Adaptive Robust Learning using Latent Bernoulli Variables

Aleksandr Karakulev [1] **Dave Zachariah**[*1] **Prashant Singh**[*1 2]

Abstract

We present an adaptive approach for robust learning from corrupted training sets. We identify corrupted and non-corrupted samples with latent Bernoulli variables and thus formulate the learning problem as maximization of the likelihood where latent variables are marginalized. The resulting problem is solved via variational inference, using an efficient Expectation-Maximization based method. The proposed approach improves over the state-of-the-art by automatically inferring the corruption level, while adding minimal computational overhead. We demonstrate our robust learning method and its parameter-free nature on a wide variety of machine learning tasks including online learning and deep learning where it adapts to different levels of noise and maintains high prediction accuracy.

1. Introduction

Several statistical learning problems are formulated as estimation of parameters $\boldsymbol{\theta} \in \Theta$ of a probabilistic model by maximizing its likelihood function $\prod_{i=1}^{n} p(\boldsymbol{z}_i|\boldsymbol{\theta})$ given n independent observations $Z = \{\boldsymbol{z}_i\}_{i=1}^{n}$, $\boldsymbol{z}_i \sim p(\boldsymbol{z})$. By defining the loss function as the negative log-likelihood $\ell_{\boldsymbol{\theta}}(\boldsymbol{z}) = -\ln p(\boldsymbol{z}|\boldsymbol{\theta})$, one can equivalently solve

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}\in\Theta} \prod_{i=1}^{n} p(\boldsymbol{z}_i|\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}\in\Theta} \sum_{i=1}^{n} \ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i). \quad (1)$$

However, real-world data is often corrupted: samples arise from the true distribution $p(\boldsymbol{z})$ and a corrupting source $q(\boldsymbol{z})$,

$$\boldsymbol{z}_i \sim (1-\varepsilon)p(\boldsymbol{z}) + \varepsilon q(\boldsymbol{z}), \quad (2)$$

leading to a suboptimal solution $\boldsymbol{\theta}_{\mathrm{ML}}$. In this contaminated mixture, corrupted samples $\boldsymbol{z} \sim q(\boldsymbol{z})$ may result from inaccurate measurements, errors or oversight in data acquisition or labeling, or even malicious attacks. Further, the corrupting distribution $q(\boldsymbol{z})$ is typically unknown, and the level of corruption ε, difficult to determine. Equation (2) used herein follows the well-known Huber contamination model from the robust statistics literature (Huber, 2011; Maronna et al., 2019).

Existing approaches require an estimation of the noise structure or the corruption level ε. This imposes restrictions on using these methods in settings where such pre-processing becomes impractical, e.g., in online learning, wherein the assumption about ε needs to be optimized in continuously arriving data. In Figure 1, we show an example: the accuracy of binary classification trained on data that is continuously collected from the HAR dataset (Helou, 2023), and subsequently corrupted with varying number of randomly flipped labels in each batch (details in Section 4.2).

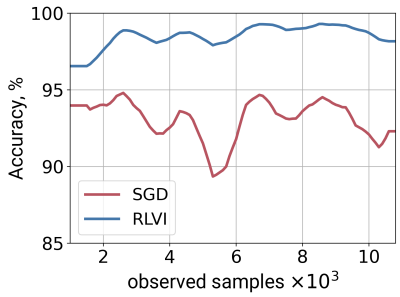


Figure 1: Classification of online streaming data with varying number of corrupted labels. Adaptive nature of our approach (RLVI) allows for automatic identification of outliers when learning from batches of data with different ε. Our method is robust and thus has higher accuracy than the standard stochastic optimization of the likelihood (SGD).

Contribution. This paper presents a principled approach for robust learning from corrupted data that is

- widely applicable given any likelihood function,
- robust against a wide class of contamination sources,
- adaptable and tuning-parameter free,
- scalable for large data sets and deep learning models.

[*]Co-senior authors [1]Uppsala University, Sweden [2]Science for Life Laboratory, Sweden. Correspondence to: Aleksandr Karakulev <aleksandr.karakulev@it.uu.se>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

Robust Learning via Variational Inference. The proposed approach, denoted as RLVI, employs latent Bernoulli variables to identify corrupted and non-corrupted training samples. The distribution of the latent variables is characterized from training data by maximizing the variational lower bound of marginal likelihood (variational inference) (Murphy, 2023). This allows detection of corrupted samples and computation of the optimal corruption level ε automatically, subsequently learning the model $\boldsymbol{\theta}$ in a robust way. The problem formulation entailing RLVI is theoretically simple and also leads to an efficient and straightforward implementation.

Related work. While some approaches address learning from corrupted data with a special loss function, e.g., the Huber loss (Huber, 1992), others construct certain criteria to separate samples from the true distribution and the corrupted ones (Bhatia et al., 2015; 2017). Recently, new general approaches, such as SEVER (Diakonikolas et al., 2019) and Robust Risk Minimization (RRM) (Osama et al., 2020), were introduced. Both methods are not model-specific. SEVER is a general gradient-based learning method that penalizes the gradient components of the loss function corresponding to outliers. RRM involves obtaining sample weights from the constraint on the entropy of a weighted empirical distribution and minimizing the modified empirical risk. Robust Risk Minimization does not require calibrating multiple hyperparameters, unlike SEVER. Despite not requiring the exact value of ε, both approaches are reliant on its upper-bound estimate $\widetilde{\varepsilon} \geq \varepsilon$. A very relevant work is (Wang et al., 2017), where the authors introduce latent variables $\boldsymbol{w}$ and raise likelihood terms to these variables to account for the departure from model's assumptions. The paper suggests to impose a prior distribution for $\boldsymbol{w}$ that suits the problem at hand and infer $\boldsymbol{w}$ together with the model parameters. Thus, being quite general, this framework provides much freedom to users while leaving inference of $\boldsymbol{w}$ as a computationally challenging task solved with probabilistic programming. In this paper, we rather consider a specific case and define Bernoulli latent variables based on the Huber model (2), which enables efficient computation and does not require to specify any hyperparameters for a prior distribution.

In the recent past, learning from corrupted data has also received attention in the field of deep learning, and several approaches have been proposed to train a neural network in a robust way. For instance, one can use the correction of the regular loss function (Zhang & Sabuncu, 2018; Patrini et al., 2017). Alternatively, some approaches directly estimate the noise transition matrix (Goldberger & Ben-Reuven, 2016; Patrini et al., 2017). Others utilize an additional neural network to identify and prevent overfitting of the initial model, as in (Jiang et al., 2018), and to obtain a better accuracy by exchanging the information between two simultaneously trained networks, as in (Han et al., 2018) and (Wei et al., 2020). Approach from (Ren et al., 2018) is based on re-weighting the loss terms for each observation, wherein sample weights are treated as hyperparameters optimized with additional gradient computations. Another option is to combine techniques that are specific to deep learning: early-stopping (Xia et al., 2020) or dropout (Xu et al., 2023), with a criterion that aims to eliminate corrupted samples. However, training a neural network in the regular context where the contamination issue is not addressed, already requires certain hyperparameters to improve generalization, e.g., learning rate, batch size, number of layers, etc. The aforementioned approaches introduce additional parameters to combat the problem of corrupted labels, including ε, which makes their performance dependent on the efficacy of these parameters and the underlying assumptions.

2. Problem Formulation

Central to our approach is the introduction of a latent variable t_i for each observation $\boldsymbol{z}_i \in Z$ such that

$$t_i = \begin{cases} 1, \, \boldsymbol{z}_i \sim p(\boldsymbol{z}), \\ 0, \, \boldsymbol{z}_i \sim q(\boldsymbol{z}). \end{cases} \tag{3}$$

Knowing the values of these variables will allow avoiding minimization of the losses on corrupted samples in the dataset in (1) by simply dropping the corresponding terms from the sum. In other words, latent variables enable us to define a likelihood function with respect to the non-corrupted data, such that

$$p(Z|\boldsymbol{t}, \boldsymbol{\theta}) = \prod_{i=1}^{n} p(\boldsymbol{z}_i|\boldsymbol{\theta})^{t_i}, \tag{4}$$

where $\boldsymbol{t} = (t_1, t_2, \ldots, t_n)$. Optimization with respect to this likelihood function implies a combinatorial search over the latent variables $\boldsymbol{t}$, which is computationally infeasible even for moderate n.

We observe, however, that the contamination model (2) implies that each sample has a probability ε of being corrupted. Thus we have the following distribution over $\boldsymbol{t}$,

$$p(\boldsymbol{t}|\varepsilon) = \prod_{i=1}^{n} (1-\varepsilon)^{t_i} \varepsilon^{1-t_i}. \tag{5}$$

This prior information enables us to marginalize out the latent variables and obtain the *marginalized* likelihood,

$$p(Z|\boldsymbol{\theta}, \varepsilon) = \sum_{\boldsymbol{t}} p(Z|\boldsymbol{t}, \boldsymbol{\theta}) p(\boldsymbol{t}|\varepsilon). \tag{6}$$

Then the maximum marginal likelihood solution,

$$\widehat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \max_{\varepsilon} p(Z|\boldsymbol{\theta}, \varepsilon), \tag{7}$$

addresses the problem of robust learning of $\boldsymbol{\theta}$ while obviating the need for specifying ε or a combinatorial search over $\boldsymbol{t}$. We now turn to developing a method to approximately solve (7) in an efficient manner.

3. Method

We begin by introducing a variational bound on the marginal likelihood in (7) and then turn to implementation-specific aspects of the derived method.

Variational inference. To solve the formulated optimization problem, we need a tractable way to compute the sum over $\boldsymbol{t}$ in (6). Using Bayes' rule, we express the marginal likelihood as

$$p(Z|\boldsymbol{\theta}, \varepsilon) = \frac{p(Z|\boldsymbol{t}, \boldsymbol{\theta}) p(\boldsymbol{t}|\varepsilon)}{p(\boldsymbol{t}|Z, \boldsymbol{\theta}, \varepsilon)}, \tag{8}$$

where the denominator is the posterior distribution of latent variables $\boldsymbol{t}$ given the data Z, specified model $\boldsymbol{\theta}$, and the corruption level ε. As this posterior is intractable, we consider its variational approximation denoted $r(\boldsymbol{t}|\boldsymbol{\pi})$. Specifically, we use a distribution of n independent Bernoulli variables with probabilities $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$,

$$r(\boldsymbol{t}|\boldsymbol{\pi}) = \prod_{i=1}^{n} \pi_i^{t_i} (1 - \pi_i)^{1 - t_i}. \tag{9}$$

Therefore, instead of maximizing the marginal likelihood function $p(Z|\boldsymbol{\theta}, \varepsilon)$ directly, we apply the variational inference framework and optimize the so-called evidence lower-bound (ELBO) (Murphy, 2023). This lower bound has the following general form,

$$\ln p(Z|\boldsymbol{\theta}, \varepsilon) \geq \underbrace{\mathbb{E}_{r(\boldsymbol{t}|\boldsymbol{\pi})}\Big[\ln p(Z|\boldsymbol{\theta}, \boldsymbol{t})\Big] - \mathrm{KL}\Big[r(\boldsymbol{t}|\boldsymbol{\pi}) \Big|\Big| p(\boldsymbol{t}|\varepsilon)\Big]}_{\mathrm{ELBO}(\boldsymbol{\theta}, \boldsymbol{\pi}, \varepsilon)}, \tag{10}$$

where the first term is the expected value of the log-likelihood over latent variables and the second term is the Kullback–Leibler divergence between the variational approximation and the prior. The first ELBO term can be rewritten using the loss function $\ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i) = -\ln p(\boldsymbol{z}_i|\boldsymbol{\theta})$,

$$\mathbb{E}_{r(\boldsymbol{t}|\boldsymbol{\pi})}\Big[-\sum_{i=1}^{n} t_i \ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i)\Big] = -\sum_{i=1}^{n} \pi_i \ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i), \tag{11}$$

where the equality follows from (9). This term corresponds to an average loss with non-uniform sample weights, with π_i representing the probability of sample i being non-corrupted.

The second ELBO term consists of the closed form,

$$\mathrm{KL}\Big[r \,\Big|\Big|\, p\Big] = \sum_{i=1}^{n} \pi_i \ln \frac{\pi_i}{1 - \varepsilon} + (1 - \pi_i) \ln \frac{1 - \pi_i}{\varepsilon}. \tag{12}$$

Note that corruption level does not appear in the first term of the ELBO and we can optimize ε directly and independently of the model $\boldsymbol{\theta}$:

$$\varepsilon = \arg\max \mathrm{ELBO}(\boldsymbol{\theta}, \boldsymbol{\pi}, \varepsilon) = 1 - \frac{1}{n} \sum_{i=1}^{n} \pi_i. \tag{13}$$

That is, variational inference framework resolves an unknown hyperparameter by explicitly optimizing the corruption level. Moreover, we get an intuitively satisfying result that parameters π_i defining the probability of each sample being non-corrupted, should sum up to the number of non-corrupted samples:

$$n(1 - \varepsilon) = \sum_{i=1}^{n} \pi_i. \tag{14}$$

Hence we arrive at the following objective – the negative ELBO, expressed as

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{i=1}^{n} \pi_i \ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i) + \pi_i \ln \frac{\pi_i}{\langle \boldsymbol{\pi} \rangle} + (1 - \pi_i) \ln \frac{1 - \pi_i}{1 - \langle \boldsymbol{\pi} \rangle}, \tag{15}$$

where $\langle \boldsymbol{\pi} \rangle := \sum_{i=1}^{n} \pi_i \,/\, n$ is the average of Bernoulli probabilities.

Consequently, the objective in (7) is replaced by the optimum of its variational bound, i.e.,

$$\boxed{\boldsymbol{\theta}_{\mathrm{RLVI}} = \arg\min_{\boldsymbol{\theta} \in \Theta} \min_{\boldsymbol{\pi} \in (0;1)^n} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi}).} \tag{16}$$

Numerical optimization. The resulting optimization problem is solved using the block-wise Algorithm 1, which follows the general Expectation-Maximization (EM) scheme (Bishop & Nasrabadi, 2006). The E-step consists of minimizing $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi})$ in $\boldsymbol{\pi}$ for fixed parameters $\boldsymbol{\theta}$. In the M-step, the inferred probabilities are used to maximize the re-scaled log-likelihood, $-\sum_{i=1}^{n} \pi_i \ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i)$. As an example, in case of linear regression the latter step solves a weighted least-squares problem. For a classification task, we minimize the cross-entropy loss corrected with sample weights.

Note that the E-step can be performed efficiently, as the objective $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi})$ is convex in $\boldsymbol{\pi}$. Therefore, to find the optimal parameters $\boldsymbol{\pi}$ for a fixed model, the derivative of $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi})$ is equated to zero w.r.t. π_j for all $j = 1, \ldots, n$,

$$\frac{\partial \mathcal{L}}{\partial \pi_j} = 0 \iff \pi_j = \left(1 + \frac{1 - \langle \boldsymbol{\pi} \rangle}{\langle \boldsymbol{\pi} \rangle} e^{\ell_{\boldsymbol{\theta}}(\boldsymbol{z}_j)}\right)^{-1}. \tag{17}$$

This is a system of nonlinear equations for $\boldsymbol{\pi}$, which we solve with the fixed-point iterations,

$$\pi_j^{\mathrm{new}} = \left(1 + \frac{1 - \langle \boldsymbol{\pi} \rangle^{\mathrm{old}}}{\langle \boldsymbol{\pi} \rangle^{\mathrm{old}}} e^{\ell_{\boldsymbol{\theta}}(\boldsymbol{z}_j)}\right)^{-1}. \tag{18}$$

In (18), for the mean weight, we use the value from the previous fixed-point iteration and compute π_j^{new} independently with simple vector operations that scale well for large data. A proof for $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi})$ being convex in $\boldsymbol{\pi}$ and iterations (18) converging to a stationary point is provided in the Appendix.

Algorithm 1 RLVI: robust learning from corrupted data

Input: data $Z = \{\boldsymbol{z}_i\}_{i=1}^n$
$\boldsymbol{\theta}^0 \leftarrow \arg\min_{\boldsymbol{\theta}\in\Theta} \sum_{i=1}^n \ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i)$ // $\pi_i^0 = 1$
for $k = 1, 2, \ldots$
  Evaluate $\ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i)$ for each $\boldsymbol{z}_i \in Z$ using $\boldsymbol{\theta}^{k-1}$
  $\boldsymbol{\pi}^k \leftarrow$ fixed-point iterations (18) // *E-step*
  $\boldsymbol{\theta}^k \leftarrow \arg\min_{\boldsymbol{\theta}\in\Theta} \sum_{i=1}^n \pi_i^k \ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i)$ // *M-step*
  if $\|\boldsymbol{\theta}^k - \boldsymbol{\theta}^{k-1}\| \leq tolerance$
    return $\boldsymbol{\theta}^k$

Stochastic approximation. The formulated objective $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi})$ alleviates the need to know or assess ε, which is particularly useful when ε is not fixed, such as in the online learning setting. Another useful property of this variational bound is its structure: with respect to $\boldsymbol{\theta}$, the function consists of n independent terms, just as the standard likelihood function, which can be employed by stochastic optimization of $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi})$. If one does not have access to the full dataset but only to its batches, the M-step in Algorithm 1 can be replaced with a stochastic gradient descent (SGD) update

$$\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1} - \alpha \sum_{i=1}^{b} \pi_i^k \nabla \ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i), \tag{19}$$

where the loss function and, therefore, the gradient components for each batch of b samples $\boldsymbol{z}_i$ are re-weighted with π_i^k, $i = 1, \ldots, b$. The latter are computed using the corresponding loss values $\ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i)$ at the preceding E-step with (18). Batch minimization of $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi})$, in effect, corresponds to a stochastic variant of the EM algorithm – see, e.g., stepwise EM in (Murphy, 2023).

Truncation as a form of regularization. Since the objective in RLVI is suitable for stochastic optimization, our robust learning approach can also be used in deep learning, where SGD-like approaches are dominant. However, neural networks are overparameterized models and thus are prone to overfitting, which in theory (and in practice: see Figure 5) hinders the performance of RLVI. Indeed, if we substitute zero loss for all training samples into the stationary point condition (17), we find the corresponding minimum for all $i = 1, \ldots, n$,

$$\pi_i^\star = \left(1 + \frac{\varepsilon}{1-\varepsilon} e^{\ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i)}\right)^{-1} \Bigg|_{\ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i)=0} = 1 - \varepsilon. \tag{20}$$

That is, when overfitting commences, the marginal likelihood approach treats all samples as non-corrupted since overparameterized model is capable of minimizing loss to zero on both 'clean' and corrupted samples. Nevertheless, as is generally the case, to prevent overfitting, one can use regularization of the loss function. In this work, we introduce regularization to the RLVI algorithm, making the algorithm effective in the overparameterized regime as well. Namely, samples that have a low probability of being non-corrupt are eliminated from SGD updates:

$$\pi_i < \tau \implies \pi_i \leftarrow 0. \tag{21}$$

Furthermore, we define the threshold τ based on the following criterion: maximize the number of samples to be used for learning (maximize τ) subject to a bounded type II error (number of corrupted samples treated as 'clean'). Hence, $\tau = \max\{\pi_1, \pi_2, \ldots, \pi_n\}$, such that

$$\frac{\mathbb{E}_r\Big[\#\text{False Clean}\Big]}{\mathbb{E}_r\Big[\#\text{Corrupted}\Big]} = \frac{\sum_{i=1}^n (1-\pi_i)\mathbb{1}[\pi_i \geq \tau]}{\sum_{i=1}^n (1-\pi_i)} \leq 0.05. \tag{22}$$

In this criterion, the admissable type II error is common for statistical hypothesis testing and equals 5%. Also note how the obtained posterior approximation is employed: the numerator is the expected number of corrupted samples considered as 'clean', and the denominator is the expected total number of corrupted samples, based on $r(\boldsymbol{t}|\boldsymbol{\pi})$.

The resulting variant of RLVI, to be used for overparameterized models, is listed as Algorithm 2. It implements RLVI as stochastic gradient optimization of neural network parameters $\boldsymbol{\theta}$. Parameters π_i are updated at the end of each epoch using efficient iterations (18). To prevent model's overfitting to corrupted samples, the algorithm eliminates gradient terms corresponding to low π_i. Threshold for truncation is re-computed across epochs as a non-decreasing value from the type II error criterion (22). Note that Bernoulli probabilites are updated once at each epoch, which makes solution for $\boldsymbol{\pi}$ less dependent on the batch size in this case.

4. Experiments

The proposed RLVI method is compared to existing approaches in three problem settings: standard parameter estimation, online learning, and deep learning[1].

4.1. Benchmark on standard learning problems

First, we demonstrate that our method is applicable to different maximum likelihood problems and achieves higher

[1]Implementation of RLVI and our experiments are available at https://github.com/akarakulev/rlvi.

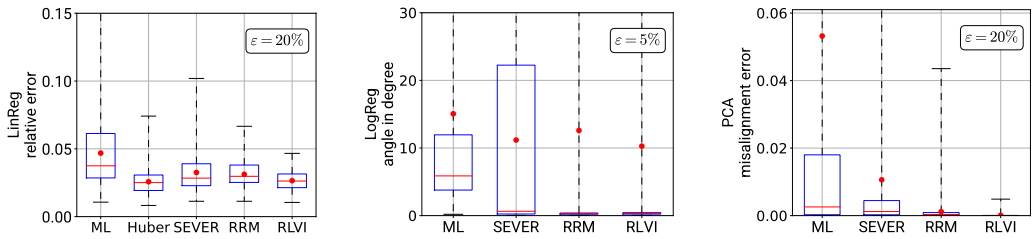


Figure 2: Box plots of relative errors for a fixed value of corruption level ε. *Left*. Linear regression: relative errors $\|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\|_2 / \|\boldsymbol{\theta}^\star\|_2$. *Middle*. Logistic regression: angle in degrees between the true separating hyperplane $\boldsymbol{\theta}^\star$ and estimates $\widehat{\boldsymbol{\theta}}$. *Right*. PCA: misalignment errors $1 - |\cos(\widehat{\boldsymbol{\theta}}^\top \boldsymbol{\theta}^\star)|$ for the subspace spanned by the first principal component. Each box spans the 25th to 75th quantiles; red dots depict the means. For all plots, 100 Monte Carlo runs are used.

model accuracy over contemporary alternatives. Thereto, we reproduce the experiments from (Osama et al., 2020) comparing various algorithms for robust learning on three problems: linear regression, logistic regression, and dimensionality reduction using principal component analysis (PCA). Since each problem can be formulated as likelihood maximization, we define the corresponding loss functions as the negative log-likelihood. In the experiments, synthetic data (n samples) is generated from the mixture of $p(\boldsymbol{z})$ and $q(\boldsymbol{z})$ with a fixed ratio of corrupted samples ε using the specified model $\boldsymbol{\theta}^\star$. The optimal model $\widehat{\boldsymbol{\theta}}$ is estimated using different algorithms: standard ML, HUBER (Zoubir et al., 2018), SEVER (Diakonikolas et al., 2019), RRM (Osama et al., 2020), and RLVI. The reader is referred to (Osama et al., 2020) for more details on the three test problems, including specifics of distributions used as true and corrupted. Subsequently, we perform 100 Monte Carlo runs and plot the statistics for corresponding errors with boxplots in Figure 2. For linear regression the results, shown in Figure 3, also include average relative error for ε varying in $[0; 0.4]$. Again, 100 Monte Carlo runs are used for each fixed ε.

Problem	dimension	n	ε	$\widetilde{\varepsilon}$
LinReg	$\boldsymbol{\theta} \in \mathbb{R}^{10}$	40	0.2	0.4
LogReg	$\boldsymbol{\theta} \in \mathbb{R}^{3}$	100	0.05	0.3
PCA	$\boldsymbol{\theta} \in \{\mathbb{R}^2 : \|\boldsymbol{\theta}\| = 1\}$	40	0.2	0.4

Table 1: Experiments reproduced from (Osama et al., 2020): $\boldsymbol{\theta}$ is a parametric model learned from synthetic data (n samples in total, εn are corrupted); $\widetilde{\varepsilon} \geq \varepsilon$ is the upper-bound used for HUBER, SEVER, and RRM.

Note that using the unbounded likelihood for defining $\ell_{\boldsymbol{\theta}}(\boldsymbol{z})$ within RLVI can lead to a degenerate solution and thus might require additional regularization – see an example with a covariance estimation problem in the Appendix.

From Figures 2 and 3, one can see that RLVI achieves better average accuracy and tighter confidence intervals than the competing methods.

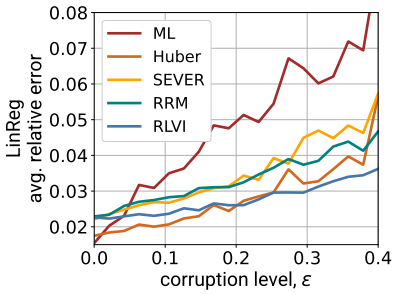


Figure 3: Linear regression. Average relative error versus varying corruption level ε; 100 Monte Carlo runs are used.

4.2. Online learning

To further evaluate RLVI's adaptivity, we apply it to online learning – the setting where ε changes dynamically. Online learning is used when dealing with, for example, signals from remote sensors, user clicks on a website, daily weather conditions, etc. In such cases, data is being collected sequentially and the model is incrementally learned from each new set of observations. This allows continuous refinement of the model and out-of-core inference when the dataset is too large to be stored and handled entirely.

As described in Section 3, the objective $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi})$ in RLVI allows for an incremental learning scheme by replacing the M-step of EM algorithm with an update of stochastic gradient descent: $\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1} - \alpha \sum_{i=1}^{b} \pi_i^k \nabla \ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i)$. Here, parameters $\boldsymbol{\pi}^k$ are computed at the E-step with (18). The computational overhead over the standard stochastic likelihood maximization is not significant: the fixed-point algorithm only involves $O(b)$ vectorized operations at each SGD step.

We consider the Human Activity Recognition dataset from (Helou, 2023) containing $24,075$ measurements from smartphone sensors. Each measurement $\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i)$ consists of $\boldsymbol{x}_i \in \mathbb{R}^{60}$ features extracted from accelerometers during different human activities: Sitting, Standing, Walking, Run-

ning, and Dancing. We perform binary classification and towards that end, partition the five initial labels into two: resting state ($y = 0$) and active state ($y = 1$). To simulate a data stream, at each iteration we retrieve data in batches of size $2b$, where $b = 100$ samples are used for training and another $b = 100$ samples serve performance evaluation. To introduce noise, we corrupt the data in each training batch by randomly flipping ε percent of positive labels. Moreover, the corruption level ε changes in each iteration. Since we are testing the robustness of RLVI against corruption, we focus specifically on variation of noise – not on the gradual change of $\boldsymbol{\theta}$ (concept drift) or other possible challenges related more to the method to learn $\boldsymbol{\theta}$ incrementally. For each batch, ε is sampled from a linearly transformed Beta distribution (called PERT (Johnson et al., 1995)) that is simple to parameterize with three values, so that the samples are within the interval $[\varepsilon_{\text{min}}, \varepsilon_{\text{max}}]$ with the mode $\varepsilon_{\text{mode}}$. To simulate a typical case, we set $\varepsilon_{\text{min}} = 0$, $\varepsilon_{\text{max}} = 0.3$, and $\varepsilon_{\text{mode}} = 0.1$. Thus, the ratio of corrupted labels in each batch varies according to the distribution in Figure 4.

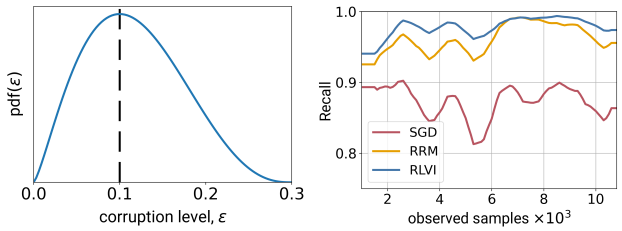


Figure 4: Online classification. *Left*. Distribution of corruption level across batches of streaming data. *Right*. Recall (true positive rate) on left-out data versus total number of observed samples (smoothed with a moving average filter).

Since RRM is also based on block-wise optimization, it can similarly be implemented as incremental learning, where inference for $\boldsymbol{\theta}$ consists of anti-gradient steps minimizing the modified empirical risk for each batch. Hence, the experiments compare three versions of online classification: the standard stochastic maximization of the likelihood (SGD), incremental minimization of the risk function defined in RRM (Osama et al., 2020) with a threshold for ε set to $\varepsilon_{\text{max}} = 0.3$, and the proposed approach, RLVI.

Classification performance is evaluated with recall (true positive rate) computed for 100 test samples in each iteration. Figure 4 shows the evolution of recall smoothed with a moving average filter with a window of 10 batches (also, see the smoothed accuracy curve in Figure 1). It can be seen that standard SGD clearly suffers from label corruption, with RRM being affected to a lesser extent. RLVI attains consistently higher accuracy and recall values as it does not depend on a global estimate of the noise magnitude, and robustly evaluates it from incoming data.

4.3. Overparameterized model

We now extend RLVI to learning an overparameterized model and consider image classification using a convolutional neural network when training labels are corrupted.

Existing approaches. State-of-the-art performance is achieved in this setting by methods that identify and distill corrupted samples based on some criterion. The algorithm Co-teaching (Han et al., 2018) is based on training two neural networks in parallel and learning only on the samples that attain a small loss for both models. The JoCoR approach (Wei et al., 2020) also trains two models but aims to reduce the diversity between their predictions. Due to simultaneous training of two models, Co-teaching and JoCoR effectively double the computational time. CDR (Xia et al., 2020) trains one model – it employs weight decay to diminish the impact of network parameters that overfit to corrupted samples, and early-stopping using a validation set. The recent algorithm USDNL (Xu et al., 2023) estimates prediction uncertainty for training samples using dropout, thus identifying the samples with corrupted labels. Each of the above methods combines its own criterion for corrupted samples with a schedule to gradually consider fewer samples and thus account for model overfitting occurring in later epochs. The schedule is a non-decreasing function defined with the corruption level ε (assumed to be known in advance) so that, by the end of training, only $(1 - \varepsilon)$ ratio of the training set is being used. The algorithm BARE from (Patel & Sastry, 2023) aims to robustly train neural networks for classification independently of ε. It removes samples from the loss function using batch statistics, assuming that the class conditional noise has a special structure.

Regularization. As discussed in Section 3, highly overparameterized models overfit to all samples, including the corrupted ones. To prevent overfitting within RLVI, we use hard truncation (21), where truncation boundary τ is defined by the type II error criterion (22). Furthermore, as studied empirically (Jiang et al., 2018; Nguyen et al., 2019; Xia et al., 2020), neural networks overfit in later epochs. Thus, regularization (21) is applied after the model starts to overfit, which can be identified by a decrease in prediction accuracy on a contaminated validation set.

Figure 5 presents an example of image classification on CIFAR10 with randomly flipped labels. It illustrates how the introduced regularization functions in practice: if no regularization is used, the proportion of identified corrupted observations decreases while model gradually fits $q(\boldsymbol{z})$ despite marginal likelihood formulation, thus attaining lower test accuracy. But with regularization, differentiation of corrupted and non-corrupted data points according to π_i is more effective during all iterations, which leads to better generalization. In the Appendix, we provide similar plots for various types and levels of noise.

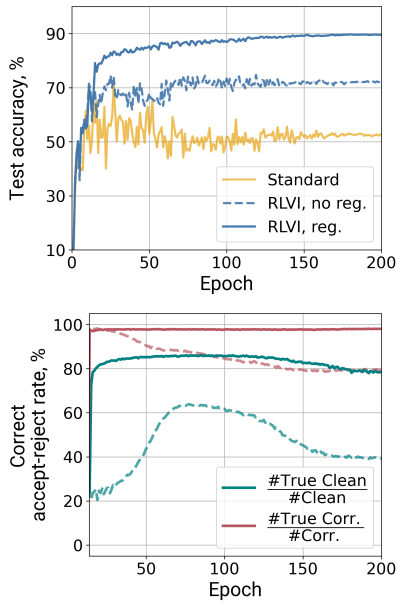


Figure 5: Image classification on CIFAR10 corrupted with synthetic noise (pairflip, $\varepsilon = 45\%$). *Top*. Accuracy on the clean testing set for standard SGD and RLVI. *Bottom*. Percentage of corrupted and non-corrupted samples correctly identified with the decision boundary $\pi_i < \tau$. In both plots, the dashed line corresponds to RLVI with no regularization – solid line indicates that truncation ($\pi_i < \tau \implies \pi_i \leftarrow 0$) is used for the terms in anti-gradient updates. Threshold τ is computed from (22). Regularization based on the bounded type II error makes differentiation of corrupted samples more effective, ultimately improving test accuracy for the overparameterized setting.

Synthetic corruption. To test the algorithm, we conduct the experiments in image classification using convolutional neural networks and cross-entropy loss as $\ell_{\boldsymbol{\theta}}$. The datasets being used are MNIST (LeCun, 1998), that contains handwritten digits, and CIFAR10 and CIFAR100 (Krizhevsky, 2009) – both containing the same images that are classified into ten and one hundred categories respectively. Subsequently, the training labels in these datasets are corrupted with four types of synthetic noise: symmetric, asymmetric, and pairflip (class-dependent) and instance (feature-dependent). These noise types have been commonly used in previous works (Han et al., 2018; Wei et al., 2020; Xia et al., 2020; Xu et al., 2023). Additionally, we perform the experiment in which data contains both the corrupted labels and the out-of-distribution samples. To this end, we consider the dataset CIFAR80N-O (Yao et al., 2021) which is obtained from CIFAR100 as follows: the last 20 classes in CIFAR100 are regarded as out-of-distribution images, and images from the remaining 80 classes are subsequently corrupted by one of the class-dependent synthetic noise types (pairflip, symmetric, and asymmetric). Thereto, we demonstrate that, owing to the generality of Huber contamination model (2), RLVI can be succesfully applied to learning in the presence of noise of arbitrary structure. In these experiments, we employ commonly used hyperparameter settings found in literature specific to each dataset and model architecture. These settings can be found in Table 4 of the Appendix.

Algorithm 2 RLVI: robust training of neural networks

Input: training set $Z_{tr} = \{\boldsymbol{z}_i\}_{i=1}^n$, noisy validation set Z_{val}, learning rate α, batch size b
$\boldsymbol{\theta} \leftarrow$ initialize neural network parameters
$\pi_i \leftarrow 1,\ i = 1, \dots, n$
$L \leftarrow$ empty array of size n // to store $\ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i), \boldsymbol{z}_i \in Z_{tr}$
$overfit \leftarrow$ False
$\tau \leftarrow 0$
for $epoch = 1, 2, \dots, n_{\text{epochs}}$
  for $\mathcal{I}_b \sim U\{1, \dots, n\}$ // for a batch of b indices
    $L_i \leftarrow \ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i),\ i \in \mathcal{I}_b$ // store loss value
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \sum_{i \in \mathcal{I}_b} \pi_i \nabla \ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i)$
  $\boldsymbol{\pi} \leftarrow$ fixed-point (18) using loss values in L
  if $overfit$
    $\tau^\star \leftarrow \max\{\pi_1, \dots, \pi_n\}$, s.t. error bound (22)
    $\tau \leftarrow \max(\tau, \tau^\star)$ // can only increase
    $\pi_i \leftarrow \pi_i$ **if** $\pi_i \geq \tau$ **else** 0 // regularization
  else if accuracy on Z_{val} dropped
    $overfit \leftarrow$ True
Output: $\boldsymbol{\theta}$

We compare the attained classification accuracy on the test set with the standard likelihood maximization approach and recently proposed alternative methods: Co-teaching, JoCoR, CDR, USDNL, and BARE. For CDR and RLVI, 10% of the training data is used as a validation set: in RLVI, we apply regularization (21) after validation accuracy at the current epoch becomes less than the average of its two previous values. In contrast, CDR uses a validation set for early-stopping: optimization concludes if the validation accuracy exceeds some specified threshold. The latter implies that for CDR we report the test accuracy corresponding to the lowest validation loss. For all the alternative methods, we use their default hyperparameters related to robust learning, including the same schedule for the ratio of considered samples during epochs, as defined in (Han et al., 2018), deduced from the true noise level employed. Also, since USDNL is based on uncertainty estimation using dropout, for USDNL specifically, we used variants of the corresponding neural nets with dropout layers, where dropout rate was set to 0.25, as in the original paper (Xu et al., 2023).

Table 2: Test accuracy (%) after training on corrupted datasets: mean ± standard deviation over five random initializations.

Dataset	Method	Symmetric		Asymmetric		Pairflip		Instance	
		20%	45%	20%	45%	20%	45%	20%	45%
MNIST	Standard	95.66±0.28	87.47±0.82	98.29±0.14	87.73±0.65	97.21±0.41	69.26±3.65	95.66±0.36	71.98±1.43
	Co-teaching	96.62±0.08	96.68±0.07	95.94±0.06	93.40±0.42	96.19±0.10	92.91±0.43	96.49±0.23	95.39±0.59
	JoCoR	99.07±0.05	98.38±0.10	99.02±0.05	95.21±3.70	98.96±0.09	91.06±5.01	99.03±0.05	97.81±0.35
	CDR	98.81±0.09	98.27±0.09	99.14±0.05	94.35±1.29	98.95±0.06	87.55±1.26	98.25±0.16	88.27±1.92
	USDNL	98.38±0.12	97.72±0.13	98.32±0.12	95.83±0.64	98.23±0.09	89.94±1.62	98.13±0.11	96.52±0.38
	BARE	99.07±0.11	**98.78±0.10**	**99.16±0.06**	**98.70±0.12**	99.05±0.06	98.22±0.21	99.05±0.07	**98.42±0.45**
	RLVI	**99.10±0.06**	**98.70±0.18**	98.90±0.22	**98.69±0.05**	**99.10±0.07**	**98.52±0.07**	**99.12±0.03**	98.38±0.08
CIFAR10	Standard	83.36±0.35	60.22±0.28	87.26±0.40	75.03±0.28	81.14±0.43	52.30±0.85	81.76±0.37	55.51±1.04
	Co-teaching	87.55±0.38	83.70±0.46	86.98±0.26	64.84±0.87	86.57±0.21	67.38±2.93	86.40±0.29	66.39±7.16
	JoCoR	90.42±0.07	86.33±0.23	90.60±0.19	76.94±2.56	89.25±0.30	72.53±3.57	89.08±0.36	79.48±1.37
	CDR	85.57±0.38	77.83±0.30	87.98±0.61	75.99±1.95	87.34±0.34	68.13±2.03	85.72±0.74	66.82±2.48
	USDNL	88.65±0.21	83.33±0.23	88.36±0.33	74.62±0.70	87.05±0.24	66.49±2.45	86.90±0.18	71.04±4.17
	BARE	85.67±0.46	69.90±2.06	87.38±0.24	**77.82±1.10**	84.84±0.77	57.18±2.81	85.08±0.73	69.00±2.05
	RLVI	**92.30±0.14**	**88.69±0.33**	**91.73±0.12**	76.96±0.50	**92.15±0.27**	**89.13±0.29**	**91.97±0.33**	**85.15±1.57**
CIFAR100	Standard	61.96±0.11	42.36±0.80	62.88±0.20	39.91±0.58	62.62±0.53	38.94±0.42	62.87±0.32	43.17±0.63
	Co-teaching	58.74±0.66	48.11±1.01	54.32±0.22	34.53±0.68	56.27±0.60	34.47±0.97	57.44±0.32	34.97±0.61
	JoCoR	**69.43±0.25**	62.37±0.94	62.93±0.60	38.68±0.83	65.90±0.35	40.94±0.70	67.98±0.32	52.47±0.31
	CDR	61.57±0.41	46.31±0.81	62.89±0.30	39.47±0.75	61.27±2.23	38.55±0.22	62.09±0.45	41.80±0.77
	USDNL	64.96±0.56	53.82±1.15	59.12±0.10	36.44±0.93	61.76±1.01	35.81±0.20	63.13±0.54	45.42±1.30
	BARE	59.59±1.12	46.56±1.10	52.91±1.28	29.48±1.12	53.29±1.57	30.27±1.14	56.47±0.84	37.24±1.27
	RLVI	**69.64±0.55**	**64.11±0.79**	**69.33±0.83**	**55.76±2.12**	**69.25±0.77**	**55.77±1.01**	**69.54±0.84**	**62.00±1.29**
CIFAR80N-O	Standard	59.41±0.40	37.84±0.79	61.01±0.37	39.13±0.26	60.88±0.33	38.88±0.52		
	Co-teaching	59.77±0.67	48.45±1.44	55.64±0.90	35.98±0.82	58.44±1.27	35.61±0.47		
	JoCoR	70.02±0.80	62.29±0.41	64.12±0.19	40.13±0.40	67.11±0.83	41.85±0.29		
	CDR	56.08±0.98	44.46±1.03	58.16±1.47	36.76±1.28	57.96±0.57	36.16±1.41		
	USDNL	64.07±1.63	52.00±2.89	59.43±0.69	37.04±0.46	62.20±0.52	37.12±1.09		
	BARE	57.39±1.23	42.56±2.20	54.47±1.01	30.12±2.09	55.21±1.28	30.09±0.86		
	RLVI	**71.13 ± 0.71**	**63.18 ± 0.36**	**71.96 ± 0.39**	**54.49 ± 1.76**	**71.45 ± 0.33**	**56.12 ± 0.23**		

Results are presented as mean ± standard deviation over five runs with random initialization of network's parameters. Table 2 shows that RLVI, with the described regularization, attains results competitive with alternative approaches. Also note that additional steps in Algorithm 2 involving $\boldsymbol{\pi}$ (fixed-point iterations and truncation) do not significantly increase the computational time compared to standard SGD. Table 3 shows the average time per epoch during training.

Table 3: Time per one epoch in seconds when training on CIFAR100: mean ± standard deviation over 200 epochs. (Standard: 8.82±0.08)

Co-teaching	JoCoR	CDR
17.91±0.35	18.11±0.11	13.88±0.35
USDNL	**BARE**	**RLVI**
9.40±0.22	9.94 ± 0.10	10.52±0.56

Real corruption. To demonstrate how RLVI performs in a naturally contaminated setting, we train ResNet50 – pre-trained on ImageNet (Wightman, 2019) – on the challenging dataset Food101 (Bossard et al., 2014) consisting of 101 food categories. For each class, 250 testing images were cleaned manually, while the remaining 750 training images per class still contain corrupted labels. We use the Adam optimizer with hyperparameters listed in Table 4.

In the real setting, ε is unknown and has to be optimized for Co-teaching, JoCoR, CDR, and USDNL. Figure 6 shows that by varying the estimated ε for alternative methods (from a large 20% – to a moderate 3% value), one can enhance performance. In contrast, RLVI improves over the standard approach without the need for additional optimization of the hyperparameter. In Appendix, we provide results for other assumptions on ε.

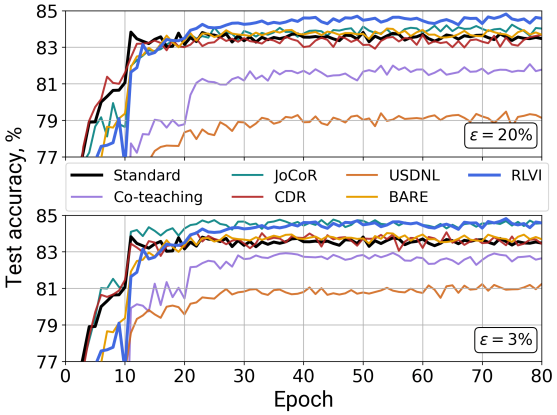


Figure 6: Food101. Test accuracy assuming high and low ε.

5. Conclusions

We presented the novel robust learning algorithm RLVI for likelihood maximization problems with corrupted datasets.

It leverages variational inference to identify corrupted samples under the Huber contamination model using latent Bernoulli variables. This alleviates the need for specifying hyperparameters such as the corruption level, which existing approaches rely on. RLVI can also be implemented as stochastic optimization, which makes it adaptive and applicable to learning from data with varying noise and out-of-core inference for large datasets. We demonstrated the effectiveness of the method on benchmark test problems in both – traditional statistical learning, as well as online and deep learning settings. The proposed RLVI algorithm meets or exceeds performance across considered experimental settings in a parameter-free and efficient manner.

Acknowledgements

The authors are thankful to Prof. Peter Stoica for insightful discussions during the work on this article. The computations/data handling were enabled by the Berzelius resource provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre and by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) at Chalmers e-Commons at Chalmers, and Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX) at Uppsala University, partially funded by the Swedish Research Council through grant agreement nos. 2022-06725 and 2018-05973. DZ and PS acknowledge support from the Swedish Research Council through grant agreement nos. 2018-05040 and 2023-05593 respectively.

Impact Statement

We introduce a novel approach for robust machine learning in likelihood maximization settings. The approach will potentially make robust machine learning easier to apply, owing to its parameter-free and general nature. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bhatia, K., Jain, P., and Kar, P. Robust regression via hard thresholding. *Advances in neural information processing systems*, 28, 2015.

Bhatia, K., Jain, P., Kamalaruban, P., and Kar, P. Consistent robust regression. *Advances in Neural Information Processing Systems*, 30, 2017.

Bishop, C. M. and Nasrabadi, N. M. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Bossard, L., Guillaumin, M., and Van Gool, L. Food-101–mining discriminative components with random forests. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13*, pp. 446–461. Springer, 2014.

Diakonikolas, I., Kamath, G., Kane, D., Li, J., Steinhardt, J., and Stewart, A. Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*, pp. 1596–1606. PMLR, 2019.

Goldberger, J. and Ben-Reuven, E. Training deep neural-networks using a noise adaptation layer. In *International conference on learning representations*, 2016.

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. *Advances in neural information processing systems*, 31, 2018.

Helou, A. E. Sensor har recognition app. matlab central file exchange. 2023. URL https://www.mathworks.com/matlabcentral/fileexchange/54138-sensor-har-recognition-app.

Huber, P. J. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pp. 492–518. Springer, 1992.

Huber, P. J. Robust Statistics. In Lovric, M. (ed.), *International Encyclopedia of Statistical Science*, pp. 1248–1251. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-04898-2. doi: 10.1007/978-3-642-04898-2_594. URL https://doi.org/10.1007/978-3-642-04898-2_594.

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. In *International conference on machine learning*, pp. 2304–2313. PMLR, 2018.

Johnson, N. L., Kotz, S., and Balakrishnan, N. *Continuous univariate distributions, volume 2*, volume 289. John wiley & sons, 1995.

Krizhevsky, A. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, 2009.

LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Maronna, R. A., Martin, R. D., Yohai, V. J., and Salibián-Barrera, M. *Robust statistics: theory and methods (with R)*. John Wiley & Sons, 2019.

Murphy, K. P. *Probabilistic machine learning: Advanced topics*. MIT press, 2023.

Nguyen, D. T., Mummadi, C. K., Ngo, T. P. N., Nguyen, T. H. P., Beggel, L., and Brox, T. Self: Learning to filter noisy labels with self-ensembling. *arXiv preprint arXiv:1910.01842*, 2019.

Osama, M., Zachariah, D., and Stoica, P. Robust risk minimization for statistical learning from corrupted data. *IEEE Open Journal of Signal Processing*, 1:287–294, 2020.

Patel, D. and Sastry, P. Adaptive sample selection for robust learning under label noise. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3932–3942, 2023.

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.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *International conference on machine learning*, pp. 4334–4343. PMLR, 2018.

Wang, Y., Kucukelbir, A., and Blei, D. M. Robust probabilistic modeling with bayesian data reweighting. In *International Conference on Machine Learning*, pp. 3646–3655. PMLR, 2017.

Wei, H., Feng, L., Chen, X., and An, B. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13726–13735, 2020.

Wightman, R. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Xia, X., Liu, T., Han, B., Gong, C., Wang, N., Ge, Z., and Chang, Y. Robust early-learning: Hindering the memorization of noisy labels. In *International conference on learning representations*, 2020.

Xu, Y., Niu, X., Yang, J., Drew, S., Zhou, J., and Chen, R. Usdnl: uncertainty-based single dropout in noisy label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10648–10656, 2023.

Yao, Y., Sun, Z., Zhang, C., Shen, F., Wu, Q., Zhang, J., and Tang, Z. Jo-src: A contrastive approach for combating noisy labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5192–5201, 2021.

Zhang, Z. and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.

Zoubir, A. M., Koivunen, V., Ollila, E., and Muma, M. *Robust statistics for signal processing*. Cambridge University Press, 2018.

A. Optimization of Bernoulli probabilities

A1. Convexity

Objective $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi})$ defined in Equation (15) is a convex function of variables $\boldsymbol{\pi} \in (0; 1)^n$.

Proof. The objective can be viewed as a sum of three terms,

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{i=1}^{n} \pi_i \ell_{\boldsymbol{\theta}}(\boldsymbol{z}_i) + \sum_{i=1}^{n} \pi_i \ln \frac{\pi_i}{\langle \boldsymbol{\pi} \rangle} + \sum_{i=1}^{n} (1 - \pi_i) \ln \frac{1 - \pi_i}{1 - \langle \boldsymbol{\pi} \rangle}. \tag{23}$$

The first term is linear in $\boldsymbol{\pi}$, and we only need to verify if the second and the third terms are convex. Therefore, we first focus on the second term and its Hessian. Using $\mathbf{1}$ for an n-dimensional vector of ones and $\mathbf{diag}(\cdot)$ for a diagonal matrix, we write

$$f(\boldsymbol{\pi}) := \sum_{i=1}^{n} \pi_i \ln \frac{\pi_i}{\langle \boldsymbol{\pi} \rangle}, \tag{24}$$

$$\nabla^2 f(\boldsymbol{\pi}) = \mathbf{diag}\left(\frac{1}{\pi_i}\right) - \frac{\mathbf{1} \cdot \mathbf{1}^\top}{n \langle \boldsymbol{\pi} \rangle}. \tag{25}$$

For $f(\boldsymbol{\pi})$ to be convex, its Hessian has to be positive semi-definite: $\nabla^2 f \succcurlyeq 0$. For clarity we multiply $\nabla^2 f$ by a positive scalar $n\langle \boldsymbol{\pi} \rangle$ and consider a matrix

$$n\langle \boldsymbol{\pi} \rangle \nabla^2 f = \mathbf{diag}\left(\frac{n\langle \boldsymbol{\pi} \rangle}{\pi_i}\right) - \mathbf{1} \cdot \mathbf{1}^\top = \boldsymbol{D} - \mathbf{1} \cdot \mathbf{1}^\top. \tag{26}$$

Notice that $\boldsymbol{D} - \mathbf{1} \cdot \mathbf{1}^\top$ is a Schur complement of the block matrix

$$\begin{pmatrix} \boldsymbol{D} & \mathbf{1} \\ \mathbf{1}^\top & 1 \end{pmatrix}. \tag{27}$$

And, since $\boldsymbol{D} \succ 0$, the following inequalities should hold simultaneously by the properties of the Schur complement:

$$\boldsymbol{D} - \mathbf{1} \cdot \mathbf{1}^\top \succcurlyeq 0 \qquad \Longleftrightarrow \qquad \begin{pmatrix} \boldsymbol{D} & \mathbf{1} \\ \mathbf{1}^\top & 1 \end{pmatrix} \succcurlyeq 0 \qquad \Longleftrightarrow \qquad 1 - \mathbf{1}^\top \boldsymbol{D}^{-1} \mathbf{1} \geq 0. \tag{28}$$

However, expression on the right holds true. Indeed,

$$1 - \mathbf{1}^\top \boldsymbol{D}^{-1} \mathbf{1} = 1 - \mathbf{1}^\top \mathbf{diag}\left(\frac{\pi_i}{n\langle \boldsymbol{\pi} \rangle}\right) \mathbf{1} = 1 - \frac{1}{n\langle \boldsymbol{\pi} \rangle} \sum_{i=1}^{n} \pi_i = 1 - \frac{\langle \boldsymbol{\pi} \rangle}{\langle \boldsymbol{\pi} \rangle} = 0. \tag{29}$$

Therefore, we conclude that $\nabla^2 f \succcurlyeq 0$ and the second term in $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi})$ is convex. To establish the convexity of the third term, it suffices to consider $1 - \pi_i$ as its variables, which makes the proof identical to the above. Hence $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi})$ is convex in $\boldsymbol{\pi}$ as a sum of convex functions. □

A2. Convergence

To see why iterations (18) converge to a minimizer of $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi})$ in Bernoulli probabilities, consider the following. If we over-parameterize $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi})$ and let $\pi_{\mathrm{o}} := \langle \boldsymbol{\pi} \rangle$ be a free variable, then the objective becomes a function of separate variables $\boldsymbol{\pi}$ and π_{o}. Equation (17) defines its closed-form minimizer with respect to $\boldsymbol{\pi}$ due to convexity. Whereas the stationary point condition in terms of π_{o} results in $\pi_{\mathrm{o}} = \sum_{i=1}^{n} \pi_i \,/\, n$. Thus iterations (18) essentially implement the coordinate descent for overparameterized objective in $\boldsymbol{\pi}$ and π_{o} and hence converge to a minimizer $\boldsymbol{\pi}^\star$ of $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi})$.

B. Online learning with varying level of noise

In Figure 7, we provide the performance metrics of the standard SGD approach, RRM, and RLVI, used for binary classification of data that arrives in batches with varying number of corrupted labels. In this figure, we show both accuracy and recall values for all three methods. The metrics are smoothed with a moving average filter.

To obtain the main results in online classification, shown in Figure 7, we used batches of 100 observations during optimization (and reserved 100 samples for computing accuracy and recall in each iteration). To demonstrate how batch size affects the performance, in Figures 8 and 9 we additionally show the same metrics when 75 and 50 examples are used for training respectively (but 100 samples are still used for testing).

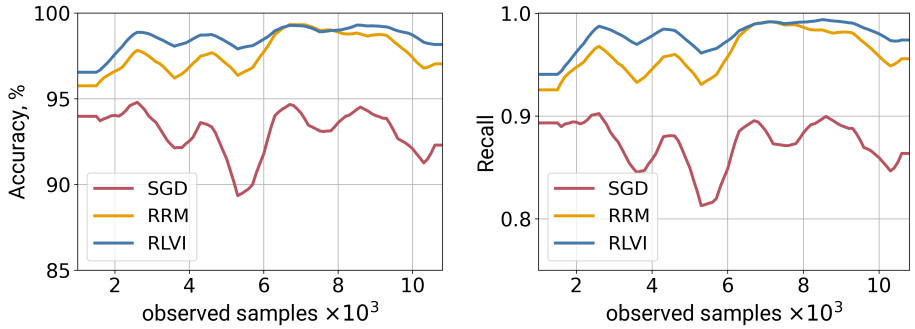


Figure 7: Online classification. Accuracy and recall when learning with the batch size 100.

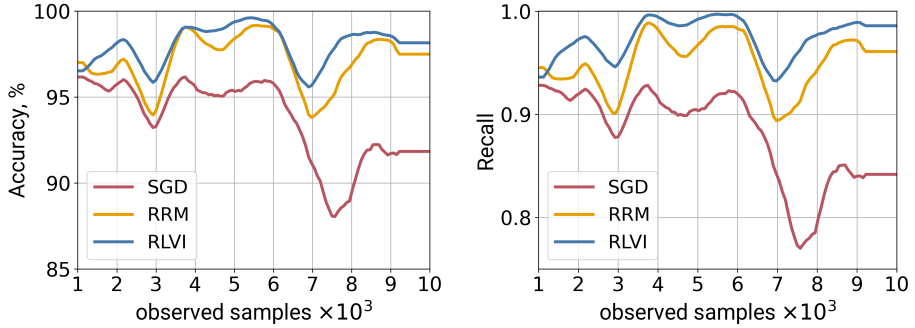


Figure 8: Online classification. Accuracy and recall when learning with the batch size 75.

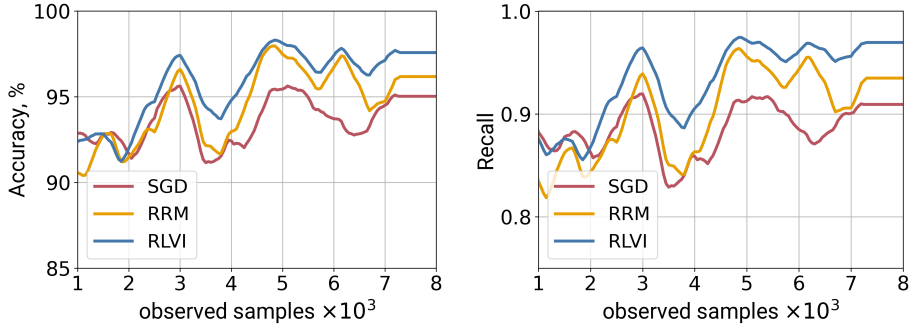


Figure 9: Online classification. Accuracy and recall when learning with the batch size 50.

C. Overparameterized setting: image classification using convolutional neural networks

C1. Hyperparameters

In the experiments on image classification, we use hyperparameter configurations that follow the common settings found in literature. These settings are listed in Table 4.

Table 4: Hyperparameter settings for the deep learning experiments (LR = learning rate, mom. = momentum)

Dataset	MNIST	CIFAR10	CIFAR100 & CIFAR80N-O	Food101
Model	LeNet	ResNet18	ResNet34	ResNet50 (pre-trained on Imagenet)
Optimizer	SGD with mom. 0.9	SGD with mom. 0.9	SGD with mom. 0.9	Adam
Epochs	100	200	200	80
Batch size	32	128	128	32
Weight decay	10^{-3}	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	10^{-4}
LR schedule	linear decay	cosine annealing	cosine annealing	multi-step
Initial LR	10^{-2}	10^{-2}	10^{-2}	10^{-3}

C2. Synthetic corruption

Figures 11 to 16 are the plots, similar to Figure 5, showing the results of image classification after training on the data corrupted with three types of synthetic noise at corruption level ε equal to 0.2 and 0.45. We also provide the ratio of correctly identified corrupted and non-corrupted images based on the introduced criterion. These figures show that, although type II error is not always below 5%, overfitting is reduced in all settings, and RLVI achieves a higher accuracy than the standard method in all problem instances.

C3. Real corruption

In case of the Food101 dataset, a part of the training images is mislabeled. The ratio of such images, ε, is unknown, and thus we run the robust learning algorithms that depend on this hyperparameter: Co-teaching, JoCoR, CDR, and USDNL, using different estimates of ε (40%, 20%, 10%, 5%, 3%, and 1%). Figure 17 presents the accuracy of different methods on the manually cleaned testing set from Food101. As the estimate of ε used in training decreases, the accuracy of alternative methods improves and becomes optimal with ε around 3%. In this experiment, RLVI achieved the best results when no regularization was used. This can be attributed to the weaker overfitting in the case of the Food101 dataset, as compared to examples with synthetic noise: test accuracy for RLVI without regularization in Figure 17 gradually increases, in contrast to the corresponding curve in Figure 5.

D. Unbounded likelihood

As we note in Section 4.1, considering unbounded likelihood function in $\ell_{\boldsymbol{\theta}}(\boldsymbol{z})$ can lead to a degenerate solution $\boldsymbol{\theta}$ when using RLVI. In general, the problem of likelihood maximization is ill-posed, and one example when such pathological estimate arises is, e.g, the Gaussian Mixture Model (Bishop & Nasrabadi, 2006). In the following, we address covariance estimation from corrupted data. The corresponding negative log-likelihood function, which is used to estimate the mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$, is

$$\ell_{\boldsymbol{\theta}}(\boldsymbol{z}) = \frac{1}{2}\left[(\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu}) + \ln\det\boldsymbol{\Sigma} + d\ln 2\pi\right], \tag{30}$$

where d is the dimension of the problem and $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The first and second terms of (30) become unbounded as $\boldsymbol{\Sigma}$ approaches rank deficiency. We note that the covariance estimate for RLVI is of the form:

$$\boldsymbol{\Sigma}_{\text{RLVI}} = \frac{1}{\boldsymbol{\pi}^\top \mathbf{1}} \sum_{i=1}^{n} \pi_i (\boldsymbol{z}_i - \boldsymbol{\mu}_{\text{RLVI}})(\boldsymbol{z}_i - \boldsymbol{\mu}_{\text{RLVI}})^\top, \tag{31}$$

where $\boldsymbol{\mu}_{\text{RLVI}} = \sum_{i=1}^{n} \pi_i \boldsymbol{z}_i \,/\, \boldsymbol{\pi}^\top \mathbf{1}$. Therefore it is possible to find a set of weights π_i such that $\boldsymbol{\Sigma}_{\text{RLVI}}$ becomes rank-deficient in a manner that minimizes the loss. This is indeed confirmed in our experiments.

To avoid such a singular solution, we impose additional regularization for $\boldsymbol{\pi}$ and ensure that enough samples are used for learning: $\sum_{i=1}^{n} \pi_i = n(1-\varepsilon) \geq n_0$. That is, the total number of non-corrupted samples should be at least n_0. The corresponding constrained optimization problem for $\boldsymbol{\pi}$ is

$$\begin{cases} \min\limits_{\boldsymbol{\pi} \in (0;1)^n} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi}), \\ \boldsymbol{\pi}^\top \mathbf{1} \geq n_0. \end{cases} \tag{32}$$

Its Lagrangian $L(\boldsymbol{\pi}, \lambda)$ is thus the sum of objective $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi})$, and a constraint weighted with Lagrange multiplier λ:

$$L(\boldsymbol{\pi}, \lambda) = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi}) + \lambda \left(n_0 - \boldsymbol{\pi}^\top \mathbf{1}\right). \tag{33}$$

In this analysis we consider the optimization step with respect to $\boldsymbol{\pi}$, and thus omit $\boldsymbol{\theta}$ from the variables of the Lagrangian above. Consequently, the corresponding Karush–Kuhn–Tucker conditions give

1. $\nabla_{\boldsymbol{\pi}} L(\boldsymbol{\pi}, \lambda) = \nabla_{\boldsymbol{\pi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\pi}) - \lambda \cdot \mathbf{1} = 0 \implies \pi_i = \left(1 + \dfrac{1 - \langle \boldsymbol{\pi} \rangle}{\langle \boldsymbol{\pi} \rangle} e^{\ell_i - \lambda}\right)^{-1}$,
2. $\lambda \geq 0$,
3. $\lambda \cdot \left(n_0 - \boldsymbol{\pi}^\top \mathbf{1}\right) = 0$.

Note that if the constraint is not active, we fall back to the default RLVI formulation in which $\boldsymbol{\pi}$ are computed with (18). In contrast, when the constraint is active, a change to the solution $\boldsymbol{\pi}$ is required.

a. Inactive constraint: $\boldsymbol{\pi}^\top \mathbf{1} > n_0 \implies \lambda = 0$, and

$$\pi_i = \left(1 + \frac{1 - \langle \boldsymbol{\pi} \rangle}{\langle \boldsymbol{\pi} \rangle} e^{\ell_i}\right)^{-1}. \tag{34}$$

b. Active constraint: $\boldsymbol{\pi}^\top \mathbf{1} = n_0 \implies \lambda > 0$. In this case, $\boldsymbol{\pi}^\top \mathbf{1} = n_0$, and for $\boldsymbol{\pi}$ and λ, we obtain the following equations:

$$\begin{cases} \pi_i = \left(1 + \dfrac{n - n_0}{n_0} e^{\ell_i - \lambda}\right)^{-1}, \\ \lambda > 0, \\ \sum\limits_{i=1}^{n} \pi_i = n_0. \end{cases} \tag{35}$$

Hence, in the case of a sparse solution (most π_i are around zero), (35) allows us to find the dual variable $\lambda_\star$ and, after a substitution, obtain the corrected $\boldsymbol{\pi}$.

To test this regularization in practice, we reproduce the covariance estimation problem from (Osama et al., 2020): $n = 50$ samples are generated synthetically, of which $\varepsilon = 20\%$ are sampled from the corrupted distribution. The dimension of samples is $d = 2$; 100 Monte Carlo tests are performed, similar to Section 4.1. Note that alternative methods, SEVER and RRM, use the following upper-bound: $\varepsilon \leq \widetilde{\varepsilon} = 30\%$. Hence, for a fair comparison, we set $n_0 = n(1 - \widetilde{\varepsilon}) = 35$. Figure 10 shows relative errors obtained with standard Maximum Likelihood (ML), SEVER, RRM, and the constrained variant of RLVI according to (32). These results demonstrate that in the case of the unbounded likelihood one might still need to employ the hyperparameter $\widetilde{\varepsilon}$. But, in this case as well, RLVI can attain a low relative error with a tight confidence interval.

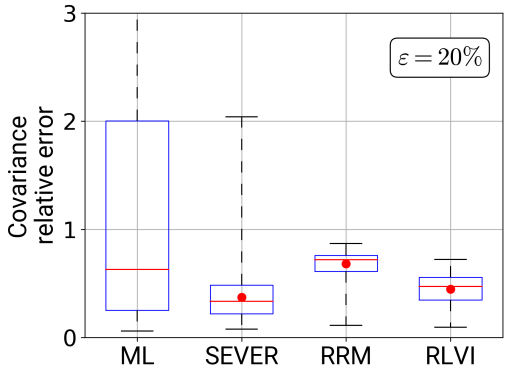
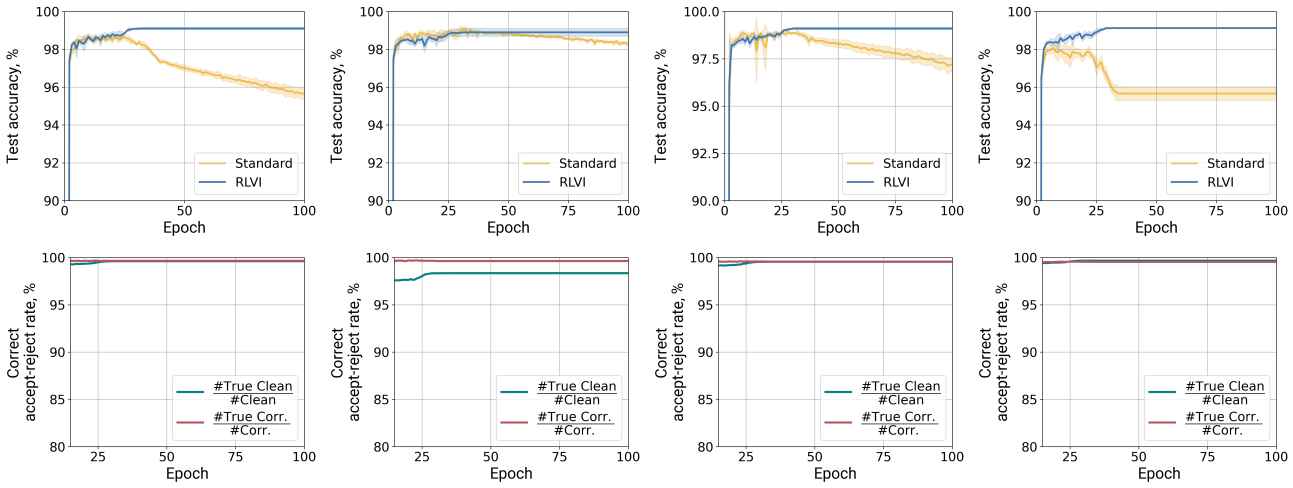
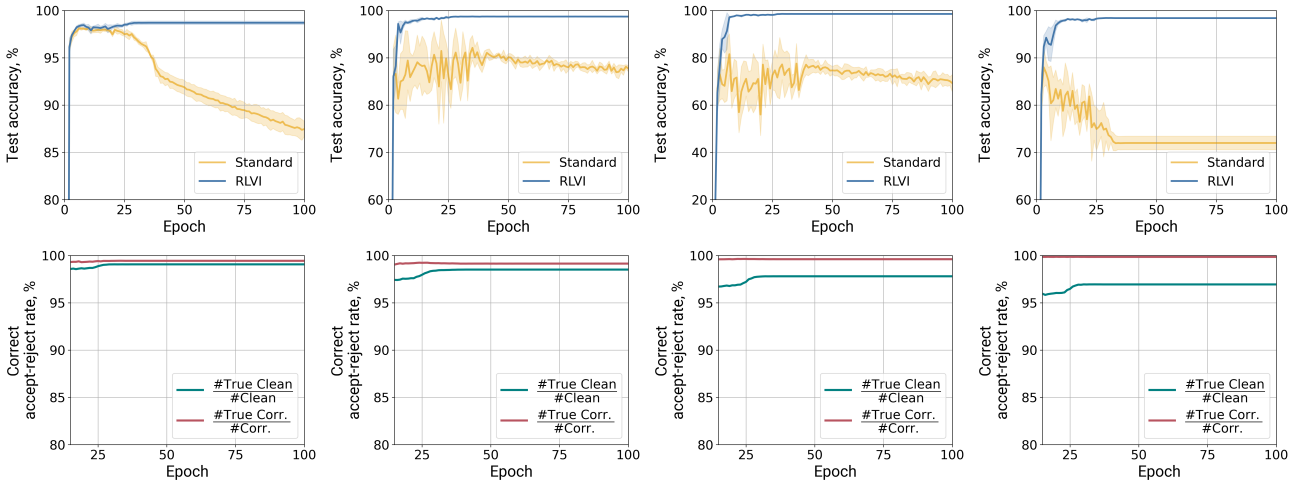


Figure 10: Covariance estimation. Boxplots of relative errors after 100 Monte Carlo runs. Constrained RLVI (32) is not subject to the singular covariance issue that arises due to the unbounded likelihood.



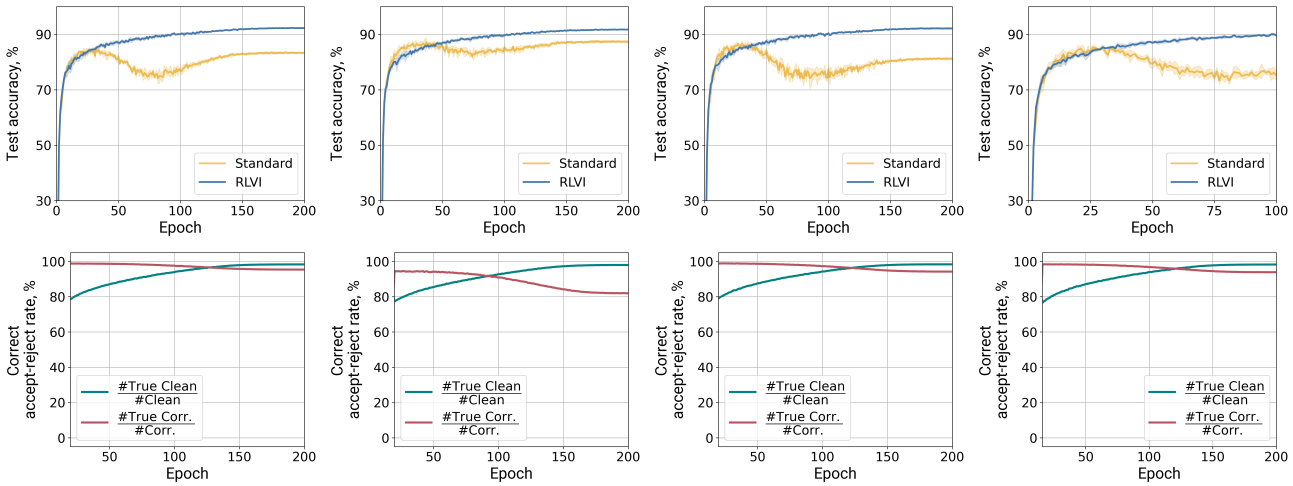
Symmetric Asymmetric Pairflip Instance

Figure 11: MNIST, 20% noise rate. *Top*: test accuracy (mean $\pm$ st. dev. over 5 runs). *Bottom*: type I and type II errors (mean over 5 runs).



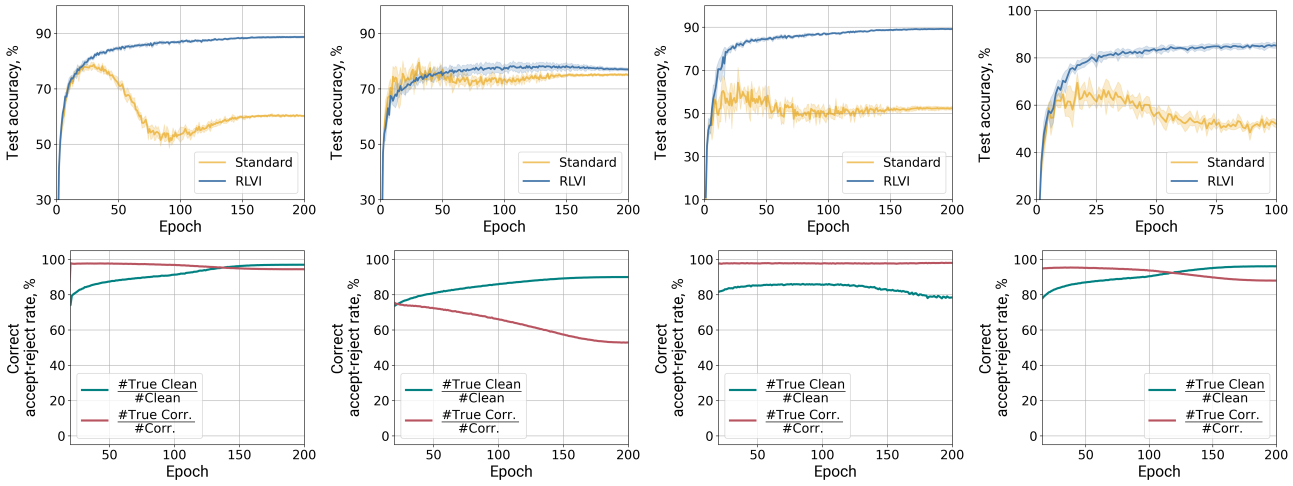
Symmetric Asymmetric Pairflip Instance

Figure 12: MNIST, 45% noise rate. *Top*: test accuracy (mean $\pm$ st. dev. over 5 runs). *Bottom*: type I and type II errors (mean over 5 runs).



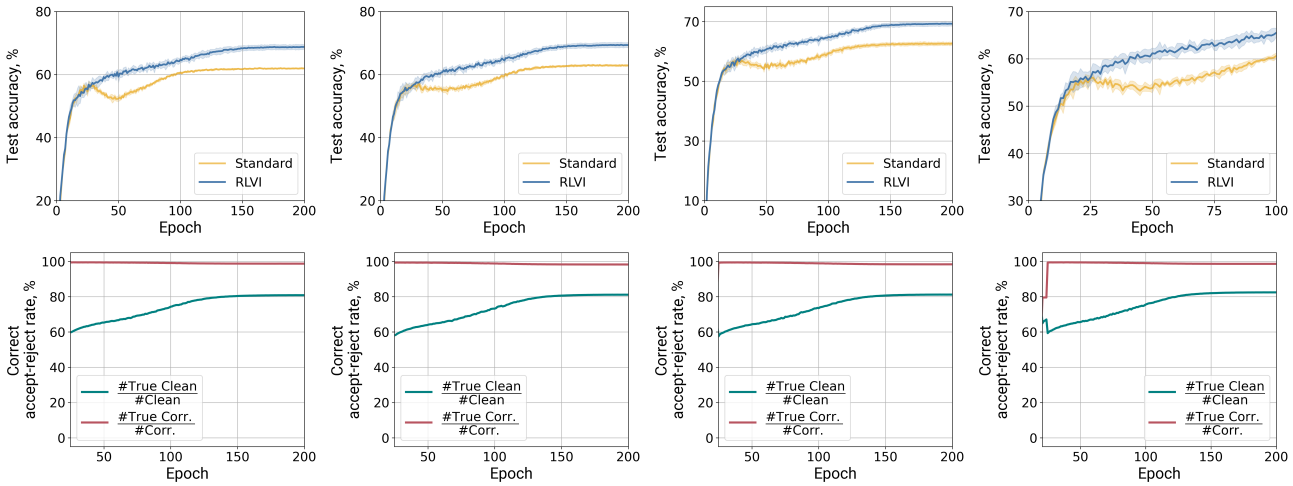
Symmetric Asymmetric Pairflip Instance

Figure 13: CIFAR10, 20% noise rate. *Top*: test accuracy (mean ± st. dev. over 5 runs). *Bottom*: type I and type II errors (mean over 5 runs).



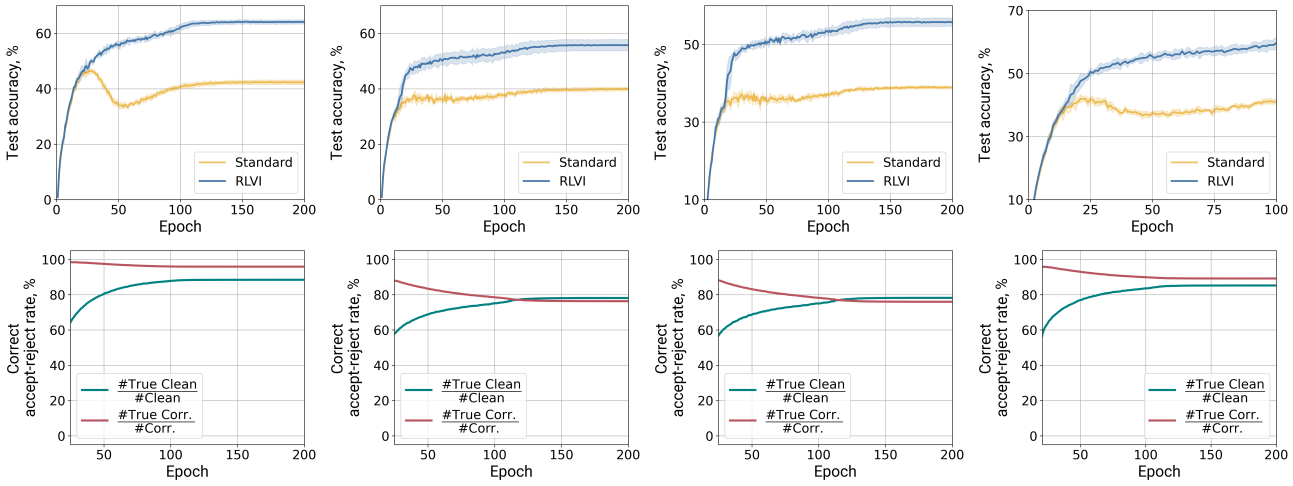
Symmetric Asymmetric Pairflip Instance

Figure 14: CIFAR10, 45% noise rate. *Top*: test accuracy (mean ± st. dev. over 5 runs). *Bottom*: type I and type II errors (mean over 5 runs).



Symmetric Asymmetric Pairflip Instance

Figure 15: CIFAR100, 20% noise rate. *Top*: test accuracy (mean ± st. dev. over 5 runs). *Bottom*: type I and type II errors (mean over 5 runs).



Symmetric Asymmetric Pairflip Instance

Figure 16: CIFAR100, 45% noise rate. *Top*: test accuracy (mean ± st. dev. over 5 runs). *Bottom*: type I and type II errors (mean over 5 runs).

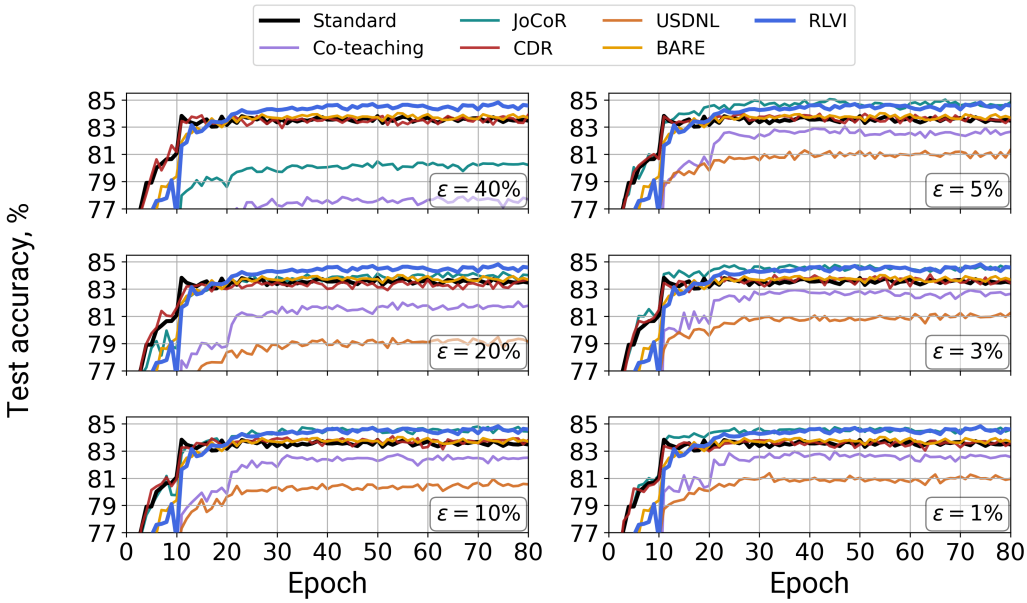


Figure 17: Test accuracy (%) for Food101, using different estimates of corruption level ε for Co-teaching, JoCoR, CDR, and USDNL.