathbf{x})$ Reduces Memorization

To demonstrate that $I(w; \mathbf{y} \mid \mathbf{x})$ is directly linked to memorization, we prove that any algorithm with small $I(w; \mathbf{y} \mid \mathbf{x})$ overfits less to label-noise in the training set.

Theorem 2.1. Consider a dataset $S = (\mathbf{x}, \mathbf{y})$ of n i.i.d. samples, $\mathbf{x} = \{x^{(i)}\}_{i=1}^n$ and $\mathbf{y} = \{y^{(i)}\}_{i=1}^n$, where the domain of labels is a finite set, $\mathcal{Y}$. Let $\mathcal{A}(w \mid S)$ be any training algorithm, producing weights for a possibly stochastic classifier $f(y \mid x, w)$. Let $\widehat{y}^{(i)}$ denote the prediction of the classifier

on the i-th example and let $e^{(i)} = \mathbb{1}\{\widehat{y}^{(i)} \neq y^{(i)}\}$ be a random variable corresponding to predicting $y^{(i)}$ incorrectly. Then, the following inequality holds:

$$\mathbb{E}\left[\sum_{i=1}^{n} e^{(i)}\right] \geq \frac{H(\mathbf{y} \mid \mathbf{x}) - I(w; \mathbf{y} \mid \mathbf{x}) - \sum_{i=1}^{n} H(e^{(i)})}{\log\left(|\mathcal{Y}| - 1\right)}.$$

This result establishes a lower bound on the expected number of prediction errors on the training set, which increases as $I(w; \mathbf{y} \mid \mathbf{x})$ decreases. For example, consider a corrupted version of the MNIST dataset where each label is changed with probability 0.8 to a uniformly random incorrect label. By the above bound, every algorithm for which $I(w; \mathbf{y} \mid \mathbf{x}) = 0$ will make at least 80% prediction errors on the training set in expectation. In contrast, if the weights retain 1 bit of label-noise information per example, the classifier will make at least 40.5% errors in expectation. The proof of Thm. 2.1 uses Fano's inequality and is presented in the supplementary material (Sec. A.1). Below we discuss the dependence of error probability on $I(w; \mathbf{y} \mid \mathbf{x})$.

Remark 1. If we let $k = |\mathcal{Y}|$ and $r = \frac{1}{n}\mathbb{E}\left[\sum_{i=1}^{n} e^{(i)}\right]$ denote the expected training error rate, then by Jensen's inequality we can simplify Thm. 2.1 as follows:

$$\begin{aligned} r &\geq \frac{H(y^{(1)} \mid x^{(1)}) - I(w; \mathbf{y} \mid \mathbf{x})/n - \frac{1}{n}\sum_{i=1}^{n} H(e^{(i)})}{\log(k-1)} \\ &\geq \frac{H(y^{(1)} \mid x^{(1)}) - I(w; \mathbf{y} \mid \mathbf{x})/n - H(r)}{\log(k-1)}. \end{aligned} \tag{2}$$

Solving this inequality for r is challenging. One can simplify the right hand side further by bounding $H(e^{(1)}) \leq 1$ (assuming that entropies are measured in bits). However, this will loosen the bound. Alternatively, we can find the smallest r_0 for which (2) holds and claim that $r \geq r_0$.

Remark 2. If $|\mathcal{Y}| = 2$, then $\log(|\mathcal{Y}| - 1) = 0$, putting which in (13) of supplementary leads to:

$$H(r) \geq H(y^{(1)} \mid x^{(1)}) - I(w; \mathbf{y} \mid \mathbf{x})/n.$$

Remark 3. When we have uniform label noise where a label is incorrect with probability p ($0 \leq p < \frac{k-1}{k}$) and $I(w; \mathbf{y} \mid \mathbf{x}) = 0$, the bound of (2) is tight, i.e., implies that $r \geq p$. To see this, we note that $H(y^{(1)} \mid x^{(1)}) = H(p) + p\log(k-1)$, putting which in (2) gives us:

$$r \geq \frac{H(p) + p\log(k-1) - H(r)}{\log(k-1)} = p + \frac{H(p) - H(r)}{\log(k-1)}. \tag{3}$$

Therefore, when $r = p$, the inequality holds, implying that $r_0 \leq p$. To show that $r_0 = p$, we need to show that for any $0 \leq r < p$, the (3) does not hold. Let $r \in [0, p)$ and assume

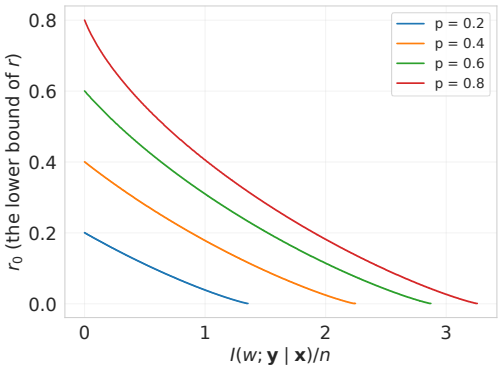


Figure 2: The lower bound r_0 on the rate of training errors r Thm. 2.1 establishes for varying values of $I(w; \mathbf{y} \mid \mathbf{x})$, in the case when label noise is uniform and probability of a label being incorrect is p.

that (3) holds. Then

$$\begin{aligned} r &\geq p + \frac{H(p) - H(r)}{\log(k-1)} \\ &\geq p + \frac{H(p) - (H(p) + (r-p)H'(p))}{\log(k-1)} \\ &\geq p + \frac{-(r-p)\log(k-1)}{\log(k-1)} = 2p - r. \end{aligned} \tag{4}$$

The second line above follows from concavity of $H(x)$; and the third line follows from the fact that $H'(p) > -\log(k-1)$ when $0 \leq p < (k-1)/k$. Eq. (4) directly contradicts with $r < p$. Therefore, Eq. (3) cannot hold for any $r < p$.

When $I(w; \mathbf{y} \mid \mathbf{x}) > 0$, we can find the smallest r_0 by a numerical method. Fig. 2 plots r_0 vs $I(w; \mathbf{y} \mid \mathbf{x})$ when the label noise is uniform. When the label-noise is not uniform, the bound of (2) becomes loose as Fano's inequality becomes loose. We leave the problem of deriving better lower bounds in such cases for a future work.

Thm. 2.1 provides theoretical guarantees that memorization of noisy labels is prevented when $I(w; \mathbf{y} \mid \mathbf{x})$ is small, in contrast to standard regularization techniques – such as dropout, weight decay, and data augmentation – which only slow it down (Zhang et al., 2016; Arpit et al., 2017). To demonstrate this empirically, we compare an algorithm that controls $I(w; \mathbf{y} \mid \mathbf{x})$ (presented in Sec. 3) against these regularization techniques on the aforementioned corrupted MNIST setup. We see in Fig. 1 that explicitly preventing memorization of label-noise information leads to optimal training performance (20% training accuracy) and good generalization on a non-corrupted validation set. Other approaches quickly exceed 20% training accuracy by incorporating label-noise information, and generalize poorly as a consequence. The classifier here is a fully connected neural network with 4 hidden layers each having 512 ReLU

units. The rates of dropout and weight decay were selected according to the performance on a validation set.

2.2. Decreasing $I(w; \mathbf{y} \mid \mathbf{x})$ Improves Generalization

The information that weights contain about a training dataset S has previously been linked to generalization (Xu & Raginsky, 2017). The following bound relates the expected difference between train and test performance to the mutual information $I(w; S)$.

Theorem 2.2. (Xu & Raginsky, 2017) Suppose $\ell(\hat{y}, y)$ is a loss function, such that $\ell(f_w(x), y)$ is σ-sub-Gaussian random variable for each w. Let $S = (\mathbf{x}, \mathbf{y})$ be the training set, $\mathcal{A}(w \mid S)$ be the training algorithm, and $(\bar{x}, \bar{y})$ be a test sample independent from S and w. Then the following holds:

$$\left| \mathbb{E}\left[\ell(f_w(\bar{x}), \bar{y}) - \frac{1}{n} \sum_{i=1}^{n} \ell\left(f_w(x^{(i)}), y^{(i)} \right) \right] \right| \leq \sqrt{\frac{2\sigma^2}{n} I(w; S)} \tag{5}$$

For good test performance, learning algorithms need to have both a small generalization gap, and good training performance. The latter may require retaining more information about the training set, meaning there is a natural conflict between increasing training performance and decreasing the generalization gap bound of (5). Furthermore, information in weights can be decomposed as follows: $I(w; S) = I(w; \mathbf{x}) + I(w; \mathbf{y} \mid \mathbf{x})$. We claim that one needs to prioritize reducing $I(w; \mathbf{y} \mid \mathbf{x})$ over $I(w; \mathbf{x})$ for the following reason. When noise is present in the training labels, fitting this noise implies a non-zero value of $I(w; \mathbf{y} \mid \mathbf{x})$, which grows linearly with the number of samples n. In such cases, the generalization gap bound of (5) becomes a constant and does not improve as n increases. To get meaningful generalization bounds via (5) one needs to limit $I(w; \mathbf{y} \mid \mathbf{x})$. We hypothesize that for efficient learning algorithms, this condition might be also sufficient.

3. Methods Limiting Label Information

We now consider how to design training algorithms that control $I(w; \mathbf{y} \mid \mathbf{x})$. We assume $f(y \mid x, w) = \text{Multinoulli}(y; s(a))$, with a as the output of a neural network $h_w(x)$, and $s(\cdot)$ as the softmax function. We consider the case when $h_w(x)$ is trained with a variant of stochastic gradient descent for T iterations. The inputs and labels of a mini-batch at iteration t are denoted by x_t and y_t respectively, and are selected using a deterministic procedure (such as cycling through the dataset, or using pseudo-randomness). Let w_0 denote the weights after initialization, and w_t the weights after iteration t. Let $\mathcal{L}(w; x, y)$ be some classification loss function (e.g, cross-entropy loss) and $g_t^{\mathcal{L}} \triangleq \nabla_w \mathcal{L}(w_{t-1}; x_t, y_t)$ be the gradient at iteration t. Let g_t denote the gradients used to update the weights, possibly different from $g_t^{\mathcal{L}}$. Let the update rule be $w_t = \Psi(w_0, g_{1:t})$, and $w_T = \Psi(w_0, g_{1:T})$ be the final weights (denoted with w for convenience).

To limit $I(w; \mathbf{y} \mid \mathbf{x})$, the following sections will discuss two approximations which relax the computational difficulty, while still provide meaningful bounds: 1) first, we show that the information in weights can be replaced by information in the gradients; 2) we introduce a variational bound on the information in gradients. The bound employs an auxiliary network that predicts gradients of the original loss without label information. We then explore two ways of incorporating predicted gradients: (a) using them in a regularization term for gradients of the original loss, and (b) using them to train the classifier.

3.1. Penalizing Information in Gradients

Looking at (1) it is tempting to add $I(w; \mathbf{y} \mid \mathbf{x})$ as a regularization to the $H_{p,f}(\mathbf{y} \mid \mathbf{x}, w)$ objective and minimize over all training algorithms:

$$\min_{\mathcal{A}(w|D)} H_{p,f}(\mathbf{y} \mid \mathbf{x}, w) + I(w; \mathbf{y} \mid \mathbf{x}). \tag{6}$$

This will become equivalent to minimizing $\mathbb{E}_{\mathbf{x},w} \text{KL}\big[p(\mathbf{y} \mid \mathbf{x}) || f(\mathbf{y} \mid \mathbf{x}, w)\big]$. Unfortunately, the optimization problem of (6) is hard to solve for two major reasons. First, the optimization is over training algorithms (rather than over the weights of a classifier, as in the standard machine learning setup). Second, the penalty $I(w; \mathbf{y} \mid \mathbf{x})$ is hard to compute/approximate.

To simplify the problem of (6), we relate information in weights to information in gradients as follows:

$$I(w; \mathbf{y} \mid \mathbf{x}) \leq I(g_{1:T}; \mathbf{y} \mid \mathbf{x}) = \sum_{t=1}^{T} I(g_t; \mathbf{y} \mid \mathbf{x}, g_{<t}), \tag{7}$$

where $g_{1:T}$ and $g_{<t}$ are shorthands for sets $\{g_1, \ldots, g_T\}$ and $\{g_1, \ldots, g_{t-1}\}$ respectively. Hereafter, we focus on constraining $I(g_t; \mathbf{y} \mid \mathbf{x}, g_{<t})$ at each iteration. Our task becomes choosing a loss function such that $I(g_t; \mathbf{y} \mid \mathbf{x}, g_{<t})$ is small and $f(y \mid x, w_t)$ is a good classifier. One key observation is that if our task is to minimize label-noise information in gradients it may be helpful to consider gradients with respect to the last layer only and compute the remaining gradients using back-propagation. As these steps of back-propagation do not use labels, by data processing inequality, subsequent gradients would have at most as much label information as the last layer gradient.

To simplify information-theoretic quantities, we add a small independent Gaussian noise to the gradients of the original loss: $\tilde{g}_t^{\mathcal{L}} \triangleq g_t^{\mathcal{L}} + \xi_t$, where $\xi_t \sim \mathcal{N}(0, \sigma_\xi^2 I)$ and σ_ξ is small enough to have no significant effect on training (less than 10^{-9} is fine). With this convention, we formulate the

following regularized objective function:

$$\min_w \mathcal{L}(w; x_t, y_t) + \lambda I(\tilde{g}_t^{\mathcal{L}}; \mathbf{y} \mid \mathbf{x}, g_{<t}), \tag{8}$$

where $\lambda > 0$ is a regularization coefficient. The term $I(\tilde{g}_t^{\mathcal{L}}; \mathbf{y} \mid \mathbf{x}, g_{<t})$ is a function of $\mathbf{x}$ and $g_{<t}$, or more explicitly, a function $\Phi(w_{t-1}; x_t)$ of x_t and w_{t-1}. Computing this function would allow the optimization of (8) through gradient descent: $g_t = g_t^{\mathcal{L}} + \xi_t + \nabla_w \Phi(w_{t-1}; x_t)$. Importantly, label-noise information is equal in both g_t and $\tilde{g}_t^{\mathcal{L}}$, as the gradient from the regularization is constant given $\mathbf{x}$ and $g_{<t}$:

$$\begin{aligned} I(g_t; \mathbf{y} \mid \mathbf{x}, g_{<t}) &= I(g_t^{\mathcal{L}} + \xi_t + \nabla_w \Phi(w_{t-1}; x_t); \mathbf{y} \mid \mathbf{x}, g_{<t}) \\ &= I(g_t^{\mathcal{L}} + \xi_t; \mathbf{y} \mid \mathbf{x}, g_{<t}) = I(\tilde{g}_t^{\mathcal{L}}; \mathbf{y} \mid \mathbf{x}, g_{<t}). \end{aligned}$$

Therefore, by minimizing $I(\tilde{g}_t^{\mathcal{L}}; \mathbf{y} \mid \mathbf{x}, g_{<t})$ in (8) we minimize $I(g_t; \mathbf{y} \mid \mathbf{x}, g_{<t})$, which is used to upper bound $I(w; \mathbf{y} \mid \mathbf{x})$ in (7). We rewrite this regularization in terms of entropy and discard the constant term, $H(\xi_t)$:

$$\begin{aligned} I(\tilde{g}_t^{\mathcal{L}}; \mathbf{y} \mid \mathbf{x}, g_{<t}) &= H(\tilde{g}_t^{\mathcal{L}} \mid \mathbf{x}, g_{<t}) - H(\tilde{g}_t^{\mathcal{L}} \mid \mathbf{x}, \mathbf{y}, g_{<t}) \\ &= H(\tilde{g}_t^{\mathcal{L}} \mid \mathbf{x}, g_{<t}) - H(\xi_t). \end{aligned} \tag{9}$$

3.2. Variational Bounds on Gradient Information

The first term in (9) is still challenging to compute, as we typically only have one sample from the unknown distribution $p(y_t \mid x_t)$. Nevertheless, we can upper bound it with the cross-entropy $H_{p,q} = -\mathbb{E}_{\tilde{g}_t^{\mathcal{L}}}\left[\log q_\phi(\tilde{g}_t^{\mathcal{L}} \mid \mathbf{x}, g_{<t})\right]$, where $q_\phi(\cdot \mid \mathbf{x}, g_{<t})$ is a variational approximation for $p(\tilde{g}_t^{\mathcal{L}} \mid \mathbf{x}, g_{<t})$:

$$H(\tilde{g}_t^{\mathcal{L}} \mid \mathbf{x}, g_{<t}) \leq -\mathbb{E}\left[\log q_\phi(\tilde{g}_t^{\mathcal{L}} \mid \mathbf{x}, g_{<t})\right].$$

This bound is correct when ϕ is a constant or a random variable that depends only on $\mathbf{x}$. With this upper bound, (8) reduces to:

$$\min_{w,\phi} \mathcal{L}(w; x_t, y_t) - \lambda \mathbb{E}_{\tilde{g}_t^{\mathcal{L}}}\left[\log q_\phi(\tilde{g}_t^{\mathcal{L}} \mid \mathbf{x}, g_{<t})\right]. \tag{10}$$

This formalization introduces a soft constraint on the classifier by attempting to make its gradients predictable without labels $\mathbf{y}$, effectively reducing $I(g_t; \mathbf{y} \mid \mathbf{x}, g_{<t})$.

Assuming $\widehat{y} = s(h_w(x))$ denotes the predicted class probabilities of the classifier and $\mathcal{L}$ is the cross-entropy loss, the gradient with respect to logits $a = h_w(x)$ is $\widehat{y} - y$ (assuming y has a one-hot encoding). Thus, $I(\tilde{g}_t^{\mathcal{L}}; \mathbf{y} \mid \mathbf{x}, g_{<t}) = I(\widehat{y}_t - y_t + \xi_t; \mathbf{y} \mid \mathbf{x}, g_{<t}) = I(y_t + \xi_t; \mathbf{y} \mid \mathbf{x}, g_{<t})$. Since this expression has no dependence on w_{t-1}, it would not serve as a meaningful regularizer. Instead, we descend an additional level to look at gradients of the final layer parameters. When the final layer of $h_w(x)$ is fully connected with inputs z and weights U (i.e., $a = Uz$), the gradients with respect to its parameters is equal to $(\widehat{y} - y)z^T$. With this gradient, $I(\tilde{g}_t^{\mathcal{L}}; \mathbf{y} \mid \mathbf{x}, g_{<t}) = I((\widehat{y}_t - y_t)z_t^T + \xi_t; \mathbf{y} \mid \mathbf{x}, g_{<t}) = I(y_t z_t^T + \xi_t; \mathbf{y} \mid \mathbf{x}, g_{<t})$. There is now dependence on w_{t-1} through z_t, as this quantity can be reduced by setting z_t to a small value. We choose to parametrize $q_\phi(\cdot \mid \mathbf{x}, g_{<t})$ as a Gaussian distribution with mean $\mu_t = (s(a_t) - s(r_\phi(x_t)))z_t^T$ and fixed covariance $\sigma_q I$, where $r_\phi(\cdot)$ is another neural network. Under this assumption, $H_{p,q}$ becomes proportional to:

$$\begin{aligned} \mathbb{E}\left[\left\|(\widehat{y}_t - y_t)z_t^T + \xi_t - \mu_t\right\|_2^2\right] &= \mathbb{E}\left[\xi_t^2\right] \\ &+ \mathbb{E}\left[\|z_t\|_2^2 \left\|y_t - s(r_\phi(x_t))\right\|_2^2\right]. \end{aligned}$$

Ignoring constants and approximating the expectation above with one Monte Carlo sample computed using the label y_t, the objective of (10) becomes:

$$\min_w \mathcal{L}(w; x_t, y_t) + \lambda\left[\|z_t\|_2^2 \left\|y_t - s(r_\phi(x_t))\right\|_2^2\right]. \tag{11}$$

While this may work in principle, in practice the dependence on w is only felt through the norm of z, making it too weak to have much effect on the overall objective. We confirm this experimentally in Sec. 4. To introduce more complex dependencies on w, one would need to model the gradients of deeper layers.

3.3. Predicting Gradients without Label Information

An alternative approach is to use gradients predicted by $q_\phi(\cdot \mid \mathbf{x}, g_{<t})$ to update classifier weights, i.e., sample $g_t \sim q_\phi(\cdot \mid \mathbf{x}, g_{<t})$. This is a much stricter condition, as it implies $I(g_t; \mathbf{y} \mid \mathbf{x}, g_{<t}) = 0$ (again assuming ϕ is a constant or a random variable that depends only on $\mathbf{x}$). Note that minimizing $H_{p,q}$ makes the predicted gradient g_t a good substitute for the cross-entropy gradients $\tilde{g}_t^{\mathcal{L}}$. Therefore, we write down the following objective function:

$$\min_{w,\phi} \tilde{\mathcal{L}}(w_{t-1}; \phi, x_t) - \lambda \mathbb{E}_{\tilde{g}_t^{\mathcal{L}}}\left[\log q_\phi(\tilde{g}_t^{\mathcal{L}} \mid \mathbf{x}, g_{<t})\right], \tag{12}$$

where $\tilde{\mathcal{L}}(w; \phi, x)$ is a probabilistic function defined implicitly such that $\nabla_w \tilde{\mathcal{L}}(w; \phi, x) \sim q_\phi(\cdot \mid x, w)$. We found that this approach performs significantly better than the penalizing approach of (10). To update w only with predicted gradients, we disable the dependence of the second term of (12) on w in the implementation. Additionally, one can set $\lambda = 1$ above as the first term depends only on w, while the second term depends only on ϕ.

We choose to predict the gradients with respect to logits only and compute the remaining gradients using backpropagation. We consider two distinct parameterizations for q_ϕ – **Gaussian:** $q_\phi(\cdot \mid x, w) = \mathcal{N}\left(\mu, \sigma_q^2 I\right)$, and **Laplace:** $q_\phi(\cdot \mid x, w) = \prod_j \text{Lap}\left(\mu_j, \sigma_q/\sqrt{2}\right)$, with $\mu = s(a) - s(r_\phi(x))$ and $r_\phi(\cdot)$ being an auxiliary neural network as before. Under these Gaussian and Laplace parameterizations, $H_{p,q}$

becomes proportional to $\mathbb{E}||\mu_t - \tilde{g}_t^{\mathcal{L}}||_2^2$ and $\mathbb{E}||\mu_t - \tilde{g}_t^{\mathcal{L}}||_1$ respectively. In the Gaussian case ϕ is updated with a mean square error loss (MSE) function, while in the Laplace case it is updated with a mean absolute error loss (MAE). The former is expected to be faster at learning, but less robust to noise (Ghosh et al., 2017).

3.4. Reducing Overfitting in Gradient Prediction

In both approaches of (10) and (12), the classifier can still overfit if $q_\phi(\cdot \mid x, w)$ overfits. There are multiple ways to prevent this. One can choose q_ϕ to be parametrized with a small network, or pre-train and freeze some of its layers in an unsupervised fashion. In this work, we choose to control the L2 norm of the mean of predicted gradients, $\|\mu\|_2^2$, while keeping the variance σ_q^2 fixed. This can be viewed as limiting the capacity of gradients g_t.

Proposition 3.1. If $g_t = \mu_t + \epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, \sigma_q^2 I_d)$ is independent noise, and $\mathbb{E}\left[\mu_t^T \mu_t\right] \leq L^2$, then the following inequality holds:

$$I(g_t; \mathbf{y} \mid \mathbf{x}, g_{<t}) \leq \frac{d}{2} \log\left(1 + \frac{L^2}{d\sigma_q^2}\right).$$

The proof is provided in the supplementary section A.2. The same bound holds when ϵ_t is sampled from a product of d univariate zero-mean Laplace distributions with variance σ_q^2, since the proof relies only on ϵ_t being zero-mean and having variance σ_q^2. The final objective of our main method becomes:

$$\min_{w,\phi} \tilde{\mathcal{L}}(w; \phi, x_t) - \lambda \mathbb{E}_{\tilde{g}_t^{\mathcal{L}}}\left[\log q_\phi(\tilde{g}_t^{\mathcal{L}} \mid \mathbf{x}, g_{<t})\right] + \beta \|\mu_t\|_2^2.$$

As before, to update w only with predicted gradients, we disable the dependence of the second and third terms above on w in the implementation. We name this final approach **LIMIT** – **l**imiting label **i**nformation **m**emorization **i**n **t**raining. We denote the variants with Gaussian and Laplace distributions as LIMIT$_{\mathcal{G}}$ and LIMIT$_{\mathcal{L}}$ respectively. The pseudocode of LIMIT is presented in the supplementary material (Alg. 1). Note that in contrast to the previous approach of (8), this follows the spirit of (6), in the sense that the optimization over ϕ can be seen as optimizing over training algorithms; namely, learning a loss function $\mathcal{L}'$ implicitly through gradients. With this interpretation, the gradient norm penalty can be viewed as a way to smooth the learned loss, which is a good inductive bias and facilitates learning.

4. Experiments

We set up experiments with noisy datasets to see how well the proposed methods perform for different types and amounts of label noise. The simplest baselines in our comparison are standard cross-entropy (CE) and mean absolute error (MAE) loss functions. The next baseline is the forward correction approach (FW) proposed by (Patrini et al., 2017), where the label-noise transition matrix is estimated and used to correct the loss function. Finally, we include the recently proposed determinant mutual information (DMI) loss, which is the log-determinant of the confusion matrix between predicted and given labels (Xu et al., 2019). Both FW and DMI baselines require initialization with the best result of the CE baseline. To avoid small experimental differences, we implement all baselines, closely following the original implementations of FW and DMI. We train all baselines except DMI using the ADAM optimizer (Kingma & Ba, 2014) with learning rate $\alpha = 10^{-3}$ and $\beta_1 = 0.9$. As DMI is very sensitive to the learning rate, we tune it by choosing the best from the following grid of values $\{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$. For all baselines, model selection is done by choosing the model with highest accuracy on a validation set that follows the noise model of the corresponding train set. All scores are reported on a clean test set. Additional experimental details, including the hyperparameter grids, are presented in supplementary section B. The implementation of the proposed method and the code for replicating the experiments is available at `https://github.com/hrayrhar/limit-label-memorization`.

4.1. MNIST with Uniform Label Corruption

To compare the variants of our approach discussed earlier and see which ones work well, we do experiments on the MNIST dataset with corrupted labels. In this experiment, we use a simple uniform label-noise model, where each label is set to an incorrect value uniformly at random with probability p. In our experiments we try 4 values of p – 0%, 50%, 80%, 89%. We split the 60K images of MNIST into training and validation sets, containing 48K and 12K samples respectively. For each noise amount we try 3 different training set sizes – 10^3, 10^4, and $4.8 \cdot 10^4$. All classifiers and auxiliary networks are 4-layer CNNs, with a shared architecture presented in the supplementary (Sec. B). For this experiment we include two additional baselines where additive noise (Gaussian or Laplace) is added to the gradients with respect to logits. We denote these baselines with names "CE + GN" and "CE + LN". The comparison with these two baselines demonstrates that the proposed method does more than simply reduce information in gradients via noise. We also consider a variant of LIMIT where instead of sampling g_t from q we use the predicted mean μ_t.

Table 1 shows the test performances of different approaches averaged over 5 training/validation splits. Standard deviations and additional combinations of p and n are presented in the supplementary (See tables 4 and 5). Additionally, Fig. 3 shows the training and testing performances of the best methods during the training when $p = 0.8$ and all training samples are used. Overall, variants of LIMIT produce

Method	$p = 0.0$		$p = 0.5$		$p = 0.8$	
	10^3	All	10^3	All	10^3	All
CE	94.3	**99.2**	71.8	97.2	27.0	87.2
CE + GN	89.5	97.1	70.5	97.4	25.9	85.3
CE + LN	90.0	96.7	66.8	97.6	30.2	74.5
MAE	94.6	**99.1**	75.6	98.1	25.1	93.2
FW	93.6	**99.2**	64.3	97.3	19.0	89.1
DMI	94.5	**99.2**	79.8	98.3	30.3	88.8
Soft reg. (11)	**95.7**	**99.2**	76.4	98.2	28.8	89.3
LIMIT$_{\mathcal{G}}$ + S	**95.6**	**99.3**	82.8	98.7	**35.9**	93.4
LIMIT$_{\mathcal{L}}$ + S	94.8	**99.3**	**88.7**	**98.9**	**35.6**	**97.6**
LIMIT$_{\mathcal{G}}$ - S	**95.7**	**99.3**	83.3	98.6	**37.1**	94.7
LIMIT$_{\mathcal{L}}$ - S	95.0	**99.3**	**88.2**	**99.0**	**35.9**	**97.7**

Table 1: Test accuracy comparison on multiple versions of MNIST corrupted with uniform label noise. Error bars are presented in the supplementary material (Sec. C)

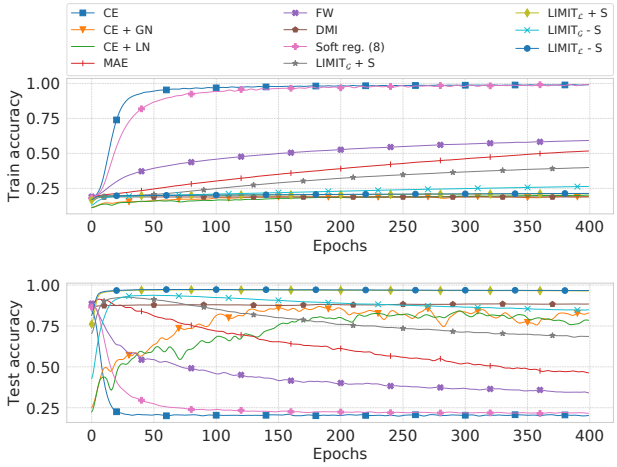


Figure 3: Smoothed training and testing accuracy plots of various approaches on MNIST with 80% uniform noise.

the best results and improve significantly over standard approaches. The variants with a Laplace distribution perform better than those with a Gaussian distribution. This is likely due to the robustness of MAE. Interestingly, LIMIT works well and trains faster when the sampling of g_t in q is disabled (rows with "-S"). Thus, hereafter we consider this as our primary approach. As expected, the soft regularization approach of (10) and cross-entropy variants with noisy gradients perform significantly worse than LIMIT. We exclude these baselines in our future experiments. Additionally, we tested the importance of penalizing norm of predicted gradients by comparing training and testing performances of LIMIT with varying regularization strength β in the supplementary (Fig. A0). We found that this penalty is essential for preventing memorization.

In our approach, the auxiliary network q should not be able to distinguish correct and incorrect samples, unless it overfits. We found that it learns to predict "correct" gradients on examples with incorrect labels (Sec. C). Motivated by this, we use the distance between predicted and cross-entropy gradients to detect samples with incorrect or misleading labels (Fig. 4). Additionally, when considering the distance as a score for for classifying correctness of a label, distance, we get 99.87% ROC AUC score (Sec. C).

4.2. CIFAR with Uniform and Pair Noise

Next we consider a harder dataset, CIFAR-10 (Krizhevsky et al., 2009), with two label noise models: uniform noise and pair noise. For pair noise, certain classes are confused with some other similar class. Following the setup of Xu et al. (2019) we use the following four pairs: truck → automobile, bird → airplane, deer → horse, cat → dog. Note in this type of noise $H(\mathbf{y} \mid \mathbf{x})$ is much smaller than in the case of uniform noise. We split the 50K images of CIFAR-10 into training and validation sets, containing 40K and 10K samples respectively. For the CIFAR experiments we use ResNet-34 networks (He et al., 2016) with standard data augmentation, consisting of random horizontal flips and random 28x28 crops padded back to 32x32. For our proposed methods, the auxiliary network q is ResNet-34 as well. We noticed that for more difficult datasets, it may happen that while q still learns to produce good gradients, the updates with these less informative gradients may corrupt the initialization of the classifier. For this reason, we add an additional variant of LIMIT, which initializes the q network with the best CE baseline, similar to the DMI and FW baselines.

Table 2 presents the results on CIFAR-10. Again, variants of LIMIT improve significantly over standard baselines, especially in the case of uniform label noise. As expected, when q is initialized with the best CE model (similar to FW and DMI), the results are better. As in the case of MNIST, our approach helps even when the dataset is noiseless.

CIFAR-100. To test proposed methods on a classification task with many classes, we apply them on CIFAR-100 with 40% uniform noise. We use the same networks as in the case of CIFAR-10. Results presented in Table 2 indicate several interesting phenomena. First, training with the MAE loss fails, which was observed by other works as well (Zhang & Sabuncu, 2018). The gradient of MAE with respect to logits is $f(x)_y(\widehat{y} - y)$. When $f(x)_y$ is small, there is small signal to fix the mistake. In fact, in the case of CIFAR-100, $f(x)_y$ is approximately 0.01 in the beginning, slowing down the training. The performance of FW degrades as the approximation error of noise transition matrix become large. The DMI does not give significant improvement over CE due to numerical issues with computing a determinant of a 100x100 confusion matrix. LIMIT$_{\mathcal{L}}$ performs worse than other variants, as training q with MAE becomes challenging.

Figure 4: Most mislabeled examples in MNIST (top left), CIFAR-10 (top right), and Clothing1M (bottom) datasets, according to the distance between predicted and cross-entropy gradients. More examples are presented in the supplementary (Sec. C).

Method	CIFAR-10									CIFAR-100	Clothing1M
	no noise	uniform noise				pair noise				uniform noise	noisy train set
	0.0	0.2	0.4	0.6	0.8	0.1	0.2	0.3	0.4	0.4	-
CE	92.7	85.2	81.0	69.0	38.8	90.0	88.1	87.2	81.8	44.9 ± 0.5	68.91 ± 0.46
MAE	84.4	85.4	64.6	15.4	12.0	88.6	83.2	72.1	61.1	1.8 ± 0.1	6.52 ± 0.23
FW	92.9	86.2	81.4	69.7	34.4	90.1	88.0	86.8	84.6	23.3 ± 0.4	68.70 ± 0.45
DMI	93.0	88.3	85.0	72.5	38.9	91.4	90.6	90.4	89.6	46.1 ± 0.5	$\mathbf{71.19 \pm 0.43}$
LIMIT$_{\mathcal{G}}$	**93.5**	90.7	86.6	73.7	38.7	92.8	91.3	89.2	86.0	58.4 ± 0.4	70.32 ± 0.42
LIMIT$_{\mathcal{L}}$	93.1	91.5	88.2	75.7	35.8	91.9	91.1	88.8	84.2	49.5 ± 0.5	70.35 ± 0.45
LIMIT$_{\mathcal{G}}$ + init.	**93.3**	**92.4**	**90.3**	81.9	**44.1**	**93.3**	**92.9**	**90.8**	88.3	59.2 ± 0.5	$\mathbf{71.39 \pm 0.44}$
LIMIT$_{\mathcal{L}}$ + init.	**93.3**	**92.2**	**90.2**	**82.9**	**44.3**	**93.0**	92.3	**91.1**	**90.0**	$\mathbf{60.8 \pm 0.5}$	70.53 ± 0.44

Table 2: Test accuracy comparison on CIFAR-10, corrupted with various label noise types, on CIFAR-100 with 40% uniform label noise and on Clothing1M dataset. The error bars are computed by bootstrapping the test set 1000 times. The missing error bars are presented in the supplementary material (Sec. C).

However, performance improves when q is initialized with the CE model. LIMIT$_{\mathcal{G}}$ does not suffer from the mentioned problem and works with or without initialization.

4.3. Clothing1M

Finally, as in our last experiment, we consider the Clothing1M dataset (Xiao et al., 2015), which has 1M images labeled with one of 14 possible clothing labels. The dataset has very noisy training labels, with roughly 40% of examples incorrectly labeled. More importantly, the label noise in this dataset is realistic and instance dependent. For this dataset we use ResNet-50 networks and employ standard data augmentation, consisting of random horizontal flips and random crops of size 224x224 after resizing images to size 256x256. The results shown in the last column of Table 2 demonstrate that DMI and LIMIT with initialization perform the best, producing similar results.

5. Related Work

Our approach is related to many works that study memorization and learning with noisy labels. Our work also builds on theoretical results studying how generalization relates to information in neural network weights. In this section we present the related work and discuss the connections.

5.1. Learning with Noisy Labels

Learning with noisy labels is a longstanding problem and has been studied extensively (Frenay & Verleysen, 2014). Many works studied and proposed loss functions that are robust to label noise. Natarajan et al. (2013) propose robust loss functions for binary classification with label-dependent noise. Ghosh et al. (2017) generalize this result for multi-class classification problem and show that the mean absolute error (MAE) loss function is tolerant to label-dependent noise. Zhang & Sabuncu (2018) propose a new loss function, called generalized cross-entropy (GCE), that interpolates between MAE and CE with a single parameter $q \in [0, 1]$. Xu et al. (2019) propose a new loss function (DMI), which is equal to the log-determinant of the confusion matrix between predicted and given labels, and show that it is robust to label-dependent noise. These loss functions are robust in the sense that the best performing hypothesis on clean data and noisy data are the same in the regime of infinite data. When training on finite datasets, training with these loss functions may result in memorization of training labels.

Another line of research seeks to estimate label-noise and correct the loss function accordingly (Sukhbaatar et al., 2014; Tong Xiao et al., 2015; Goldberger & Ben-Reuven, 2017; Patrini et al., 2017; Hendrycks et al., 2018; Yao et al., 2019). Some works use meta-learning to treat the problem

of noisy/incomplete labels as a decision problem in which one determines the reliability of a sample (Jiang et al., 2017; Ren et al., 2018; Shu et al., 2019). Others seek to detect incorrect examples and relabel them (Reed et al., 2014; Tanaka et al., 2018; Ma et al., 2018; Han et al., 2019; Arazo et al., 2019). Han et al. (2018); Yu et al. (2019) employ an approach where two networks select training examples for each other using the small-loss trick. While our approach also has a teaching component, the network uses all samples instead of filtering. Li et al. (2019) propose a meta-learning approach that optimizes a classification loss along with a consistency loss between predictions of a mean teacher and predictions of the model after a single gradient descent step on a synthetically labeled mini-batch.

Some approaches assume particular label-noise models, while our approach assumes that $H(\mathbf{y} \mid \mathbf{x}) > 0$, which may happen because of any type of label noise or attribute noise (e.g., corrupted images or partially observed inputs). Additionally, the techniques used to derive our approach can be adopted for regression or multilabel classification tasks. Furthermore, some methods require access to small clean validation data, which is not required in our approach.

5.2. Information in Weights and Generalization

Defining and quantifying information in neural network weights is an open challenge and has been studied by multiple authors. One approach is to relate information in weights to their description length. A simple way of measuring description length was proposed by Hinton & van Camp (1993) and reduces to the L2 norm of weights. Another way to measure it is through the intrinsic dimension of an objective landscape (Li et al., 2018; Blier & Ollivier, 2018). Li et al. (2018) observed that the description length of neural network weights grows when they are trained with noisy labels (Li et al., 2018), indicating memorization of labels.

Achille & Soatto (2018) define information in weights as the KL divergence from the posterior of weights to the prior. In a subsequent study they provide generalization bounds involving the KL divergence term (Achille & Soatto, 2019). Similar bounds were derived in the PAC-Bayesian setup and have been shown to be non-vacuous (Dziugaite & Roy, 2017). With an appropriate selection of prior on weights, the above KL divergence becomes the Shannon mutual information between the weights and training dataset, $I(w; S)$. Xu & Raginsky (2017) derive generalization bounds that involve this latter quantity. Pensia et al. (2018) upper bound $I(w; S)$ when the training algorithm consists of iterative noisy updates. They use the chain-rule of mutual information as we did in (7) and bound information in updates by adding independent noise. It has been observed that adding noise to gradients can help to improve generalization in certain cases (Neelakantan et al., 2015). Another approach restricts information in gradients by clipping them (Menon et al., 2020) .

Achille & Soatto (2018) also introduce the term $I(w; \mathbf{y} \mid \mathbf{x})$ and show the decomposition of the cross-entropy described in (1). In a recent work, Yin et al. (2020) consider a similar term in the context of meta-learning and use it as a regularization to prevent memorization of meta-testing labels. Given a meta-learning dataset $\mathcal{M}$, they consider the information in the meta-weights θ about the labels of meta-testing tasks given the inputs of meta-testing tasks, $I(\theta; \mathcal{Y} \mid \mathcal{X})$. They bound this information with a variational upper bound $\mathrm{KL}\left(q(\theta \mid \mathcal{M}) || r(\theta)\right)$ and use multivariate Gaussian distributions for both. For isotropic Gaussians with equal covariances, the KL divergence reduces to $\|\theta - \theta_0\|_2^2$, which was studied by Hu et al. (2020) as a regularization to achieve robustness to label-noise. Note that this bounds not only $I(\theta; \mathcal{Y} \mid \mathcal{X})$ but also $I(\theta; \mathcal{X}, \mathcal{Y})$. In contrast, we bound only $I(w; \mathbf{y} \mid \mathbf{x})$ and work with information in gradients.

6. Conclusion and Future Work

Several recent theoretical works have highlighted the importance of the information about the training data that is memorized in the weights. We distinguished two components of it and demonstrated that the conditional mutual information of weights and labels given inputs is closely related to memorization of labels and generalization performance. By bounding this quantity in terms of information in gradients, we were able to derive the first practical schemes for controlling label information in the weights and demonstrated that this outperforms approaches for learning with noisy labels. In the future, we plan to explore ways of improving the bound of (7) and to design a better bottleneck in the gradients. Additionally, we aim to extend the presented ideas to reducing instance memorization.

Acknowledgements

We thank Alessandro Achille for his valuable comments. We also thank the anonymous reviewers whose suggestions helped improve this manuscript. Hrayr Harutyunyan was supported by the USC Annenberg Fellowship. This work is based in part on research sponsored by Air Force Research Laboratory (AFRL) under agreement number FA8750-19-1-1000. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation therein. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Laboratory, DARPA or the U.S. Government.

References

Achille, A. and Soatto, S. Emergence of invariance and disentanglement in deep representations. *J. Mach. Learn. Res.*, 19(1):1947–1980, January 2018. ISSN 1532-4435.

Achille, A. and Soatto, S. Where is the information in a deep neural network? *arXiv preprint arXiv:1905.12213*, 2019.

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., and McGuinness, K. Unsupervised label noise modeling and loss correction. *arXiv preprint arXiv:1904.11238*, 2019.

Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 233–242. JMLR. org, 2017.

Blier, L. and Ollivier, Y. The description length of deep learning models. In *Advances in Neural Information Processing Systems*, pp. 2216–2226, 2018.

Chen, P., Liao, B., Chen, G., and Zhang, S. Understanding and utilizing deep neural networks trained with noisy labels. *arXiv preprint arXiv:1905.05040*, 2019.

Cover, T. M. and Thomas, J. A. *Elements of information theory*. Wiley-Interscience, 2006.

Dziugaite, G. and Roy, D. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. 03 2017.

Frenay, B. and Verleysen, M. Classification in the presence of label noise: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25(5):845–869, May 2014. ISSN 2162-2388. doi: 10.1109/TNNLS.2013.2292894.

Ghosh, A., Kumar, H., and Sastry, P. Robust loss functions under label noise for deep neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Goldberger, J. and Ben-Reuven, E. Training deep neural-networks using a noise adaptation layer. In *ICLR*, 2017.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pp. 8527–8537, 2018.

Han, J., Luo, P., and Wang, X. Deep self-learning from noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5138–5147, 2019.

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

Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in neural information processing systems*, pp. 10456–10465, 2018.

Hinton, G. E. and van Camp, D. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, COLT '93, pp. 5–13, New York, NY, USA, 1993. Association for Computing Machinery. ISBN 0897916115. doi: 10.1145/168304.168306. URL https://doi.org/10.1145/168304.168306.

Hu, W., Li, Z., and Yu, D. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hke3gyHYwH.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. *arXiv preprint arXiv:1712.05055*, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Li, C., Farkhoor, H., Liu, R., and Yosinski, J. Measuring the intrinsic dimension of objective landscapes. 04 2018.

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

Ma, X., Wang, Y., Houle, M. E., Zhou, S., Erfani, S. M., Xia, S.-T., Wijewickrema, S., and Bailey, J. Dimensionality-driven learning with noisy labels. *arXiv preprint arXiv:1806.02612*, 2018.

Menon, A. K., Rawat, A. S., Reddi, S. J., and Kumar, S. Can gradient clipping mitigate label noise? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rklB76EKPr.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. Learning with noisy labels. In *Advances in neural information processing systems*, pp. 1196–1204, 2013.

Neelakantan, A., Vilnis, L., Le, Q. V., Sutskever, I., Kaiser, L., Kurach, K., and Martens, J. Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*, 2015.

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1944–1952, 2017.

Pensia, A., Jog, V., and Loh, P.-L. Generalization error bounds for noisy, iterative algorithms. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 546–550. IEEE, 2018.

Reed, S. E., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. *CoRR*, abs/1412.6596, 2014.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*, 2018.

Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, pp. 1917–1928, 2019.

Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L. D., and Fergus, R. Training convolutional networks with noisy labels. In *ICLR 2015*, 2014.

Tanaka, D., Ikami, D., Yamasaki, T., and Aizawa, K. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5552–5560, 2018.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2691–2699, June 2015. doi: 10.1109/CVPR.2015.7298885.

Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2691–2699, 2015.

Xu, A. and Raginsky, M. Information-theoretic analysis of generalization capability of learning algorithms. In *Advances in Neural Information Processing Systems*, pp. 2524–2533, 2017.

Xu, Y., Cao, P., Kong, Y., and Wang, Y. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *Advances in Neural Information Processing Systems*, pp. 6222–6233, 2019.

Yao, J., Wu, H., Zhang, Y., Tsang, I., and Sun, J. Safeguarded dynamic label regression for noisy supervision. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:9103–9110, 07 2019. doi: 10.1609/aaai.v33i01.33019103.

Yin, M., Tucker, G., Zhou, M., Levine, S., and Finn, C. Meta-learning without memorization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BklEFpEYwS.

Yu, X., Han, B., Yao, J., Niu, G., Tsang, I. W.-H., and Sugiyama, M. How does disagreement help generalization against label corruption? In *ICML*, 2019.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Zhang, Z. and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in neural information processing systems*, pp. 8778–8788, 2018.

Supplementary material: Improving Generalization by Controlling Label-Noise Information in Neural Network Weights

Algorithm 1 LIMIT: limiting label information memorization in training.
Our implementation is available at https://github.com/hrayrhar/limit-label-memorization.

Input: Training dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^n$.
Input: Gradient norm regularization coefficient β. $\{\lambda$ is set to $1\}$
Initialize classifier $f(y \mid x, w)$ and gradient predictor $q_\phi(\cdot \mid \mathbf{x}, g_{<t})$.
for $t = 1..T$ **do**
  Fetch the next batch (x_t, y_t) and compute the predicted logits a_t.
  Compute the cross-entropy gradient, $g_t^{\mathcal{L}} \leftarrow s(a_t) - y_t$.
  if sampling of gradients is enabled **then**
    $g_t \sim q_\phi(\cdot \mid \mathbf{x}, g_{<t})$.
  else
    $g_t \leftarrow \mu_t$ {the mean of predicted gradient}
  end if
  Starting with g_t, backpropagate to compute the gradient with respect to w.
  Update w_{t-1} to w_t.
  Update ϕ using the gradient of the following loss: $-\log q_\phi(\tilde{g}_t^{\mathcal{L}} \mid \mathbf{x}, g_{<t}) + \beta \|\mu_t\|_2^2$.
end for

A. Proofs

This section presents the proofs and some remarks that were not included in the main text due to space constraints.

A.1. Proof of Thm. 2.1

Theorem A.1. (Thm. 2.1 restated) Consider a dataset $S = (\mathbf{x}, \mathbf{y})$ of n i.i.d. samples, $\mathbf{x} = \{x^{(i)}\}_{i=1}^n$ and $\mathbf{y} = \{y^{(i)}\}_{i=1}^n$, where the domain of labels is a finite set, $\mathcal{Y}$, with $|\mathcal{Y}| > 2$. Let $\mathcal{A}(w \mid S)$ be any training algorithm, producing weights for possibly stochastic classifier $f(y \mid x, w)$. Let $\widehat{y}^{(i)}$ denote the prediction of the classifier on i-th example and $e^{(i)} = \mathbb{1}\{\widehat{y}^{(i)} \neq y^{(i)}\}$ be a random variable corresponding to predicting $y^{(i)}$ incorrectly. Then, the following holds

$$\mathbb{E}\left[\sum_{i=1}^n e^{(i)}\right] \geq \frac{H(\mathbf{y} \mid \mathbf{x}) - I(w; \mathbf{y} \mid \mathbf{x}) - \sum_{i=1}^n H(e^{(i)})}{\log\left(|\mathcal{Y}| - 1\right)}.$$

Proof. For each example we consider the following Markov chain:

$$y^{(i)} \to \begin{bmatrix}\mathbf{x} \\ \mathbf{y}\end{bmatrix} \to \begin{bmatrix}x^{(i)} \\ w\end{bmatrix} \to \widehat{y}^{(i)}.$$

In this setup Fano's inequality gives a lower bound for the error probability:

$$H(e^{(i)}) + \mathbb{P}(e^{(i)} = 1) \log\left(|\mathcal{Y}| - 1\right) \geq H(y^{(i)} \mid x^{(i)}, w), \tag{13}$$

which can be written as:

$$\mathbb{P}(e^{(i)} = 1) \geq \frac{H(y^{(i)} \mid x^{(i)}, w) - H(e^{(i)})}{\log\left(|\mathcal{Y}| - 1\right)}.$$

Summing this inequality for $i = 1, \dots, n$ we get

$$\begin{aligned}\sum_{i=1}^{n} \mathbb{P}(e^{(i)} = 1) &\geq \frac{\sum_{i=1}^{n} \left(H(y^{(i)} \mid x^{(i)}, w) - H(e^{(i)})\right)}{\log\left(|\mathcal{Y}| - 1\right)} \\ &\geq \frac{\sum_{i=1}^{n} \left(H(y^{(i)} \mid \mathbf{x}, w) - H(e^{(i)})\right)}{\log\left(|\mathcal{Y}| - 1\right)} \\ &\geq \frac{H(\mathbf{y} \mid \mathbf{x}, w) - \sum_{i=1}^{n} H(e^{(i)})}{\log\left(|\mathcal{Y}| - 1\right)}.\end{aligned}$$

The correctness of the last step follows from the fact that total correlation is always non-negative (Cover & Thomas, 2006):

$$\sum_{i=1}^{n} H(y^{(i)} \mid \mathbf{x}, w) - H(\mathbf{y} \mid \mathbf{x}, w) = \mathrm{TC}(\mathbf{y} \mid \mathbf{x}, w) \geq 0.$$

Finally, using the fact that $H(\mathbf{y} \mid \mathbf{x}, w) = H(\mathbf{y} \mid \mathbf{x}) - I(w; \mathbf{y} \mid \mathbf{x})$, we get that the desired result:

$$\mathbb{E}\left[\sum_{i=1}^{n} e^{(i)}\right] \geq \frac{H(\mathbf{y} \mid \mathbf{x}) - I(w; \mathbf{y} \mid \mathbf{x}) - \sum_{i=1}^{n} H(e^{(i)})}{\log\left(|\mathcal{Y}| - 1\right)}. \tag{14}$$

□

A.2. Proof of Prop. 3.1

Proposition A.1. (Prop. 3.1 restated) If $g_t = \mu_t + \epsilon_t$, where $\epsilon_t \sim \mathcal{N}(0, \sigma_q^2 I_d)$ is an independent noise and $\mathbb{E}\left[\mu_t^T \mu_t\right] \leq L^2$, then the following inequality holds:

$$I(g_t; \mathbf{y} \mid \mathbf{x}, g_{<t})) \leq \frac{d}{2} \log\left(1 + \frac{L^2}{d\sigma_q^2}\right).$$

Proof. Given that ϵ_t and μ_t are independent, let us bound the expected L2 norm of g_t:

$$\begin{aligned}\mathbb{E}\left[g_t^T g_t\right] &= \mathbb{E}\left[(\epsilon_t + \mu_t)^T (\epsilon_t + \mu_t)\right] \\ &= \mathbb{E}\left[\epsilon_t^T \epsilon_t\right] + \mathbb{E}\left[\mu_t^T \mu_t\right] \\ &\leq d\sigma_q^2 + L^2.\end{aligned}$$

Among all random variables Z with $\mathbb{E}[Z^T Z] \leq C$ the Gaussian distribution $Y \sim \mathcal{N}\left(0, \frac{C}{d} I_d\right)$ has the largest entropy, given by $H(Y) = \frac{d}{2} \log\left(\frac{2\pi e C}{d}\right)$. Therefore,

$$H(g_t) \leq \frac{d}{2} \log\left(\frac{2\pi e (d\sigma_q^2 + L^2)}{d}\right).$$

With this we can upper bound the $I(g_t; \mathbf{y} \mid \mathbf{x}, g_{<t})$ as follows:

$$\begin{aligned}I(g_t; \mathbf{y} \mid \mathbf{x}, g_{<t}) &= H(g_t \mid \mathbf{x}, g_{<t}) - H(g_t \mid \mathbf{x}, \mathbf{y}, g_{<t}) \\ &= H(g_t \mid \mathbf{x}, g_{<t}) - H(\epsilon_t) \\ &\leq \frac{d}{2} \log\left(\frac{2\pi e (d\sigma_q^2 + L^2)}{d}\right) - \frac{d}{2} \log\left(2\pi e \sigma_q^2\right) \\ &= \frac{d}{2} \log\left(1 + \frac{L^2}{d\sigma_q^2}\right).\end{aligned} \tag{15}$$

□

Layer type	Parameters
Conv	32 filters, 4×4 kernels, stride 2, padding 1, batch normalization, ReLU
Conv	32 filters, 4×4 kernels, stride 2, padding 1, batch normalization, ReLU
Conv	64 filters, 3×3 kernels, stride 2, padding 0, batch normalization, ReLU
Conv	256 filters, 3×3 kernels, stride 1, padding 0, batch normalization, ReLU
FC	128 units, ReLU
FC	10 units, linear activation

Table 3: The architecture of MNIST classifiers.

Note that the proof will work for arbitrary ϵ_t that has zero mean and independent components, where the L2 norm of each component is bounded by σ_q^2. This holds because in such cases $H(\epsilon_t) \leq \frac{d}{2}\log(2\pi e\sigma_q^2)$ (as Gaussians have highest entropy for fixed L2 norm) and the transition of (15) remains correct. Therefore, the same result holds when ϵ_t is sampled from a product of univariate zero-mean Laplace distributions with scale parameter $\sigma_q/\sqrt{2}$ (which makes the second moment equal to σ_q^2).

A similar result has been derived by Pensia et al. (2018) (lemma 5) to bound $I(w_t; (x_t, y_t) \mid w_{t-1})$.

B. Experimental Details

In this section we describe the details of experiments and implementations.

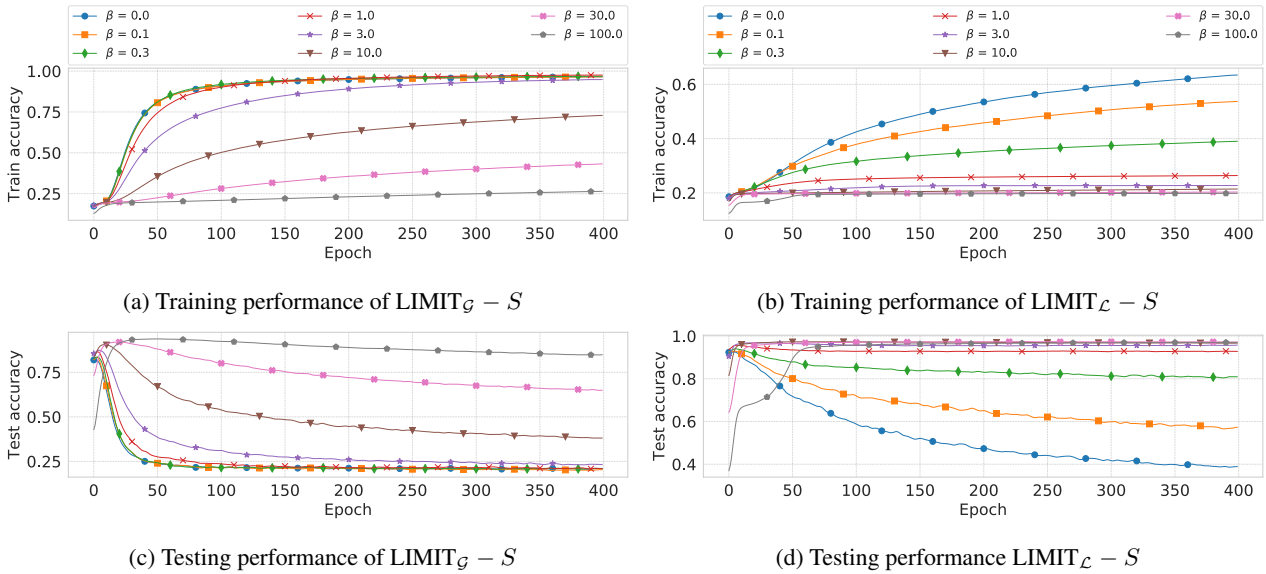
Classifier architectures. The architecture of classifiers used in MNIST experiments is presented in Table 3. The ResNet-34 used in CIFAR-10 and CIFAR-100 experiments differs from the standard ResNet-34 architecture (which is used for 224×224 images) in two ways: (a) the first convolutional layer has 3x3 kernels and stride 1 and (b) the max pooling layer after it is skipped. The architecture of ResNet-50 used in the Clothing1M experiment follows the original (He et al., 2016).

Hyperparameter search. The CE, MAE, and FW baselines have no hyperparameters. For the DMI, we tuned the learning rate by setting the best value from the following list: $\{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$. The soft regularization approach of (11) has two hyperparameters: λ and β. We select λ from $[0.001, 0.01, 0.03, 0.1]$ and β from $[0.0, 0.01, 0.1, 1.0, 10.0]$. The objective of LIMIT instances has two terms: $\lambda H_{p,q}$ and $\beta\|\mu_t\|_2^2$. Consequently, we need only one hyperparameter instead of two. We choose to set $\lambda = 1$ and select β from $[0.0, 0.1, 0.3, 1.0, 3.0, 10.0, 30.0, 100.0]$. When sampling is enabled, we select σ_q from $[0.01, 0.03, 0.1, 0.3]$. In MNIST and CIFAR experiments, we trained all models for 400 epochs and terminated the training early when the best validation accuracy was not improved in the last 100 epochs. All models for Clothing1M were trained for 30 epochs.

C. Additional Results

Effectiveness of gradient norm penalty. In the main text we discussed that the proposed approach may overfit if the gradient predictor $q_\phi(\cdot \mid \mathbf{x}, g_{<t})$ overfits and proposed to penalize the L2 norm of predicted gradients as a simply remedy for this issue. To demonstrate the effectiveness of this regularization, we present the training and testing accuracy curves of LIMIT with varying values of β in Fig. A0. We see that increasing β decreases overfitting on the training set and usually results in better generalization.

Detecting incorrect samples. In the proposed approach, the auxiliary network q should not be able to distinguish correct and incorrect samples, unless it overfits. In fact, Fig. A1 shows that if we look at the norm of predicted gradients, examples with correct and incorrect labels are indistinguishable in easy cases (MNIST with 80% uniform noise and CIFAR-10 with 40% uniform noise) and have large overlap in harder cases (CIFAR-10 with 40% pair noise and CIFAR-100 with 40% uniform noise). Therefore, we hypothesize that the auxiliary network learns to utilize incorrect samples effectively by predicting "correct" gradients. This also hints that the distance between the predicted and cross-entropy gradients might be useful for detecting samples with incorrect or confusing labels. Fig. A2 confirms this intuition, demonstrating that this distance separates correct and incorrect samples perfectly in easy cases (MNIST with 80% uniform noise and CIFAR-10 with 40% uniform noise) and separates them well in harder cases (CIFAR-10 with 40% pair noise and CIFAR-100 with 40% uniform noise). If we interpret this distance as a score for classifying correctness of a label, we get 91.1% ROC AUC score



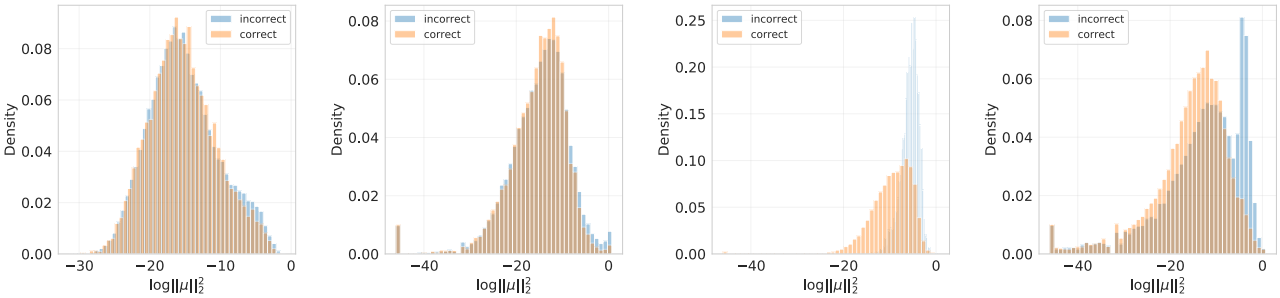
(a) Training performance of LIMIT$_{\mathcal{G}} - S$

(b) Training performance of LIMIT$_{\mathcal{L}} - S$

(c) Testing performance of LIMIT$_{\mathcal{G}} - S$

(d) Testing performance LIMIT$_{\mathcal{L}} - S$

Figure A0: Training and testing accuracies of "LIMIT$_{\mathcal{G}} - S$" and "LIMIT$_{\mathcal{L}} - S$" instances with varying values of β on MNIST with 80% uniform label noise. The curves are smoothed for better presentation.



(a) MNIST
80% uniform noise

(b) CIFAR-10
40% uniform noise

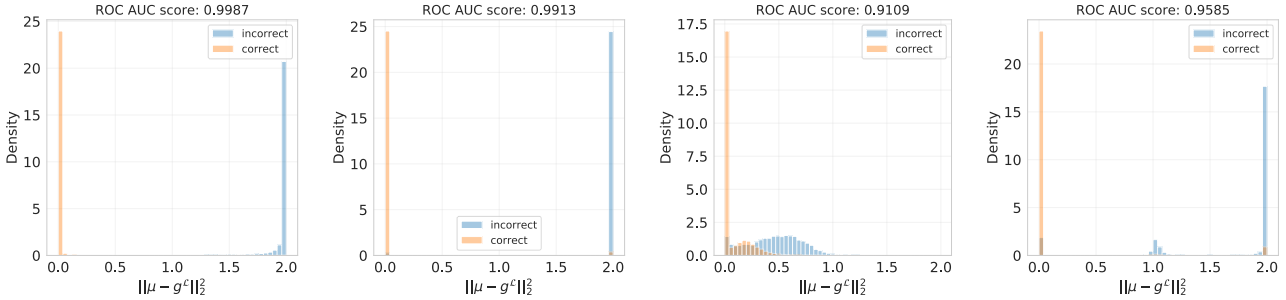
(c) CIFAR-10
40% pair noise

(d) CIFAR-100
40% uniform noise

Figure A1: Histograms of the norm of predicted gradients for examples with correct and incorrect labels. The gradient predictions are done using the best instances of LIMIT.

in the hardest case: CIFAR-10 with 40% pair noise, and more than 99% score in the easier cases.Motivated by this results, we use this analysis to detect samples with incorrect or confusing labels in the original MNIST, CIFAR-10, and Clothing1M datasets. We present a few incorrect/confusing labels for each class in Figures A3 and A4.

Quantitative results. Tables 4, 5, 6, and 7 present test accuracy comparisons on multiple corrupted versions of MNIST and CIFAR-10. The presented error bars are standard deviations. In case of MNIST, we compute them over 5 training/validation splits. In the case of CIFAR-10, due to high computational cost, we have only one run for each model and dataset pair. The standard deviations are computed by resampling the corresponding test sets 1000 times with replacement.



(a) MNIST
80% uniform noise

(b) CIFAR-10
40% uniform noise

(c) CIFAR-10
40% pair noise

(d) CIFAR-100
40% uniform noise

Figure A2: Histograms of the distance between predicted and actual gradient for examples with correct and incorrect labels. The gradient predictions are done using the best instances of LIMIT.

Method	$p = 0.0$			$p = 0.5$		
	$n = 10^3$	$n = 10^4$	All	$n = 10^3$	$n = 10^4$	All
CE	94.3 ± 0.5	98.4 ± 0.2	$\mathbf{99.2 \pm 0.0}$	71.8 ± 4.3	93.1 ± 0.6	97.2 ± 0.2
CE + GN	89.5 ± 0.8	95.4 ± 0.5	97.1 ± 0.5	70.5 ± 3.5	92.3 ± 0.7	97.4 ± 0.5
CE + LN	90.0 ± 0.5	95.3 ± 0.6	96.7 ± 0.7	66.8 ± 1.3	92.0 ± 1.5	97.6 ± 0.1
MAE	94.6 ± 0.5	98.3 ± 0.2	$\mathbf{99.1 \pm 0.1}$	75.6 ± 5.0	95.7 ± 0.5	98.1 ± 0.1
FW	93.6 ± 0.6	98.4 ± 0.1	$\mathbf{99.2 \pm 0.1}$	64.3 ± 9.1	91.6 ± 2.0	97.3 ± 0.3
DMI	94.5 ± 0.5	98.5 ± 0.1	$\mathbf{99.2 \pm 0.0}$	79.8 ± 2.9	95.7 ± 0.3	98.3 ± 0.1
Soft reg. (11)	$\mathbf{95.7 \pm 0.2}$	98.4 ± 0.1	$\mathbf{99.2 \pm 0.0}$	76.4 ± 2.4	95.7 ± 0.0	98.2 ± 0.1
LIMIT$_{\mathcal{G}}$ + S	$\mathbf{95.6 \pm 0.3}$	$\mathbf{98.6 \pm 0.1}$	$\mathbf{99.3 \pm 0.0}$	82.8 ± 4.6	97.0 ± 0.1	98.7 ± 0.1
LIMIT$_{\mathcal{L}}$ + S	94.8 ± 0.3	$\mathbf{98.6 \pm 0.2}$	$\mathbf{99.3 \pm 0.0}$	$\mathbf{88.7 \pm 3.8}$	$\mathbf{97.6 \pm 0.1}$	$\mathbf{98.9 \pm 0.0}$
LIMIT$_{\mathcal{G}}$ - S	$\mathbf{95.7 \pm 0.2}$	$\mathbf{98.7 \pm 0.1}$	$\mathbf{99.3 \pm 0.1}$	83.3 ± 2.3	97.1 ± 0.2	98.6 ± 0.1
LIMIT$_{\mathcal{L}}$ - S	95.0 ± 0.2	$\mathbf{98.7 \pm 0.1}$	$\mathbf{99.3 \pm 0.1}$	$\mathbf{88.2 \pm 2.9}$	$\mathbf{97.7 \pm 0.1}$	$\mathbf{99.0 \pm 0.1}$

Table 4: Test accuracy comparison on multiple versions of MNIST corrupted with uniform label noise.

Method	$p = 0.8$			$p = 0.89$		
	$n = 10^3$	$n = 10^4$	All	$n = 10^3$	$n = 10^4$	All
CE	27.0 ± 3.8	69.9 ± 2.6	87.2 ± 1.0	10.3 ± 1.6	13.4 ± 3.3	13.2 ± 1.8
CE + GN	25.9 ± 4.6	51.9 ± 10.5	85.3 ± 8.3	10.4 ± 4.5	10.2 ± 3.3	11.1 ± 0.4
CE + LN	30.2 ± 4.8	53.1 ± 6.4	74.5 ± 19.1	11.9 ± 3.9	8.8 ± 5.4	14.1 ± 4.3
MAE	25.1 ± 3.3	74.6 ± 2.7	93.2 ± 1.1	10.9 ± 1.4	12.1 ± 3.9	17.6 ± 8.1
FW	19.0 ± 4.1	61.2 ± 5.0	89.1 ± 2.1	8.7 ± 2.8	11.4 ± 1.4	12.3 ± 1.8
DMI	30.3 ± 5.1	79.0 ± 1.5	88.8 ± 0.9	10.5 ± 1.2	14.1 ± 5.1	12.5 ± 1.5
Soft reg. (11)	28.8 ± 2.2	67.0 ± 1.9	89.3 ± 0.6	10.3 ± 1.6	10.5 ± 0.8	12.7 ± 2.6
LIMIT$_{\mathcal{G}}$ + S	$\mathbf{35.9 \pm 6.3}$	80.6 ± 2.8	93.4 ± 0.5	10.0 ± 1.0	14.3 ± 5.4	13.1 ± 4.3
LIMIT$_{\mathcal{L}}$ + S	$\mathbf{35.6 \pm 3.2}$	$\mathbf{93.3 \pm 0.3}$	$\mathbf{97.6 \pm 0.3}$	10.1 ± 0.7	12.5 ± 2.1	$\mathbf{28.3 \pm 8.1}$
LIMIT$_{\mathcal{G}}$ - S	$\mathbf{37.1 \pm 5.4}$	82.0 ± 1.5	94.7 ± 0.6	9.9 ± 1.0	12.6 ± 0.3	16.0 ± 5.9
LIMIT$_{\mathcal{L}}$ - S	$\mathbf{35.9 \pm 4.3}$	$\mathbf{93.9 \pm 0.8}$	$\mathbf{97.7 \pm 0.2}$	11.1 ± 0.7	11.8 ± 1.0	$\mathbf{28.6 \pm 4.0}$

Table 5: Test accuracy comparison on multiple versions of MNIST corrupted with uniform label noise.

Method	$p = 0.0$	$p = 0.2$	$p = 0.4$	$p = 0.6$	$p = 0.8$
CE	92.7 ± 0.3	85.2 ± 0.4	81.0 ± 0.4	69.0 ± 0.5	38.8 ± 0.5
MAE	84.4 ± 0.4	85.4 ± 0.4	64.6 ± 0.5	15.4 ± 0.4	12.0 ± 0.3
FW	92.9 ± 0.3	86.2 ± 0.3	81.4 ± 0.4	69.7 ± 0.5	34.4 ± 0.5
DMI	93.0 ± 0.3	88.3 ± 0.3	85.0 ± 0.3	72.5 ± 0.4	38.9 ± 0.5
LIMIT$_{\mathcal{G}}$	$\mathbf{93.5 \pm 0.2}$	90.7 ± 0.3	86.6 ± 0.3	73.7 ± 0.4	38.7 ± 0.5
LIMIT$_{\mathcal{L}}$	93.1 ± 0.3	91.5 ± 0.3	88.2 ± 0.3	75.7 ± 0.4	35.8 ± 0.5
LIMIT$_{\mathcal{G}}$ + init.	$\mathbf{93.3 \pm 0.3}$	$\mathbf{92.4 \pm 0.3}$	$\mathbf{90.3 \pm 0.3}$	81.9 ± 0.4	$\mathbf{44.1 \pm 0.5}$
LIMIT$_{\mathcal{L}}$ + init.	$\mathbf{93.3 \pm 0.2}$	$\mathbf{92.2 \pm 0.3}$	$\mathbf{90.2 \pm 0.3}$	$\mathbf{82.9 \pm 0.4}$	$\mathbf{44.3 \pm 0.5}$

Table 6: Test accuracy comparison on CIFAR-10, corrupted with uniform label noise. The error bars are computed by bootstrapping the test set 1000 times.

Method	$p = 0.0$	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$
CE	92.7 ± 0.3	90.0 ± 0.3	88.1 ± 0.3	87.2 ± 0.3	81.8 ± 0.4
MAE	84.4 ± 0.4	88.6 ± 0.3	83.2 ± 0.4	72.1 ± 0.4	61.1 ± 0.5
FW	92.9 ± 0.3	90.1 ± 0.3	88.0 ± 0.3	86.8 ± 0.3	84.6 ± 0.3
DMI	93.0 ± 0.3	91.4 ± 0.3	90.6 ± 0.3	90.4 ± 0.3	$\mathbf{89.6 \pm 0.3}$
LIMIT$_{\mathcal{G}}$	$\mathbf{93.5 \pm 0.2}$	92.8 ± 0.3	91.3 ± 0.3	89.2 ± 0.3	86.0 ± 0.3
LIMIT$_{\mathcal{L}}$	93.1 ± 0.3	91.9 ± 0.3	91.1 ± 0.3	88.8 ± 0.3	84.2 ± 0.4
LIMIT$_{\mathcal{G}}$ + init.	$\mathbf{93.3 \pm 0.3}$	$\mathbf{93.3 \pm 0.3}$	$\mathbf{92.9 \pm 0.3}$	$\mathbf{90.8 \pm 0.3}$	88.3 ± 0.3
LIMIT$_{\mathcal{L}}$ + init.	$\mathbf{93.3 \pm 0.2}$	$\mathbf{93.0 \pm 0.2}$	92.3 ± 0.3	$\mathbf{91.1 \pm 0.3}$	$\mathbf{90.0 \pm 0.3}$

Table 7: Test accuracy comparison on CIFAR-10, corrupted with pair noise, described in Sec. 4.2. The error bars are computed by bootstrapping the test set 1000 times.

(a) MNIST

(b) CIFAR-10

Figure A3: Most confusing 8 labels per class in the MNIST (on the left) and CIFAR-10 (on the right) datasets, according to the distance between predicted and cross-entropy gradients. The gradient predictions are done using the best instances of LIMIT.

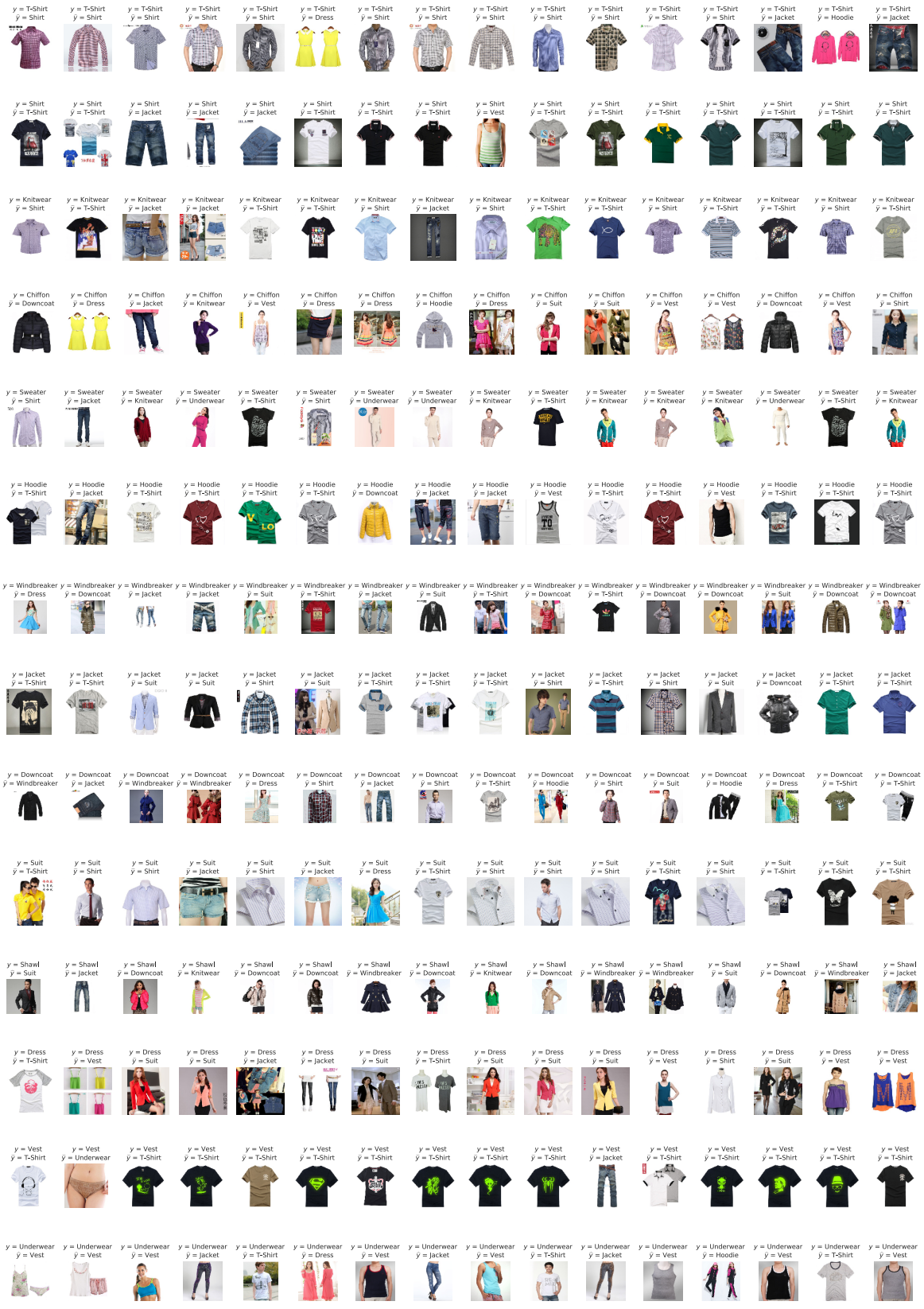


Figure A4: Most confusing 16 labels per class in the Clothing1M dataset, according to the distance between predicted and cross-entropy gradients. The gradient predictions are done using the best instance of LIMIT.