Supplementary Materials

1. Co-PRL(L) Algorithm

We borrow the framework from the co-teaching framework(Han et al., 2018). The only difference is the filtering criteria. Co-teaching uses loss value as the filtering criteria while Co-PRL(L) uses the loss-layer-gradient norm as the filtering criteria.

Algorithm 1 Co-PRL(L)

input: initialize w_f and w_g, learning rate η, fixed τ, epoch T_k and T_{max}, iterations N_{max}
Return: model parameter w_f and w_g
for $T = 1, 2, ..., T_{max}$ **do**
  for $N = 1, ..., N_{max}$ **do**
    random sample a minibatch $\mathbf{M}$ from $\mathbf{D}_x, \mathbf{D}_y^\epsilon$ (noisy dataset)
    get the predicted label $\hat{\mathbf{Y}}_f$ and $\hat{\mathbf{Y}}_g$ from $\mathbf{M}$ by w_f. w_g
    calculate the individual loss $l_f = \mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}_f)$, $l_g = \mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}_g)$
    calculate the gradient norm of loss layer $score_f = \|\frac{\partial l_f}{\partial \hat{\mathbf{y}}_f}\|$, $score_g = \|\frac{\partial l_g}{\partial \hat{\mathbf{y}}_g}\|$.
    sample $R(T)\%$ small-loss-layer-gradient-norm instances by $score_f$ and $score_g$ to get $\mathbf{N}_f$, $\mathbf{N}_g$
    update $w_f = w_f - \eta \nabla_{w_f} \mathcal{L}(\mathbf{N}_f, w_f)$, $w_g = w_g - \eta \nabla_{w_g} \mathcal{L}(\mathbf{N}_g, w_g)$ (selected dataset)
    update model $\mathbf{x}_{t+1} = \mathbf{x}_t - \gamma_t \hat{\mu}$
  end for
  Update $R(T) = 1 - \min\left\{\frac{T}{T_k}\tau, \tau\right\}$
end for

2. Further Illustration of the difference between SPL and PRL(G)

In this section, we will further illustrate the difference between SPL and PRL(G). In order to have a more intuitive understanding of our algorithm, we could look at the Figure 1(a) and 1(b). Since we are in the agnostic label corruption setting, it is difficult to filtering out the correct corrupted data. We showed two situations when loss filtering failed and gradient filtering failed. As we could see that when loss filtering method failed, the remaining corrupted data could have large impact on the overall loss surface while when gradient filtering method failed, the remaining corrupted data only have limited impact on the overall loss surface, thus gaining robustness.

3. Networks and Hyperparameters

The hyperparameters are in Table 1. For Classification, we use the same hyperparameters in (Han et al., 2018). For CelebA, we use 3-layer fully connected network with 256 hidden nodes in hidden layer and leakly-relu as activation function. We also released our code in `https://github.com/illidanlab/PRL`.

Data\HyperParameter	BatchSize	Learning Rate	Optimizer	Momentum
CF-10	128	0.001	Adam	0.9
CF-100	128	0.001	Adam	0.9
CelebA	512	0.0003	Adam	0.9

Table 1. Main Hyperparmeters

Data	$\epsilon - 0.1$	$\epsilon - 0.05$	ϵ	$\epsilon + 0.05$	$\epsilon + 0.1$
CF10-Pair-45%	65.07±0.83	70.07±0.67	73.78±0.17	77.56±0.55	79.36±0.43
CF10-Sym-50%	69.21±0.35	72.53±0.45	75.43 ± 0.09	77.65±0.27	78.10±0.31
CF10-Sym-70%	53.88±0.64	58.49±0.97	60.26 ± 0.42	60.89±0.43	54.91±0.68
CF100-Pair-45%	32.60±0.45	34.17±0.40	34.43 ± 0.05	36.87±0.41	38.34±0.78
CF100-Sym-50%	37.74±0.41	39.72±0.36	40.64 ± 0.11	43.02±0.36	43.92±0.61
CF100-Sym-70%	24.40±0.47	25.50±0.45	27.27 ± 0.10	27.80±0.50	28.20±0.97

Table 2. sensitivity analysis for estimated ϵ

4. Learning Curve

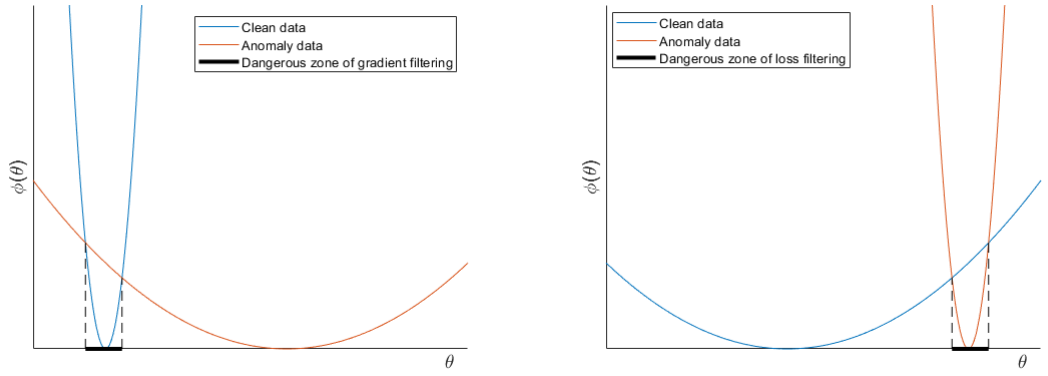
We show how testing evaluation changes along the training process for both classifcation and regression tasks in this section. The regression curve for CelebA data is showed in Figure 2. Note the for regression, the SPL and co-teaching are actually equivalent to our algorithm (i.e. PRL(L) and (Co-PRL(L))). The classification curve is in Figure 3.

5. Sensitivity Analysis

Since in real-world problems, it is hard to know that the ground-truth corruption rate, we perform the sensitivity analysis in classification tasks to show the effect of ϵ. The results are in Table 2. As we could see, the performance is stable if we overestimate the corruption rate, this is because only when we overestimate the ϵ, we could guarantee that the gradient norm of the remaining set is small. However, when we underestimate the corruption rate, in the worst case, there is no guarantee that the gradient norm of the remaining set is small. By using the empirical mean, even one large bad individual gradient would ruin the gradient estimation, and according to the convergence analysis of biased gradient descent, the final solution could be very bad in terms of clean data. That explains why to underestimate the corruption rate gives bad results. Also, from Table 2, we could see that using the ground truth corruption rate will lead to small uncertainty.

6. Empirical Results on Running Time

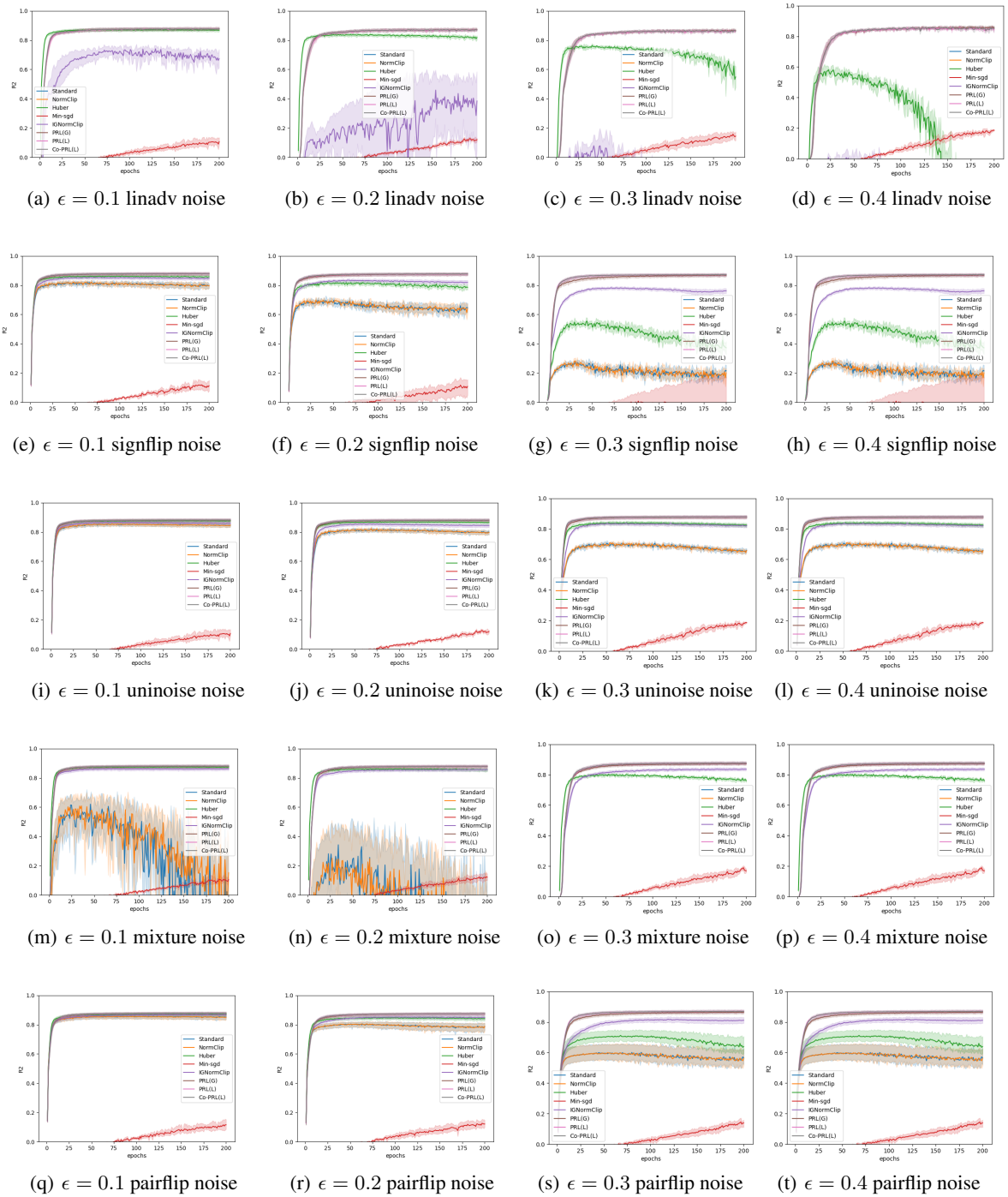
As we claimed in paper, the algorithm 2 (PRL(G)) is not efficient. In here we attached the execution time for one epoch for three different methods: ***Standard***, ***PRL(G)***, ***PRL(L)***. For fair comparison, we replace all batch normalization module to group normalization for this comparison, since it is hard to calculate individual gradient when using batch normalization. For PRL(G), we use opacus libarary to calculate the individual gradient. The results are showed in Table 3



(a) When gradient filtering method failed to pick out right corrupted data, the remaining corrupted data is relatively smooth, thus has limited impact on overall loss surface.

(b) When loss filtering method failed to pick out right corrupted data, the remaining corrupted data could be extremely sharp, thus has large impact on overall loss surface.

Figure 1. Further Illustration of difference between SPL and PRL(G)



(a) $\epsilon = 0.1$ linadv noise (b) $\epsilon = 0.2$ linadv noise (c) $\epsilon = 0.3$ linadv noise (d) $\epsilon = 0.4$ linadv noise

(e) $\epsilon = 0.1$ signflip noise (f) $\epsilon = 0.2$ signflip noise (g) $\epsilon = 0.3$ signflip noise (h) $\epsilon = 0.4$ signflip noise

(i) $\epsilon = 0.1$ uninoise noise (j) $\epsilon = 0.2$ uninoise noise (k) $\epsilon = 0.3$ uninoise noise (l) $\epsilon = 0.4$ uninoise noise

(m) $\epsilon = 0.1$ mixture noise (n) $\epsilon = 0.2$ mixture noise (o) $\epsilon = 0.3$ mixture noise (p) $\epsilon = 0.4$ mixture noise

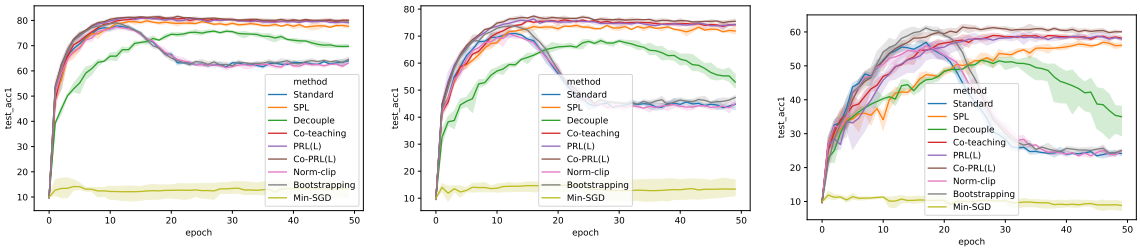
(q) $\epsilon = 0.1$ pairflip noise (r) $\epsilon = 0.2$ pairflip noise (s) $\epsilon = 0.3$ pairflip noise (t) $\epsilon = 0.4$ pairflip noise

Figure 2. Testing R-square for CelebA during the training phase.

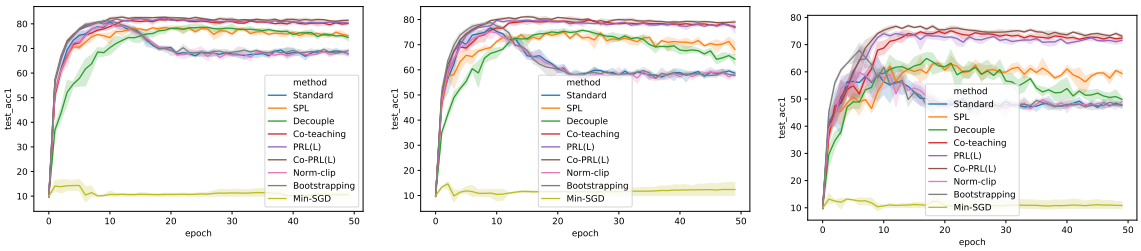
7. Proofs

7.1. Proof of Convergence of Biased SGD

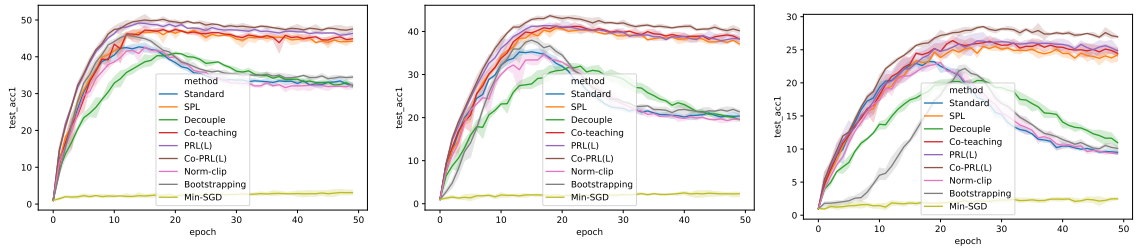
We gave the proof of the theorem of how biased gradient affect the final convergence of SGD. We introduce several assumptions and definition first:



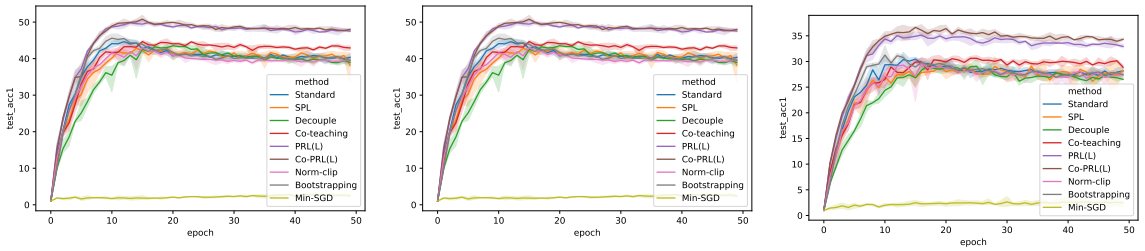
(a) CF10 $\epsilon = 0.3$ symmetric noise (b) CF10 $\epsilon = 0.5$ symmetric noise (c) CF10 $\epsilon = 0.7$ symmetric noisee



(d) CF10 $\epsilon = 0.25$ pairflip noise (e) CF10 $\epsilon = 0.35$ pairflip noise (f) CF10 $\epsilon = 0.45$ pairflip noise



(g) CF100 $\epsilon = 0.3$ symmetric noise (h) CF100 $\epsilon = 0.5$ symmetric noise (i) CF100 $\epsilon = 0.7$ symmetric noise



(j) CF100 $\epsilon = 0.25$ pairflip noise (k) CF100 $\epsilon = 0.35$ pairflip noise (l) CF100 $\epsilon = 0.45$ pairflip noise

Figure 3. CIFAR10 and CIFAR100 Testing Curve During Training. X axis represents the epoch number, Y axis represents the testing accuracy. The shadow represents the confidence interval, which is calculated across 3 random seed. As we see, PRL(L), and Co-PRL(L) are robust against different types of corruptions.

Assumption 1 (L-smoothness) *The function* ϕ*:* $\mathbb{R}^d \to \mathbb{R}$ *is differentiable and there exists a constant* $L > 0$ *such that for all* $\theta_1, \theta_2 \in \mathbb{R}^d$*, we have* $\phi(\theta_2) \leq \phi(\theta_1) + \langle \nabla\phi(\theta_1), \theta_2 - \theta_1 \rangle + \frac{L}{2}\|\theta_2 - \theta_1\|^2$

Definition 1 (Biased gradient oracle) *A map* $\mathbf{g} : \mathbb{R}^d \times \mathcal{D} \to \mathbb{R}^d$*, such that* $\mathbf{g}(\theta, \xi) = \nabla\phi(\theta) + \mathbf{b}(\theta, \xi) + \mathbf{n}(\theta, \xi)$ *for a bias* $\mathbf{b} : \mathbb{R}^d \to \mathbb{R}^d$ *and zero-mean noise* $\mathbf{n} : \mathbb{R}^d \times \mathcal{D} \to \mathbb{R}^d$*, that is* $\mathbb{E}_\xi \mathbf{n}(\theta, \xi) = 0$.

Compared to standard stochastic gradient oracle, the above definition introduces the bias term $\mathbf{b}$. In noisy-label settings, the $\mathbf{b}$ is generated by the data with corrupted labels.

Assumption 2 (σ-Bounded noise) *There exists constants* $\sigma > 0$*, such that* $\mathbb{E}_\xi \|\mathbf{n}(\theta, \xi)\|^2 \leq \sigma, \quad \forall \theta \in \mathbb{R}^d$

Method	Standard	PRL(G)	PRL(L)
CF10-Pair-45%	37.03s	145.55s	54.80s

Table 3. Execution Time of Single Epoch in CIFAR-10 Data

Assumption 3 (ζ-Bounded bias) *There exists constants $\zeta > 0$, such that for any ξ, we have* $\|\mathbf{b}(\theta, \xi)\|^2 \leq \zeta^2, \quad \forall \theta \in \mathbb{R}^d$

For simplicity, assume the learning rate is constant γ, then in every iteration, the biased SGD performs update $\theta_{t+1} \leftarrow \theta_t - \gamma_t \mathbf{g}(\theta_t, \xi)$. Then the following theorem showed the gradient norm convergence with biased SGD.

Theorem 1 (Convergence of Biased SGD(formal)) *Under assumptions 1, 2, 3, define $F = \phi(\theta_0) - \phi^*$and step size $\gamma = \min\left\{\frac{1}{L}, (\sqrt{\frac{LF}{\sigma T}})\right\}$, denote the desired accuracy as k, then*

$$T = \mathcal{O}\left(\frac{1}{k} + \frac{\sigma^2}{k^2}\right)$$

iterations are sufficient to obtain $\min_{t\in[T]} \mathbb{E}\left(\|\nabla\phi(\theta_t)\|^2\right) = \mathcal{O}(k + \zeta^2)$.

Remark 1 *Let $k = \zeta^2$, $T = \mathcal{O}\left(\frac{1}{\zeta^2} + \frac{\sigma^2}{\zeta^4}\right)$ iterations is sufficient to get* $\min_{t\in[T]} \mathbb{E}\left(\|\nabla\phi(\theta_t)\|^2\right) = \mathcal{O}(\zeta^2)$*, and performing more iterations does not improve the accuracy in terms of convergence.*

Since this is a standard results, more general results are showed in (Hu et al., 2020; Ajalloeian & Stich, 2020). For the sake of completeness, we provide the proof here.
Proof: by L-smooth, we have:

$$\phi(\theta_2) \leq \phi(\theta_1) + \langle \nabla\phi(\theta_1), \theta_2 - \theta_1\rangle + \frac{L}{2}\|\theta_2 - \theta_1\|^2$$

by using $\gamma \leq \frac{1}{L}$, we have

$$\begin{aligned}
\mathbb{E}\phi\left(\theta_{\mathbf{1}t+1}\right) &\leq \phi\left(\theta_{\mathbf{1}t}\right) - \gamma\left\langle\nabla\phi\left(\theta_{\mathbf{1}t}\right), \mathbb{E}\mathbf{g}_t\right\rangle + \frac{\gamma^2 L}{2}\left(\mathbb{E}\left\|\mathbf{g}_t - \mathbb{E}\mathbf{g}_t\right\|^2 + \mathbb{E}\left\|\mathbb{E}\mathbf{g}_t\right\|^2\right) \\
&= \phi\left(\theta_{\mathbf{1}t}\right) - \gamma\left\langle\nabla\phi\left(\theta_{\mathbf{1}t}\right), \nabla\phi\left(\theta_{\mathbf{1}t}\right) + \mathbf{b}_t\right\rangle + \frac{\gamma^2 L}{2}\left(\mathbb{E}\left\|\mathbf{n}_t\right\|^2 + \mathbb{E}\left\|\nabla\phi\left(\theta_{\mathbf{1}t}\right) + \mathbf{b}_t\right\|^2\right) \\
&\leq \phi\left(\theta_{\mathbf{1}t}\right) + \frac{\gamma}{2}\left(-2\left\langle\nabla\phi\left(\theta_{\mathbf{1}t}\right), \nabla\phi\left(\theta_{\mathbf{1}t}\right) + \mathbf{b}_t\right\rangle + \left\|\nabla\phi\left(\theta_{\mathbf{1}t}\right) + \mathbf{b}_t\right\|^2\right) + \frac{\gamma^2 L}{2}\mathbb{E}\left\|\mathbf{n}_t\right\|^2 \\
&= \phi\left(\theta_{\mathbf{1}t}\right) + \frac{\gamma}{2}\left(-\left\|\nabla\phi\left(\theta_{\mathbf{1}t}\right)\right\|^2 + \left\|\mathbf{b}_t\right\|^2\right) + \frac{\gamma^2 L}{2}\mathbb{E}\left\|\mathbf{n}_t\right\|^2
\end{aligned}$$

Since we have $\|\mathbf{b}_t\|^2 \leq \zeta^2$, $\|\mathbf{n}_t\|^2 \leq \sigma^2$, by plug in the learning rate constraint, we have

$$\begin{aligned}
\mathbb{E}\phi\left(\theta_{\mathbf{1}t+1}\right) &\leq \phi\left(\theta_{\mathbf{1}t}\right) - \frac{\gamma}{2}\left\|\nabla\phi\left(\theta_{\mathbf{1}t}\right)\right\|^2 + \frac{\gamma}{2}\zeta^2 + \frac{\gamma^2 L}{2}\sigma^2 \\
\mathbb{E}\phi\left(\theta_{\mathbf{1}t+1}\right) - \phi\left(\theta_{\mathbf{1}t}\right) &\leq -\frac{\gamma}{2}\left\|\nabla\phi\left(\theta_{\mathbf{1}t}\right)\right\|^2 + \frac{\gamma}{2}\zeta^2 + \frac{\gamma^2 L}{2}\sigma^2
\end{aligned}$$

Then, removing the gradient norm to left hand side, and sum it across different iterations, we could get

$$\frac{1}{2T}\sum_{t=0}^{T-1}\mathbb{E}\|\phi\left(\theta_{\mathbf{1}t}\right)\| \leq \frac{F}{T\gamma} + \frac{\zeta^2}{2} + \frac{\gamma L\sigma^2}{2}$$

Take the minimum respect to t and substitute the learning rate condition will directly get the results.

7.2. Proof of Corollary 1

We first prove the gradient estimation error.

Denote $\tilde{\mathbf{G}}$ to be the set of corrupted minibatch, $\mathbf{G}$ to be the set of original clean minibatch and we have $|\mathbf{G}| = |\tilde{\mathbf{G}}| = m$. Let $\mathbf{N}$ to be the set of remaining data and according to our algorithm, the remaining data has the size $|\mathbf{N}| = n = (1-\epsilon)m$. Define $\mathbf{A}$ to be the set of individual clean gradient, which is not discarded by algorithm 1. $\mathbf{B}$ to be the set of individual corrupted gradient, which is not discarded. According to our definition, we have $\mathbf{N} = \mathbf{A} \cup \mathbf{B}$. $\mathbf{AD}$ to be the set of individual good gradient, which is discarded, $\mathbf{AR}$ to be the set of individual good gradient, which is replaced by corrupted data. We have $\mathbf{G} = \mathbf{A} \cup \mathbf{AD} \cup \mathbf{AR}$. $\mathbf{BD}$ is the set of individual corrupted gradient, which is discarded by our algorithm. Denote the good gradient to be $\mathbf{g}_i = \alpha_i \mathbf{W}_i$, and the bad gradient to be $\tilde{\mathbf{g}}_i$, according to our assumption, we have $\|\tilde{\mathbf{g}}_i\| \leq L$.

Now, we have the l2 norm error:

$$\begin{aligned}
\|\mu(\mathbf{G}) - \mu(\mathbf{N})\| &= \|\frac{1}{m}\sum_{i\in\mathbf{G}}^{m}\mathbf{g}_i - \left(\frac{1}{n}\sum_{i\in\mathbf{A}}\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{B}}\tilde{\mathbf{g}}_i\right)\| \\
&= \|\frac{1}{n}\sum_{i=1}^{m}\frac{n}{m}\mathbf{g}_i - \left(\frac{1}{n}\sum_{i\in\mathbf{A}}\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{B}}\tilde{\mathbf{g}}_i\right)\| \\
&= \|\frac{1}{n}\sum_{i\in\mathbf{A}}\frac{n}{m}\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{AD}}\frac{n}{m}\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{AR}}\frac{n}{m}\mathbf{g}_i - \left(\frac{1}{n}\sum_{i\in\mathbf{A}}\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{B}}\tilde{\mathbf{g}}_i\right)\| \\
&= \|\frac{1}{n}\sum_{i\in\mathbf{A}}(\frac{n-m}{m})\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{AD}}\frac{n}{m}\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{AR}}\frac{n}{m}\mathbf{g}_i - \frac{1}{n}\sum_{i\in\mathbf{B}}\tilde{\mathbf{g}}_i\| \\
&\leq \|\frac{1}{n}\sum_{i\in\mathbf{A}}(\frac{n-m}{m})\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{AD}}\frac{n}{m}\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{AR}}\frac{n}{m}\mathbf{g}_i\| + \|\frac{1}{n}\sum_{i\in\mathbf{B}}\tilde{\mathbf{g}}_i\| \\
&\leq \|\sum_{\mathbf{A}}\frac{m-n}{nm}\mathbf{g}_i + \sum_{\mathbf{AD}}\frac{1}{m}\mathbf{g}_i + \sum_{\mathbf{AR}}\frac{1}{m}\mathbf{g}_i\| + \sum_{\mathbf{B}}\frac{1}{n}\|\tilde{\mathbf{g}}_i\| \\
&\leq \sum_{\mathbf{A}}\|\frac{m-n}{nm}\mathbf{g}_i\| + \sum_{\mathbf{AD}}\|\frac{1}{m}\mathbf{g}_i\| + \sum_{\mathbf{AR}}\|\frac{1}{m}\mathbf{g}_i\| + \sum_{\mathbf{B}}\frac{1}{n}\|\tilde{\mathbf{g}}_i\|
\end{aligned}$$

By using the filtering algorithm, we could guarantee that $\|\tilde{\mathbf{g}}_i\| \leq L$. Let $|\mathbf{A}| = x$, we have $|\mathbf{B}| = n - x = (1-\epsilon)m - x$, $|\mathbf{AR}| = m - n = \epsilon m$, $|\mathbf{AD}| = m - |\mathbf{A}| - |\mathbf{AR}| = m - x - (m-n) = n - x = (1-\epsilon)m - x$. Thus, we have:

$$\begin{aligned}
\|\mu(\mathbf{G}) - \mu(\mathbf{N})\| &\leq x\frac{m-n}{nm}L + (n-x)\frac{1}{m}L + (m-n)\frac{1}{m}L + (n-x)\frac{1}{n}L \\
&\leq x(\frac{m-n}{nm} - \frac{1}{m})L + n\frac{1}{m}L + (m-n)\frac{1}{m}L + (n-x)\frac{1}{n}L \\
&= \frac{1}{m}(\frac{2\epsilon-1}{1-\epsilon})xL + L + L - \frac{1}{n}xL \\
&= xL(\frac{2\epsilon-2}{n}) + 2L
\end{aligned}$$

To minimize the upper bound, we need x to be as small as possible since $2\epsilon - 2 < 1$. According to our problem setting, we have $x = n - m\epsilon \leq (1-2\epsilon)m$, substitute back we have:

$$\begin{aligned}
\|\mu(\mathbf{G}) - \mu(\mathbf{N})\| &\leq (1-2\epsilon)Lm(\frac{2\epsilon-2}{n}) + 2L \\
&= \frac{1-2\epsilon}{1-\epsilon}2L + 2L \\
&= 4L - \frac{\epsilon}{1-\epsilon}2L
\end{aligned}$$

Since $\epsilon < 0.5$, we use tylor expansion on $\frac{\epsilon}{1-\epsilon}$, by ignoring the high-order terms, we have

$$\|\mu(\mathbf{G}) - \mu(\mathbf{N})\| = \mathcal{O}(\epsilon L)$$

Note, if the Lipschitz continuous assumption does not hold, then L should be dimension dependent (i.e. $\sqrt{d}$).

Combining above gradient estimation error upper bound and Theorem 1, we could get the results in Corollary 1.

7.3. Proof of Randomized Filtering Algorithm

Lemma 1 (Gradient Estimation Error for Randomized Filtering) *Given a corrupted matrix $\tilde{\mathbf{G}} \in \mathbb{R}^{m\times d}$ generated in Problem 2. Let $\mathbf{G} \in \mathbb{R}^{m\times d}$ be the original clean gradient matrix. Suppose we are arbitrary select $n = (1-\epsilon)m$ rows from $\tilde{\mathbf{G}}$ to get remaining set $\mathbf{N} \in \mathbb{R}^{n\times d}$. Let μ to be the empirical mean function, assume the clean gradient before loss layer has bounded operator norm: $\|\mathbf{W}\|_{op} \leq C$, the maximum clean gradient in loss layer $\max_i \|\alpha_i\| = k$, the maximum corrupted gradient in loss layer $\max_i \|\delta_i\| = v$, assume $\epsilon < 0.5$, then we have:*

$$\|\mu(\mathbf{G}) - \mu(\mathbf{N})\| \leq Ck\frac{3\epsilon - 4\epsilon^2}{1-\epsilon} + Cv\frac{\epsilon}{1-\epsilon}$$

7.4. Proof of lemma 1

Denote $\tilde{\mathbf{G}}$ to be the set of corrupted minibatch, $\mathbf{G}$ to be the set of original clean minibatch and we have $|\mathbf{G}| = |\tilde{\mathbf{G}}| = m$. Let $\mathbf{N}$ to be the set of remaining data and according to our algorithm, the remaining data has the size $|\mathbf{N}| = n = (1-\epsilon)m$. Define $\mathbf{A}$ to be the set of individual clean gradient, which is not discarded by any filtering algorithm. $\mathbf{B}$ to be the set of individual corrupted gradient, which is not discarded. According to our definition, we have $\mathbf{N} = \mathbf{A} \cup \mathbf{B}$. $\mathbf{AD}$ to be the set of individual good gradient, which is discarded, $\mathbf{AR}$ to be the set of individual good gradient, which is replaced by corrupted data. We have $\mathbf{G} = \mathbf{A} \cup \mathbf{AD} \cup \mathbf{AR}$. $\mathbf{BD}$ is the set of individual corrupted gradient, which is discarded by our algorithm. Denote the good gradient to be $\mathbf{g}_i = \alpha_i \mathbf{W}_i$, and the bad gradient to be $\tilde{\mathbf{g}}_i = \delta_i \mathbf{W}_i$, according to our assumption, we have $\|\mathbf{W}_i\|_{op} \leq C$.

Now, we have the l2 norm error:

$$\begin{aligned}
\|\mu(\mathbf{G}) - \mu(\mathbf{N})\| &= \|\frac{1}{m}\sum_{i\in\mathbf{G}}^{m}\mathbf{g}_i - \left(\frac{1}{n}\sum_{i\in\mathbf{A}}\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{B}}\tilde{\mathbf{g}}_i\right)\| \\
&= \|\frac{1}{n}\sum_{i=1}^{m}\frac{n}{m}\mathbf{g}_i - \left(\frac{1}{n}\sum_{i\in\mathbf{A}}\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{B}}\tilde{\mathbf{g}}_i\right)\| \\
&= \|\frac{1}{n}\sum_{i\in\mathbf{A}}\frac{n}{m}\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{AD}}\frac{n}{m}\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{AR}}\frac{n}{m}\mathbf{g}_i - \left(\frac{1}{n}\sum_{i\in\mathbf{A}}\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{B}}\tilde{\mathbf{g}}_i\right)\| \\
&= \|\frac{1}{n}\sum_{i\in\mathbf{A}}(\frac{n-m}{m})\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{AD}}\frac{n}{m}\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{AR}}\frac{n}{m}\mathbf{g}_i - \frac{1}{n}\sum_{i\in\mathbf{B}}\tilde{\mathbf{g}}_i\| \\
&\leq \|\frac{1}{n}\sum_{i\in\mathbf{A}}(\frac{n-m}{m})\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{AD}}\frac{n}{m}\mathbf{g}_i + \frac{1}{n}\sum_{i\in\mathbf{AR}}\frac{n}{m}\mathbf{g}_i\| + \|\frac{1}{n}\sum_{i\in\mathbf{B}}\tilde{\mathbf{g}}_i\| \qquad (1)
\end{aligned}$$

Let $|\mathbf{A}| = x$, we have $|\mathbf{B}| = n - x = (1-\epsilon)m - x$, $|\mathbf{AR}| = m - n = \epsilon m$, $|\mathbf{AD}| = m - |\mathbf{A}| - |\mathbf{AR}| = m - x - (m-n) = n - x = (1-\epsilon)m - x$. Thus, we have:

$$\begin{aligned}
\|\mu(\mathbf{G}) - \mu(\mathbf{N})\| &\leq \|\sum_{\mathbf{A}}\frac{m-n}{nm}\mathbf{g}_i + \sum_{\mathbf{AD}}\frac{1}{m}\mathbf{g}_i + \sum_{\mathbf{AR}}\frac{1}{m}\mathbf{g}_i\| + \sum_{\mathbf{B}}\frac{1}{n}\|\tilde{\mathbf{g}}_i\| \\
&\leq \sum_{\mathbf{A}}\|\frac{m-n}{nm}\mathbf{g}_i\| + \sum_{\mathbf{AD}}\|\frac{1}{m}\mathbf{g}_i\| + \sum_{\mathbf{AR}}\|\frac{1}{m}\mathbf{g}_i\| + \sum_{\mathbf{B}}\frac{1}{n}\|\tilde{\mathbf{g}}_i\|
\end{aligned}$$

For individual gradient, according to the label corruption gradient definition in problem 2, assuming the $\|\mathbf{W}\|_{op} \leq C$, we have $\|\mathbf{g}_i\| \leq \|\alpha_i\|\|\mathbf{W}_i\|_{op} \leq C\|\alpha_i\|$. Also, denote $\max_i \|\alpha_i\| = k$, $\max_i \|\delta_i\| = v$, we have $\|\mathbf{g}_i\| \leq Ck$, $\|\tilde{\mathbf{g}}_i\| \leq Cv$.

$$\|\mu(\mathbf{G}) - \mu(\mathbf{N})\| \leq Cx\frac{m-n}{nm}k + C(n-x)\frac{1}{m}k + C(m-n)\frac{1}{m}k + C(n-x)\frac{1}{n}v$$

Note the above upper bound holds for any x, thus, we would like to get the minimum of the upper bound respect to x. Rearrange the term, we have

$$\begin{aligned}\|\mu(\mathbf{G}) - \mu(\mathbf{N})\| &\leq Cx(\frac{m-n}{nm} - \frac{1}{m})k + Cn\frac{1}{m}k + C(m-n)\frac{1}{m}k + C(n-x)\frac{1}{n}v \\ &= C\frac{1}{m}(\frac{2\epsilon-1}{1-\epsilon})xk + Ck + Cv - \frac{1}{n}Cxv \\ &= Cx\left(\frac{k(2\epsilon-1)}{m(1-\epsilon)} - \frac{v}{n}\right) + Ck + Cv \\ &= Cx\left(\frac{k(2\epsilon-1)-v}{m(1-\epsilon)}\right) + Ck + Cv\end{aligned}$$

Since when $\epsilon < 0.5$, $\frac{k(2\epsilon-1)-v}{m(1-\epsilon)} < 0$, we knew that x should be as small as possible to continue the bound. According to our algorithm, we knew $n - m\epsilon = m(1-\epsilon) - m\epsilon = (1-2\epsilon)m \leq x \leq n = (1-\epsilon)m$. Then, substitute $x = (1-2\epsilon)m$, we have

$$\begin{aligned}\|\mu(\mathbf{G}) - \mu(\mathbf{N})\| &\leq Ck(1-2\epsilon)\frac{2\epsilon-1}{1-\epsilon} + Ck + Cv - Cv\frac{1-2\epsilon}{1-\epsilon} \\ &= Ck\frac{3\epsilon-4\epsilon^2}{1-\epsilon} + Cv\frac{\epsilon}{1-\epsilon}\end{aligned}$$

7.5. Proof of Theorem 2

According to algorithm2, we could guarantee that $v \leq k$. By lemma 1, we will have:

$$\begin{aligned}\|\mu(\mathbf{G}) - \mu(\mathbf{N})\| &\leq Ck\frac{3\epsilon-4\epsilon^2}{1-\epsilon} + Cv\frac{\epsilon}{1-\epsilon} \\ &\leq Ck\frac{4\epsilon-4\epsilon^2}{1-\epsilon} \\ &= 4\epsilon Ck \\ &\approx \mathcal{O}(\epsilon\sqrt{q})\text{(C is constant, k is the norm of } q\text{-dimensional vector)}\end{aligned}$$

7.6. Proof of Lemma 2

Assume we have a d class label $\mathbf{y} \in \mathcal{R}^d$, where $y_k = 1, y_i = 0, i \neq k$. We have two prediction $\mathbf{p} \in \mathcal{R}^d$, $\mathbf{q} \in \mathcal{R}^d$.

Assume we have a d class label $\mathbf{y} \in \mathbb{R}^d$, where $y_k = 1, y_i = 0, i \neq k$. With little abuse of notation, suppose we have two prediction $\mathbf{p} \in \mathbb{R}^d$, $\mathbf{q} \in \mathbb{R}^d$. Without loss of generality, we could assume that $\mathbf{p}_1$ has smaller cross entropy loss, which indicates $\mathbf{p}_k \geq \mathbf{q}_k$

For MSE, assume we have opposite result

$$\begin{aligned}&\|\mathbf{p} - \mathbf{y}\|^2 \geq \|\mathbf{q} - \mathbf{y}\|^2 \\ \Rightarrow &\sum_{i\neq k} p_i^2 + (1-p_k)^2 \geq \sum_{i\neq k} q_i^2 + (1-q_k)^2\end{aligned} \tag{2}$$

For each $p_i, i \neq k$, We have

$$Var(p_i) = E(p_i^2) - E(p_i)^2 = \frac{1}{d-1}\sum_{i\neq k} p_i^2 - \frac{1}{(d-1)^2}(1-p_k)^2 \tag{3}$$

Then

$$\begin{aligned}
&\sum_{i\neq k} p_i^2 + (1-p_k)^2 \geq \sum_{i\neq k} q_i^2 + (1-q_k)^2 \\
\Rightarrow &Var_{i\neq k}(\mathbf{p}_i) + \frac{d}{(d-1)^2}(1-p_k)^2 \geq Var_{i\neq k}(\mathbf{q}_i) + \frac{d}{(d-1)^2}(1-q_k)^2 \\
\Rightarrow &Var_{i\neq k}(\mathbf{p}_i) - Var_{i\neq k}(\mathbf{q}_i) \geq \frac{d}{(d-1)^2}\left((1-q_k)^2 - (1-p_k)^2\right) \\
\Rightarrow &Var_{i\neq k}(\mathbf{p}_i) - Var_{i\neq k}(\mathbf{q}_i) \geq \frac{d}{(d-1)^2}\left((p_k-q_k)(2-p_k-q_k)\right)
\end{aligned} \tag{4}$$

References

Ajalloeian, A. and Stich, S. U. Analysis of sgd with biased gradient estimators. *arXiv preprint arXiv:2008.00051*, 2020.

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

Hu, Y., Zhang, S., Chen, X., and He, N. Biased stochastic gradient descent for conditional stochastic optimization. *arXiv preprint arXiv:2002.10790*, 2020.