[Re] Can gradient clipping mitigate label noise?

David Mizrahi
EPFL
david.mizrahi@epfl.ch

Oğuz Kaan Yüksel
EPFL
oguz.yuksel@epfl.ch

Aiday Marlen Kyzy
EPFL
aiday.marlenkyzy@epfl.ch

Reproducibility Summary

Scope of Reproducibility

The original paper proposes partially Huberised losses, which possess label noise robustness. The authors claim that there exists label noise scenarios which defeat Huberised but not partially Huberised losses, and that partially Huberised versions of existing losses perform well on real-world datasets subject to symmetric label noise.

Methodology

All the experiments described in the paper were fully re-implemented using NumPy, SciPy and PyTorch. The experiments on synthetic data were run on a CPU, while the deep learning experiments were run using a Nvidia RTX 2080 Ti GPU. Running the experimentation necessary to gain some insight on some of the network architectures used and reproducing part of the real-world experiments required over 500 GPU hours.

Results

Overall, our results mostly support the claims of the original paper. For the synthetic experiments, our results differ when using the exact values described in the paper, although they still support the main claim. After slightly modifying some of the experiment settings, our reproduced figures are nearly identical to the figures from the original paper. For the deep learning experiments, our results differ, with some of the baselines reaching a much higher accuracy on both MNIST and CIFAR-100. Nonetheless, with the help of an additional experiment, our results support the authors' claim that partially Huberised losses perform well on real-world datasets subject to label noise.

What was easy

The original paper is well written and insightful, which made it fairly easy to implement the partially Huberised version of standard losses based on the information given. In addition, recreating the synthetic datasets used in two of the original paper's experiments was relatively straightforward.

What was difficult

Even though the authors were very detailed in their feedback, finding the exact hyper-parameters used in the real-world experiments required many iterations of inquiry and experimentation. In addition, the CIFAR-10 and CIFAR-100 experiments can be difficult to reproduce due to the high number of experiment configurations, resulting in many training runs and a relatively high computational cost of over 600 GPU hours. As a consequence, the CIFAR-10 results could not be obtained in time, although the code to run this experiment was implemented.

Communication with original authors

We contacted the authors regarding some of the hyper-parameters used in their experiments, to which they promptly replied with very detailed explanations.

Preprint. Under review.

1 Introduction

Gradient clipping, a well-established technique in machine learning, is mainly studied with an optimization perspective. For example, the well-known problem of exploding gradients is studied in Bengio et al. (1994). The effect of clipping in optimization dynamics is studied in Hazan et al. (2015); Levy (2016); Zhang et al. (2019a),

In this work however, we reproduce[1] the paper "Can gradient clipping mitigate label noise?" (referenced as "the paper" or "the original paper") by Menon et al. (2019) (referenced as "the authors") which focuses on *robustness* properties of gradient clipping. More specifically, clipping is studied under the common problem of label noise.

The authors make the following main contributions:

a) Gradient clipping does not provide label noise robustness to even simple models. Specifically in a simple setup, clipping is linked to using a *Huberised* loss, which preserves classification-calibration but is not robust to symmetric label noise.

b) A new clipping variant for composite losses is proposed, where only the contribution from the base loss is considered for clipping. The equivalent *partially Huberised* loss preserves classification-calibration and is robust to label noise.

c) Two experiments on synthetic data highlight the robustness properties of partially Huberised losses. Moreover, partially Huberised versions of existing losses are empirically shown to behave well under label noise, using real world datasets.

We provide a high-level summary of the contributions *a)* and *b)* in section 3. We thoroughly reproduce the experiments listed in *c)* in section 4 and section 5. Lastly, we evaluate the empirical evidence in section 6 to assess the claims made in *c)*.

2 Background

Gradient clipping. Consider a supervised learning task with samples $(x, y) \in (\mathcal{X} \times \mathcal{Y}) \sim D$, and a loss function $l_\theta : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. For this setting the gradient $g(\theta)$ and the clipped gradient $\bar{g}_\tau(\theta)$ are defined as follows:

$$g(\theta) = \frac{1}{N} \sum_{n=1}^{N} \nabla l_\theta(x_n, y_n) \qquad \bar{g}_\tau(\theta) = \begin{cases} \tau \frac{g(\theta)}{\|g(\theta)\|_2} & \text{if } \|g(\theta)\|_2 \geq \tau \\ g(\theta) & \text{otherwise} \end{cases}$$

Label noise. In classification under label noise, one has samples from a noisy distribution $P_{\bar{D}}(x, y)$ instead of a clean distribution $P_D(x, y)$. For example, under *symmetric* label noise, all instances have a constant probability of their labels being flipped uniformly to any of the other classes. The task remains to minimize risk over the the clean D. Some recent loss-based proposals for learning under symmetric label noise are the linear or unhinged loss (van Rooyen et al., 2015) and the generalized cross-entropy loss (Zhang and Sabuncu, 2018).

3 Summary of the paper

The robustness properties of gradient clipping are studied under the problem of symmetric label noise. More specifically, whether gradient clipping can provide any robustness against such noises if used, is inspected in a simple setting: stochastic gradient descent with linear models in a binary classification task.

Analysis begins by noting that in this setting, for *margin* losses, using gradient clipping is equivalent to a simple modification of the base loss. Additionally, gradient clipping is linked to a loss-based version of gradient clipping, named L-clipping, where only the contribution from the loss function itself is clipped instead of the full gradient. Moreover, L-clipping is shown to be equivalent to using a Huberised version of the base loss.

The analysis then turns to studying Huberised losses instead of clipping itself, which is the main technique the authors use to study the robustness properties of various clipping strategies. Next, Huberised losses are shown to preserve classification calibration and class probabilities. However, it is also shown that if the underlying loss is convex, the Huberised loss is still convex and thus still vulnerable to outliers. Then, a formal treatment proves that there exists a separable distribution for which, under symmetric label noise, the optimal linear classifier for a Huberised loss is tantamount to random guessing. Note that all of these steps complement or generalize the already established literature on Huber or square Huber losses (Huber, 1964).

[1] Our code is available at: https://github.com/dmizr/phuber

After showing traditional clipping does not yield any robustness to label noise, the authors consider a slight variation for *composite* losses: since the link functions is typically Lipschitz (e.g. for sigmoid and softmax), they note that one can consider clipping only on the base loss. This variant, named CL-clipping, is found to be equivalent to applying partial Huberisation to the underlying loss. Following treatment for Huberisation, partially Huberised losses are shown to be classification-calibrated and for suitably large τ, probability preserving. Lastly but most importantly, for partially Huberised losses, the optimal solution in noisy regimes is shown to be close to the one for the clean regime, establishing their robustness to label noise.

After the analysis, the authors propose a multi-class generalization of their partially Huberised version of cross-entropy loss (Equation 1). Suppose we have softmax probability estimates $p_\theta(x, y) \propto \exp(m_\theta(x, y))$, then the *partially Huberised softmax cross-entropy loss* (PHuber-CE) is defined for $\tau > 1$ as:

$$\tilde{\ell}_\theta(x, y) = \begin{cases} -\tau \cdot p_\theta(x, y) + \log \tau + 1, & \text{if } p_\theta(x, y) \leq \frac{1}{\tau} \\ -\log p_\theta(x, y), & \text{otherwise.} \end{cases} \tag{1}$$

Finally, the authors evaluate their partially Huberised losses in two experiment on synthetic data (referenced as "synthetic experiments") to demonstrate its behavior under symmetric label noise. Moreover, they also assess the effectiveness of partial Huberisation on real-world datasets subject to symmetric label noise (referenced as "real-world experiments").

All of these experiments are covered extensively in section 4 and section 5 of our paper. Note that section 3 and 4 with Appendix A of the original paper are quite technical and the summary provided here is merely a high-level overview. For more information, we kindly refer to the original paper (Menon et al., 2019).

4 Synthetic experiments

We now study two synthetic experiments proposed by authors, to show the existence of label noise scenarios that defeat Huberised but not partially Huberised losses. We will start by discussing the 2D setting proposed in Long and Servedio (2010) and then discuss the 1D outliers setting given in Ding (2013). These experiments are fully re-implemented with NumPy (Harris et al., 2020) and SciPy (Virtanen et al., 2020). Experimental setups including methods and hyper-parameters are fully verified according to the original paper and in necessary cases, according to the additional details obtained from the authors. Our experiments are configurable through the Hydra framework (Yadan, 2019).

4.1 Long & Servedio dataset

Long and Servedio (2010) considers a set of four positive labeled points: one *large margin* example $(1, 0)$, one *puller* example $(\gamma, 5\gamma)$ and two *penalizer* examples $(\gamma, -\gamma)$ where $0 < \gamma < \frac{1}{6}$- in a binary classification task with a linear model without a bias term. The halfspace $x > 0$ correctly classifies all the samples. However, one can show that under symmetric label noise, minimizing over wide range of convex losses with a suitable γ will result in a predictor equivalent to a random predictor.

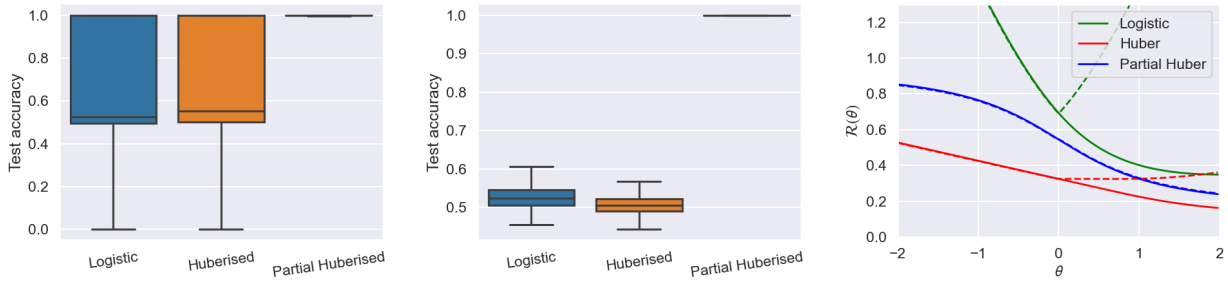
The authors build on Long and Servedio (2010), and consider a mixture of six isotropic Gaussians $\mathcal{N}(\mu_i, \sigma^2 I_2)$, with $\sigma = 0.01$ and $\mu_i \in \{\pm(1, 0), \pm(\gamma, 5\gamma), \pm(\gamma, -\gamma)\} \subset \mathbb{R}^2$, with $\gamma = \frac{1}{24}$. Mixing weights are $\frac{1}{4}$ for the two Gaussians centered around $\pm(\gamma, -\gamma)$ and $\frac{1}{8}$ for the rest. An instance (x_1, x_2) is labeled positive if $x_1 \geq 0$ and negative otherwise. $N = 1000$ random samples are drawn from this distribution, and the label of each sample is flipped with corruption probability $\rho < 0.5$. Then, a linear classifier is trained using Scipy's SLSQP (Sequential Least Squares Programming) optimizer for a maximum of 100 iterations with each of the following losses:

- the logistic loss
- the Huberised version of logistic loss, with $\tau = \sigma(-1) \sim 0.26$
- the partially Huberised version of logistic loss, with $\tau = 1 + e^{-1} \sim 1.36$

After contacting the authors, we found that the above τ values were used instead of the values provided in the original paper, which were $\tau = 1.0$ and $\tau = 2.0$ for the Huberised and the partially Huberised loss respectively.

Once trained, the classifier is evaluated on 500 clean test samples.

Figure 1a and Figure 1b show our results over 500 independent runs for $\rho = 0.45$ and $\rho = 0.2$ respectively. When using $\rho = 0.45$, as stated in the original paper, we fail to reproduce a figure that *exactly* matches the authors' results. However, through experimentation, we found that for a lower level of noise corruption such as $\rho = 0.2$, we get results that are very similar to the original paper, with the partially Huberised loss always achieving perfect classification, while the logistic and Huberised losses succumb to label noise and perform no better than chance.



(a) Long & Servedio with $\rho = 0.45$ (b) Long & Servedio with $\rho = 0.2$ (c) Effect of outliers

Figure 1: Reproduction of Long & Servedio and Ding experiments. In the Ding experiment (c), the solid curve denotes empirical risk without outliers, while the dashed curve denotes empirical risk with outliers.

4.2 Outliers dataset

The 1-D setting from Ding (2013) is composed of 10,000 linearly separable inliers: 5000 samples from the unit variance Gaussian $\mathcal{N}(1, 1)$ with positive label, and 5000 samples from the mirror image $\mathcal{N}(-1, 1)$ with negative label. In addition, 50 outliers are added: 25 samples from $\mathcal{N}(-200, 1)$ with positive label, and 25 samples from $\mathcal{N}(200, 1)$ with negative label. Assuming a linear model characterized by a scalar $\theta \in \mathbb{R}$, we comparatively evaluate the empirical risk with and without outliers. We use the same three losses as in subsection 4.1 but with $\tau = 0.1$ and $\tau = 1.1$ for the Huberised and partially Huberised loss respectively. [2]

Figure 1c shows our results where dashed and solid curves represent the cases with and without outliers respectively. As in the original paper, the optimal solutions for the logistic and Huberised loss are changed from $\theta^* = +\infty$ to $\theta^* = 0$ with the introduction of outliers, whereas the partially Huberised loss remains intact.

5 Real-world experiments

We now consider the deep learning experiments performed on three image classification datasets: MNIST, CIFAR-10 and CIFAR-100. These experiments were fully re-implemented with PyTorch (Paszke et al., 2019), according to the description from the paper and implementation details obtained from the authors after contacting them. Configuration management for these experiments was done with the help of the Hydra framework (Yadan, 2019).

5.1 MNIST

5.1.1 Methodology

MNIST is a dataset of handwritten digits, consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image associated with a label from 10 distinct classes. The dataset is normalized using the mean and standard deviation from the training set, and no data augmentation is applied. The training labels are then corrupted with symmetric noise at flip probability $\rho \in \{0.0, 0.2, 0.4, 0.6\}$. As in the original paper, the same random seed is used to corrupt the training labels across all trials.

We use a LeNet-5 (Lecun et al., 1998), with a few modifications in order to reproduce the authors' settings as accurately as possible. Most notably, the $\tanh$ activation layers from the original LeNet are changed to $ReLU$, and the weights are initialized according to a truncated normal distribution with standard deviation $\sigma = 0.1$.

This model is trained for 20 epochs using Adam (Kingma and Ba, 2017) with batch size $N = 32$, and weight decay of 10^{-3}. The initial learning rate is set to 0.001, and is lowered following an exponential decay schedule with decay rate 0.1 and decay steps of 10,000. That is, the learning rate at iteration n is set to: $\eta_n = \eta_0 \cdot r^{n/s}$, with $\eta_0 = 0.001$, $r = 0.1$ and $s = 10^4$. According to the authors, these hyper-parameter values were chosen to obtain a good baseline performance in a setting with no label noise.

For each level of label noise corruption, the test set accuracy of 6 different loss functions is compared:

[2] In the original paper, the τ values mistakenly reported as 1.0 and 2.0, along with the values in subsection 4.1. These updated values are obtained from the authors, after informing them $\tau = 1.0$ for Huberisation is equivalent to keeping base loss intact.

- the cross-entropy loss (CE)
- the linear or unhinged loss (van Rooyen et al., 2015)
- the generalized cross-entropy loss (GCE), with $\alpha = 0.7$ (Zhang and Sabuncu, 2018)
- the cross-entropy loss, with global gradient clipping applied using a max norm threshold of $\tau = 0.1$
- the partially Huberised version of the cross-entropy loss (PHuber-CE), with $\tau = 10$
- the partially Huberised version of the generalized cross-entropy loss (PHuber-GCE), with $\alpha = 0.7$ and $\tau = 10$.

The CE loss serves as a baseline, while the linear and GCE losses serve as representative noise-robust losses. The model and hyper-parameters used are identical for all losses at all levels of label noise.

5.1.2 Computational requirements

This LeNet model was trained with a Nvidia RTX 2080 Ti GPU. Each run took roughly 2 minutes. Fully reproducing the authors' experiments required training this model 72 times, in order to do 3 trials for each combination of loss function and level of label noise. This resulted in a total training time of around 2 hours.

5.1.3 Results

Our results are reported in Table 1, and a comparison with the original paper's results can be found in Figure 2. Our reproduction matches the results from the original paper for both the PHuber-CE and PHuber-GCE losses, although the CE, CE with gradient clipping and linear losses perform considerably better at high levels of label noise than what was reported. As a consequence, the partially Huberised version of these losses do not outperform the base losses at high levels of label noise, contrary to the original paper's results. It is of note that in our reproduction, all losses, with the exception of the CE loss with gradient clipping, perform comparably, with a test accuracy higher than 97.5% at all levels of label noise.

Dataset	Loss function	$\rho = 0.0$	$\rho = 0.2$	$\rho = 0.4$	$\rho = 0.6$
MNIST	CE	99.1 ± 0.1	98.8 ± 0.0	98.6 ± 0.0	98.0 ± 0.1
	CE + clipping	97.0 ± 0.0	96.5 ± 0.0	95.7 ± 0.1	94.7 ± 0.1
	Linear	95.0 ± 3.5	98.5 ± 0.1	98.2 ± 0.0	97.6 ± 0.0
	GCE	98.8 ± 0.0	98.7 ± 0.0	98.5 ± 0.0	98.1 ± 0.0
	PHuber-CE $\tau = 10$	99.0 ± 0.0	98.8 ± 0.1	98.5 ± 0.1	97.6 ± 0.0
	PHuber-GCE $\tau = 10$	98.9 ± 0.0	98.7 ± 0.0	98.4 ± 0.0	98.0 ± 0.0

Table 1: Reproduction of the MNIST experiments. The mean and standard error of the test accuracy over 3 trials is reported. The highlighted cells correspond to the best performing loss at a given ρ.

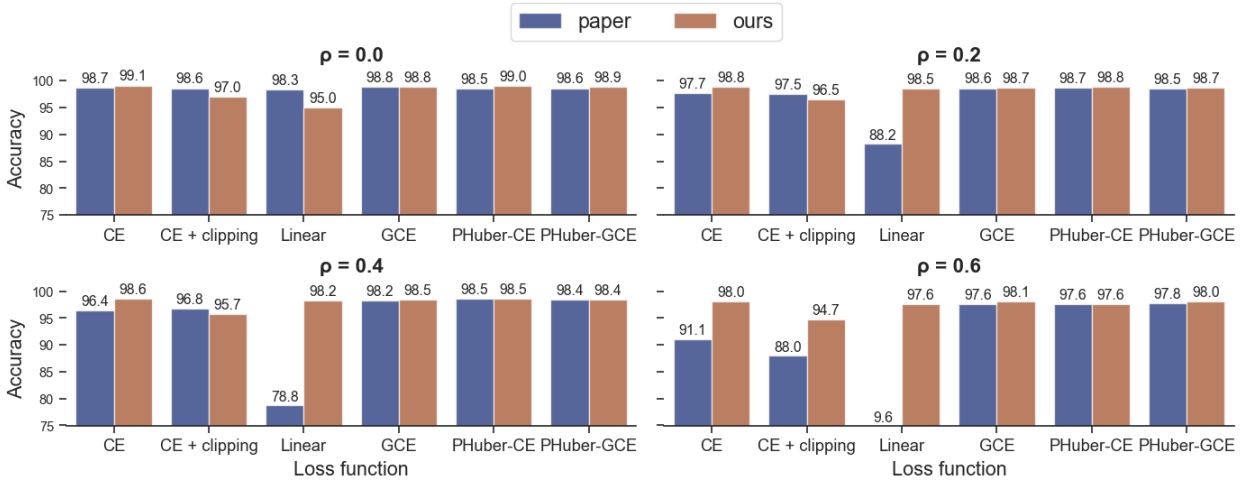


Figure 2: Test accuracy of LeNet-5 on MNIST

5.2 CIFAR-10 and CIFAR-100

Only the CIFAR-100 experiment was reproduced, due to the high computational cost of running the CIFAR-10 experiment, as described in subsubsection 5.2.2. We nonetheless describe the models and updated hyper-parameters used for both experiments.

5.2.1 Methodology

The CIFAR-10 and CIFAR-100 datasets (Krizhevsky and Hinton, 2009) both consist of a training set of 50,000 examples and a test set of 10,000 examples. Each example is a 32×32 color image, associated with a label from 10 distinct classes for CIFAR-10, and 100 distinct classes for CIFAR-100. Both datasets are normalized using per-channel mean and standard deviation, and the standard data augmentation for these datasets is applied, akin to Zagoruyko and Komodakis (2016). That is, images are zero-padded with 4 pixels on each side to obtain a 40×40 image, and then a random 32×32 crop is extracted and mirrored horizontally with 50% probability. As in the MNIST experiment, the training labels are corrupted with symmetric noise at flip probability $\rho \in \{0.0, 0.2, 0.4, 0.6\}$, with identical noise seed across trials.

For both of these experiments, we use a ResNet-50 (He et al., 2015), as implemented in Liu (2017). This implementation of the ResNet-50 differs from the one described by He et al. (2015) to make it more appropriate for classification on small images. The number of filters per layers are identical, but the first layer, originally a 7x7 convolutional layer with stride 2 and padding 3, is changed to a 3x3 convolutional layer with stride 1 and padding 1, and the max-pooling layer that follows is removed. By removing these early down-sampling layers, this architecture performs better on CIFAR-10 and CIFAR-100 than the original ResNet-50, which was designed for classification on ImageNet (Deng et al., 2009). We decided to use such an implementation for several reasons: First, it is used in many popular papers performing classification on CIFAR with ResNets, such as DeVries and Taylor (2017); Zhang et al. (2018); Li et al. (2020); Zhang et al. (2019b). Second, using the original ResNet-50 yielded poor results, especially on partially Huberised losses. Third, after contacting the authors about their implementation, they confirmed using a ResNet-50 with some of the early downsampling layers removed, but could not provide more details as to which layers were specifically changed or removed [3].

For CIFAR-10, this ResNet is trained for 400 epochs using SGD with Nesterov momentum 0.1 (Nesterov, 1983; Sutskever et al., 2013), batch size $N = 64$, and weight decay of 5×10^{-4}. [4] The initial learning is set to 0.1, and is divided by 10 at the 160th, 300th and 360th epoch. For CIFAR-100, this ResNet is trained for 200 epochs using SGD with Nesterov momentum 0.1, batch size $N = 128$, and weight decay of 5×10^{-4}. [5] The initial learning is set to 0.1, and is divided by 5 at the 60th, 120th and 160th epoch. According to the authors, these hyper-parameters were partially based on the setting from DeVries and Taylor (2017), and were chosen to obtain a good performance with CE in a setting with no label noise.

As in the MNIST experiment, the test set accuracy of the CE, CE with gradient clipping, linear, GCE, PHuber-CE and PHuber-GCE losses are compared. The tunable parameters for these losses are identical to the ones used in the MNIST experiment, with the exception of PHuber-CE for CIFAR-10, where $\tau = 2$. The model and hyper-parameters used are identical for all losses at all levels of label noise.

We also report an additional experiment, where we train CIFAR-100 using the PHuber-CE loss with $\tau = 50$. This corresponds to linearizing the base loss at probability threshold 0.02.

5.2.2 Computational requirements

We use a Nvidia RTX 2080 Ti GPU to train these models. With full precision training, a run on CIFAR-10 takes approximately 12 hours, while a run on CIFAR-100 takes approximately 4 hours, due to the lower amount of epochs and higher batch size. In order to accelerate the training process, we implement mixed precision training (Micikevicius et al., 2017), which results in a 2x speed-up with no decrease in accuracy compared to full precision training.

[3] In the ResNet paper, He et al. also propose ResNet architectures suited for CIFAR-10 classification, such as the ResNet-44 and ResNet-56, which have less filters per layer compared to the implementations from Liu (2017), resulting in faster training at the cost of lower accuracy. These ResNet architectures were not used in our reproduction as the authors specifically mentioned using a ResNet-50.

[4] In the original paper, the weight decay is mistakenly reported as 5×10^{-3}, and it was not specified that the type of momentum used was Nesterov momentum. These updated hyper-parameters were obtained from the authors, after informing them of our difficulty reproducing their experiments with the values from the paper.

[5] See previous footnote.

For CIFAR-100, fully reproducing the authors' experiments required training this model 72 times, resulting in a total training time of around 150 hours, equivalent to 6 days. Even with mixed precision training, fully reproducing the authors' experiments on CIFAR-10 was not feasible, as it would require around 450 GPU hours, equivalent to 19 days.

5.2.3 Results

Our results are reported in Table 2, and a comparison with the original paper's results can be found in Figure 3.

For nearly all configurations, our reproduction achieves better results than the original paper. Most notably, the accuracy of the CE, GCE and PHuber-GCE losses are noticeably better at all levels of noise corruption. Similarly to the original paper's results, PHuber-GCE with $\tau = 10$ achieves the best accuracy out of all losses for $\rho = 0.4$ and $\rho = 0.6$, and performs comparably to GCE for $\rho = 0.0$ and $\rho = 0.2$. Unlike the paper's results, PHuber-CE with $\tau = 10$ performs quite poorly compared to CE, even in settings with high levels of label noise where it should supposedly perform well. However, with our additional experiment using PHuber-CE with $\tau = 50$, we show that there exists values of τ for which PHuber-CE performs comparably to CE in the noise free case, and outperforms CE at high levels of label noise.

Dataset	**Loss function**	$\rho = 0.0$	$\rho = 0.2$	$\rho = 0.4$	$\rho = 0.6$
CIFAR-100	CE	75.4 ± 0.3	62.2 ± 0.4	45.8 ± 0.9	26.7 ± 0.1
	CE + clipping	23.5 ± 0.2	20.4 ± 0.4	16.2 ± 0.5	12.9 ± 0.1
	Linear	13.7 ± 0.7	8.2 ± 0.3	5.9 ± 0.7	3.9 ± 0.3
	GCE	73.3 ± 0.2	68.5 ± 0.3	59.5 ± 0.5	40.3 ± 0.4
	PHuber-CE $\tau = 10$	60.6 ± 1.1	54.8 ± 1.2	43.1 ± 1.1	24.3 ± 0.8
	PHuber-GCE $\tau = 10$	72.7 ± 0.1	68.4 ± 0.1	60.2 ± 0.2	42.2 ± 0.4
	PHuber-CE $\tau = 50$	75.4 ± 0.2	65.9 ± 0.2	49.1 ± 0.2	26.9 ± 0.0

Table 2: Reproduction of the CIFAR-100 experiments. The mean and standard error of the test accuracy over 3 trials is reported. The highlighted cells correspond to the best performing loss at a given ρ. PHuber-CE with $\tau = 50$ is an additional experiment which was not performed in the original paper.

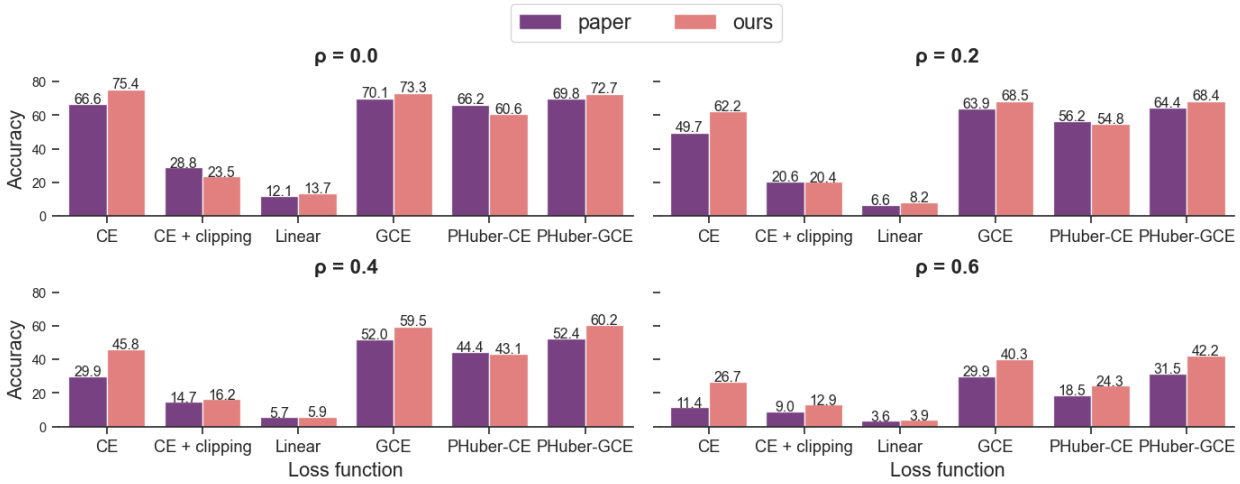


Figure 3: Test accuracy of ResNet-50 on CIFAR-100

6 Discussion

We will now discuss whether our experimental results support the claims of the paper.

For the synthetic experiments, when reusing exactly the parameters described in the paper and in their clarifications, our results do not exactly match those from the original paper for the Long & Servedio experiment. Nonetheless, these results still support the claim the authors made for these experiments, namely, that there exists label noise scenarios for both the Long and Servedio (2010) setting and the Ding (2013) setting which defeat a Huberised but not a partially Huberised loss. After experimenting with some of the parameters, we find values which produce results nearly identical to those shown in the paper. While we made a considerable effort to ensure that our implementation matches the paper's

description, the difference in results could be due to some minor differences in our implementations, or to a difference in the random seeds used to sample from the mixture model and flip the labels.

For the MNIST experiment, all losses, with the exception of CE with clipping, perform comparably, achieving very high accuracy for all levels of label noise. This differs from the results of the original paper, where CE and linear losses were affected by label noise. As a result, it is quite difficult to support or reject any claim made regarding these losses with this experiment.

For the CIFAR-100 experiment, even after fixing some of the hyper-parameters which were accidentally misreported in the original paper, our results differ from those of the paper, with our implementation yielding a noticeably higher test accuracy for the CE, GCE and PHuber-GCE losses. This is most likely due to a difference in the ResNet-50 architecture used, and to the deep learning framework used, as the authors mentioned using TensorFlow while we used PyTorch. It could also be due to the random seed used to add label noise, but we did not notice any significant difference in results when changing this seed on a subset of our experiments, as shown in Table 3. For the PHuber-GCE loss with $\tau = 10$, our reproduction supports the claim that with these specific values, it is competitive with the base loss in the noise free case, and can outperform it at high levels of label noise, but this does not hold for the PHuber-CE loss with $\tau = 10$, which performs worse than the CE loss in all cases. However, with our additional experiment, we show that there exists a value of τ for which the PHuber-CE loss performs comparably to CE in the noise free case, and improves upon it under label noise.

Our additional experiment shows that the value of τ plays a crucial role in the performance of partially Huberised losses. Both the PHuber-CE and GCE losses interpolate between the linear and the CE loss. For $\tau \to 1$ and $\alpha = 1$, PHuber-CE and GCE respectively mimic the linear loss, while for $\tau \to +\infty$ and $\alpha \to 0$, they mimic the CE loss. As the linear loss fails to train properly in our CIFAR-100 experiment, it is expected to obtain poor results for these losses if the tunable parameter used makes them too similar to the linear loss. As the PHuber-GCE combines both of these losses, it can also perform poorly in such a scenario. Furthermore, the CE loss with gradient clipping also has a tunable parameter τ which strongly affects performance, as shown in our reproduction.

In order to properly compare these losses, it would be therefore be of interest to find, for each level of label noise, the tunable parameter values for which they perform best, by using random search or a hyper-parameter tuning framework such as Optuna (Akiba et al., 2019) on a validation set. While such a hyper-parameter search has a high computational cost, it would offer some valuable insights on how well each of these losses perform, and how sensitive they are to changes to their tunable parameters.

Label noise	**Dataset**	**Loss function**	Seed 0	Seed 1	Seed 2	Seed 3	Original paper
$\rho = 0.6$	MNIST	CE	98.0 ± 0.1	97.9 ± 0.0	97.9 ± 0.0	97.9 ± 0.1	91.1 ± 0.6
		PHuber-GCE	98.0 ± 0.0	97.8 ± 0.1	98.0 ± 0.0	98.0 ± 0.0	97.8 ± 0.0
	CIFAR-100	CE	26.7 ± 0.1	27.1 ± 0.1	27.0 ± 0.5	26.9 ± 0.1	11.4 ± 0.2
		PHuber-GCE	42.2 ± 0.4	43.0 ± 0.2	42.1 ± 0.6	41.9 ± 0.1	31.5 ± 0.8

Table 3: Impact of the random seed used to generate label noise. The mean and standard error of the test accuracy over 3 trials is reported. This subset of experiments was chosen to include both a baseline and a partially Huberised loss, at the highest level of label noise. The results obtained are consistent across seeds, and differ from the original paper's results.

7 Conclusion

In this work, we fully re-implement the experiments performed in Menon et al. (2019). For the synthetic experiments, our results differ when using the exact values described in the paper, although they still support the main claim, and by slightly modifying some experiment settings, we obtain results almost identical to those of the original paper. Our results also differ for the deep learning experiments, with the baselines performing better than described. Nonetheless, these experiments still support the claim that partially Huberised losses perform well on real-world datasets subject to label noise. Our additional experiment also provides additional insight on the performance of partially Huberised losses, as it empirically shows that the value of τ can play an important role in the performance of models trained with these losses. We believe it would be of interest to reproduce the author's experiment on CIFAR-10 and to perform additional experiments focused on tuning these losses for different levels of label noise, although this would incur a relatively high computational cost. We leave such exploration for future work.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework, 2019.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Terrance DeVries and Graham W. Taylor. Improved Regularization of Convolutional Neural Networks with Cutout. *arXiv:1708.04552 [cs]*, November 2017. URL http://arxiv.org/abs/1708.04552. arXiv: 1708.04552.

Nan Ding. *Statistical machine learning in the t-exponential family of distributions*. PhD thesis, January 2013. URL https://docs.lib.purdue.edu/dissertations/AAI3591196.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2649-2. URL https://www.nature.com/articles/s41586-020-2649-2. Number: 7825 Publisher: Nature Publishing Group.

Elad Hazan, Kfir Levy, and Shai Shalev-Shwartz. Beyond convexity: Stochastic quasi-convex optimization. *Advances in neural information processing systems*, 28:1594–1602, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*, December 2015. URL http://arxiv.org/abs/1512.03385. arXiv: 1512.03385.

Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. ISSN 0003-4851. URL https://www.jstor.org/stable/2238020. Publisher: Institute of Mathematical Statistics.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. URL http://arxiv.org/abs/1412.6980. arXiv: 1412.6980.

Alex Krizhevsky and Geoffrey Hinton. Learning Multiple Layers of Features from Tiny Images. page 60, 2009.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. ISSN 1558-2256. doi: 10.1109/5.726791. Conference Name: Proceedings of the IEEE.

Kfir Y Levy. The power of normalization: Faster evasion of saddle points. *arXiv preprint arXiv:1611.04831*, 2016.

Junnan Li, Richard Socher, and Steven C. H. Hoi. DivideMix: Learning with Noisy Labels as Semi-supervised Learning. *arXiv:2002.07394 [cs]*, February 2020. URL http://arxiv.org/abs/2002.07394. arXiv: 2002.07394.

Kuang Liu. kuangliu/pytorch-cifar, December 2017. URL https://github.com/kuangliu/pytorch-cifar. original-date: 2017-01-21T05:43:20Z.

Philip M. Long and Rocco A. Servedio. Random classification noise defeats all convex potential boosters. *Machine Learning*, 78(3):287–304, March 2010. ISSN 1573-0565. doi: 10.1007/s10994-009-5165-z. URL https://doi.org/10.1007/s10994-009-5165-z.

Aditya Krishna Menon, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. Can gradient clipping mitigate label noise? September 2019. URL https://openreview.net/forum?id=rklB76EKPr.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed Precision Training. October 2017. URL https://arxiv.org/abs/1710.03740v3.

Y. E. Nesterov. A method for solving the convex programming problem with convergence rate O(1/k^2). *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983. URL https://ci.nii.ac.jp/naid/10029946121/.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv:1912.01703 [cs, stat]*, December 2019. URL http://arxiv.org/abs/1912.01703. arXiv: 1912.01703.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. page 14, 2013.

Brendan van Rooyen, Aditya Krishna Menon, and Robert C. Williamson. Learning with Symmetric Label Noise: The Importance of Being Unhinged. *arXiv:1505.07634 [cs]*, May 2015. URL http://arxiv.org/abs/1505.07634. arXiv: 1505.07634.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, March 2020. ISSN 1548-7105. doi: 10.1038/s41592-019-0686-2. URL https://www.nature.com/articles/s41592-019-0686-2. Number: 3 Publisher: Nature Publishing Group.

Omry Yadan. *Hydra - A framework for elegantly configuring complex applications*. 2019. URL https://github.com/facebookresearch/hydra.

Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *Procedings of the British Machine Vision Conference 2016*, pages 87.1–87.12, York, UK, 2016. British Machine Vision Association. ISBN 978-1-901725-59-9. doi: 10.5244/C.30.87. URL http://www.bmva.org/bmvc/2016/papers/paper087/index.html.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. *arXiv:1710.09412 [cs, stat]*, April 2018. URL http://arxiv.org/abs/1710.09412. arXiv: 1710.09412.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Analysis of gradient clipping and adaptive scaling with a relaxed smoothness condition. *arXiv preprint arXiv:1905.11881*, 2019a.

Michael R. Zhang, James Lucas, Geoffrey Hinton, and Jimmy Ba. Lookahead Optimizer: k steps forward, 1 step back. *arXiv:1907.08610 [cs, stat]*, December 2019b. URL http://arxiv.org/abs/1907.08610. arXiv: 1907.08610.

Zhilu Zhang and Mert R. Sabuncu. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. *arXiv:1805.07836 [cs, stat]*, November 2018. URL http://arxiv.org/abs/1805.07836. arXiv: 1805.07836.