SGD WITH HARDNESS WEIGHTED SAMPLING FOR DISTRIBUTIONALLY ROBUST DEEP LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Distributionally Robust Optimization (DRO) has been proposed as an alternative to Empirical Risk Minimization (ERM) in order to account for potential biases in the training data distribution. However, its use in deep learning has been severely restricted due to the relative inefficiency of the optimizers available for DRO in comparison to the wide-spread Stochastic Gradient Descent (SGD) based optimizers for deep learning with ERM. In this work, we demonstrate that SGD with hardness weighted sampling is a principled and efficient optimization method for DRO in machine learning and is particularly suited in the context of deep learning. Similar to a hard example mining strategy in essence and in practice, the proposed algorithm is straightforward to implement and computationally as efficient as SGD-based optimizers used for deep learning. It only requires adding a softmax layer and maintaining an history of the loss values for each training example to compute adaptive sampling probabilities. In contrast to typical ad hoc hard mining approaches, and exploiting recent theoretical results in deep learning optimization, we prove the convergence of our DRO algorithm for over-parameterized deep learning networks with ReLU activation and finite number of layers and parameters. Preliminary results demonstrate the feasibility and usefulness of our approach.

1 INTRODUCTION

In standard deep learning pipelines, a neural network h with parameters $\boldsymbol{\theta}$ is trained by minimizing the mean of a per-example loss $\mathcal{L}$ over a training dataset $\{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^{n}$. This corresponds to the empirical risk minimization optimization problem

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(h(\mathrm{x}_i; \boldsymbol{\theta}), \mathrm{y}_i\right) \tag{1}$$

Since the empirical risk is equal to the expectation of the per-example loss over the empirical training data distribution, an approximate solution of (1) can be obtained efficiently by Stochastic Gradient Descent (SGD) with a uniform sampling over the training data (Bottou et al., 2018).

This approach has led to spectacular results in term of average performance, but may generate outliers with high loss values compared to the average loss. Such cases can even be observed for elements belonging to the training data set. This is because approximate solutions obtained by SGD are prone to ignore a few hard examples in order to obtain a low mean per-example loss.

Far from being only of academic interest, outlier results have, for example, been consistently reported in the context of deep learning for brain tumor segmentation, as illustrated in the recent annual BRATS challenges (Bakas et al., 2018). For safety-critical systems, such as those used in healthcare, where outliers must be avoided, this is not satisfactory.

Efficient biased sampling methods, including hard example mining (Shrivastava et al., 2016; Loshchilov & Hutter, 2015; Chang et al., 2017) and weighted sampling (Bouchard et al., 2015; Berger et al., 2018; Gibson et al., 2018), have been proposed to mitigate this issue. However, even though these works typically start from an Empirical Risk Minimization formulation, it is not clear how those heuristics actually relate to Empirical Risk Minimization in theory.

Distributionally Robust Optimization (DRO) is an alternative to Empirical Risk Minimization (1) that takes into account uncertainty in the empirical data distribution. The deep neural network h is now trained by accounting for potential deviations from the empirical training data distribution. Formally, DRO corresponds to the min-max non-convex-concave optimization problem

$$\arg\min_{\boldsymbol{\theta}} \max_{q} \left(\sum_{i=1}^{n} q_i \, \mathcal{L}\left(h(\mathrm{x}_i; \boldsymbol{\theta}), \mathrm{y}_i\right) - \frac{1}{\beta} \sum_{i=1}^{n} \frac{1}{n} \phi\left(n q_i\right) \right) \tag{2}$$

where ϕ is a convex function that defines a ϕ-divergence (Csiszár et al., 2004), $q = (q_i)_{i=1}^{n}$ corresponds to arbitrary weighted sampling distributions over the training data, and $\beta > 0$ is a robustness parameter. Instead of minimizing the mean per-example loss on the training set, DRO seeks the hardest *weighted* empirical training data distribution around the (uniform) empirical training data distribution. This suggests a link between DRO and hard example mining.

The parameter β allows DRO to interpolate between ERM ($\beta \leftarrow 0$) and the minimization of maximum per-example loss ($\beta \leftarrow +\infty$). Motivations for using the minimization of maximum per-example loss for safety-critical applications has been discussed in (Shalev-Shwartz & Wexler, 2016).

DRO as a generalization of empirical risk minimization for machine learning has been studied in (Duchi et al., 2016; Rafique et al., 2018; Namkoong & Duchi, 2016; Chouzenoux et al., 2019), but still lacks optmization method as efficient as SGD in the non-convex setting of deep learning.

If one could solve the max problem in (2) for a given $\boldsymbol{\theta}$, DRO could be addressed by alternating between this max problem and a minimisation scheme akin to the standard Empirical Risk Minimization (1), but over an adaptively weighted empirical distribution. However, solving the max problem naively would require performing a forward pass over the entire training dataset. This can not be done at each iteration efficiently for large dataset. Previously proposed optimization methods for large-scale non-convex-concave problem of the form of (2) are based on the min-max structure of the problem, and consist in alternating between approximate maximization and minimization steps (Rafique et al., 2018; Lin et al., 2019; Jin et al., 2019). However, they differ from SGD methods for Empirical Risk Minimization by the introduction of additional hyperparameters for the optimizer such as a second learning rate and a ratio between the number of minimization and maximization steps. As a result, those alternate optimization methods are difficult to use as a replacement of Empirical Risk Minimization in practice.

In addition, from a practical perspective, those min-max methods do not use the link between DRO and adaptive weighted sampling, therefore departing from efficient heuristics used in hard example mining. From a theoretical perspective, they further make the assumption that the model is either smooth or weakly-convex, but none of those properties are true for the deep neural networks with ReLU activation that are largely used in practice.

In this work, we propose SGD with hardness weighted sampling, a novel, principled optimization method for training deep neural networks with Distributionally Robust Optimization inspired by hard example mining. Compared to SGD, our method only requires introducing an additional $\mathrm{softmax}$ layer and maintaining an history of the stale per-example loss to compute sampling probabilities over the training data. Since the loss is already computed at each iteration for SGD, our SGD with hardness weighted sampling is computationally as efficient as SGD methods for Empirical Risk Minimization. In practice, we show that our method performs favorably to SGD in the case of class imbalance despite using hyperparameters previously tuned for SGD.

We also formally link DRO in our method with hard example mining. As a result our method can be seen as a principled hard example mining approach. In this context, the robustness parameter β controls the trade-off between exploitation and exploration in the hard example mining process.

Last but not least, we generalize recent results in the convergence theory of SGD with ERM and over-parameterized deep learning networks with ReLU activation (Allen-Zhu et al., 2019; 2018; Cao & Gu, 2019; Zou & Gu, 2019) to our SGD with hardness weighted sampling for DRO. This is, to the best of our knowledge, the first convergence result for deep learning network with ReLU trained with DRO.

2 RELATED WORK IN DRO WITH A WASSERSTEIN DISTANCE

In this work, we focus on DRO with a ϕ-divergence (Csiszár et al., 2004). In this case, the data distributions that are considered in the DRO problem (2) are restricted to sharing the support of the empirical training distribution. In other words the weights assigned to the training data can change, but the training data itself remains unchanged.

Another popular formulation for DRO is DRO with a Wasserstein distance (Sinha et al., 2017; Duchi et al., 2016; Staib & Jegelka, 2017; Chouzenoux et al., 2019). In contrast to DRO with a ϕ-divergence, using a Wasserstein distance in DRO seeks to apply small data augmentation to the training data to make the deep learning model robust to small deformation of the data, but the sampling weights of the training data distribution typically remains unchanged. In this sense, DRO with a ϕ-divergence and DRO with a Wasserstein distance can be considered as orthogonal endeavours.

While we show that DRO with ϕ-divergence can be seen as a principled hard example mining method, it has been shown that DRO with a Wasserstein distance can be seen as a principled adversarial training method (Sinha et al., 2017; Staib & Jegelka, 2017).

Contrary to our SGD with hardness weighted sampling, the optimization methods of (Sinha et al., 2017; Staib & Jegelka, 2017) exploit alternate minimization maximization strategies that are orders of magnitude slower than SGD even for deep neural networks with smooth activation functions. In addition, the method of (Sinha et al., 2017) is NP-hard in the case of deep neural networks with ReLU activation functions.

3 MACHINE LEARNING WITH DISTRIBUTIONALLY ROBUST OPTIMISATION AND ϕ-DIVERGENCE

In machine learning based on Empirical Risk Minimization (ERM), a predictor h is trained using a training dataset $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ to perform well on average on a task for which the performance is measured on a per-example basis by a smooth criteria $\mathcal{L}$. Note that parameter regularization terms can easily be embedded in $\mathcal{L}$ since they are independent of the example. For ease of presentation, we focus on the supervised machine learning setting ,where $h : \boldsymbol{x} \mapsto \boldsymbol{y}$, and omit explicitly mentioning any parameter regularisation term.

Let $\Delta_n \subset \mathbb{R}^n$ be the set of empirical weighted training data distribution defined according to a given training dataset

$$\Delta_n = \left\{ p = (p_i)_{i=1}^n \in [0,1]^n, \; \sum_i p_i = 1 \right\} \tag{3}$$

and let $\hat{p}_{\mathrm{data}}$ be the corresponding uniform empirical training data distribution. In other words, Δ_n corresponds to all sampling weights that can be applied to the training dataset.

Let $\boldsymbol{\theta}$ be the set of parameters of the predictor $h(\,.\,;\boldsymbol{\theta}) : \boldsymbol{x} \mapsto \boldsymbol{y}$ we want to train. We assume $\mathcal{L}$ is a smooth and potentially non-convex function, and let $\boldsymbol{h} : \boldsymbol{\theta} \mapsto (h(\boldsymbol{x}_i;\boldsymbol{\theta}))_{i=1}^n$ be the vector of inferred outputs from the training data. By abuse of notation, we denote $\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta})) = (\mathcal{L}(h(\boldsymbol{x}_i;\boldsymbol{\theta}), \boldsymbol{y}_i))_{i=1}^n$.

Definition 3.1 (Mean Loss)**.**

$$M(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))) = \mathbb{E}_{\hat{p}_{\mathrm{data}}}\left[\mathcal{L}\left(h(\mathrm{x};\boldsymbol{\theta}), \boldsymbol{y}\right)\right] = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}\left(h(\mathrm{x}_i;\boldsymbol{\theta}), \boldsymbol{y}_i\right) \tag{4}$$

The ERM predictor, as used in most learning settings, is obtained by minimizing the mean loss (4).

Definition 3.2 (Empirical Risk Minimization (ERM) predictor)**.**

$$\tilde{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \; M(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))) \tag{5}$$

However, $\hat{p}_{\mathrm{data}}$ is typically biased compared to the true data distribution. Predictors trained with ERM are prone to fail on new examples that are not well represented in the training data.

Distributionally Robust Optimization (DRO) is an alternative to ERM that mitigates this issue by encouraging robustness to uncertainty in the empirical training data distribution. DRO, in its simplest form, is based on the notion of ϕ-divergence that we use to induce robustness with respect to all the empirical distributions Δ_n of the training dataset.

Definition 3.3 (ϕ-Divergence)**.** *Let $\phi : \mathbb{R}_+ \to \mathbb{R} \cup \{+\infty\}$ be a closed, convex, lower semi-continuous function such that $\forall z \in \mathbb{R}$, $\phi(z) \geq \phi(1) = 0$. The ϕ-Divergence D_ϕ is defined as, for all $p = (p_i)_{i=1}^n, q = (q_i)_{i=1}^n \in \Delta_n$*

$$D_\phi \left(q \| p\right) = \sum_{i=1}^{n} p_i \phi \left(\frac{q_i}{p_i}\right) \tag{6}$$

Example 3.1. *For $\phi : z \mapsto z \log(z)$, D_ϕ is the Kullback-Leibler (KL) divergence:*

$$D_\phi(q \| p) = D_{\mathrm{KL}}(q \| p) = \sum_{i=1}^{n} q_i \log \left(\frac{q_i}{p_i}\right) \tag{7}$$

And, for $\phi : z \mapsto (z-1)^2$, D_ϕ is the Pearson χ^2 divergence:

$$D_\phi(q \| p) = \chi^2(q \| p) = \sum_{i=1}^{n} \frac{\left(q_i - p_i\right)^2}{p_i} \tag{8}$$

Definition 3.4 (Distributionally Robust Loss)**.**

$$\begin{aligned} R(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))) &= \max_{q \in \Delta_n} \mathbb{E}_q \left[\mathcal{L}\left(h(\mathrm{x}; \boldsymbol{\theta}), \boldsymbol{y}\right)\right] - \frac{1}{\beta} D_\phi(q \| \hat{p}_{\mathrm{data}}) \\ &= \max_{q \in \Delta_n} \sum_i q_i \, \mathcal{L}\left(h(\mathrm{x}_i; \boldsymbol{\theta}), \boldsymbol{y}_i\right) - \frac{1}{n\beta} \sum_i \phi\left(n q_i\right) \end{aligned} \tag{9}$$

where $\beta > 0$ is a hyperparameter that controls the amount of robustness.

For a given ϕ-divergence, we define the DRO predictor, that is obtained by minimizing the distributionally robust loss (9) instead of the mean loss (4).

Definition 3.5 (Distributionally Robust Optimization (DRO) predictor)**.**

$$\tilde{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \; R(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))) \tag{10}$$

It is worth noting that DRO interpolates between ERM as $\beta \to 0$ and the minimization of the maximum loss as $\beta \to \infty$, and is equivalent to a mean-variance trade-off for β small (Gotoh et al., 2018).

$$\begin{cases} \max\limits_{q \in \Delta_n} \left(\mathbb{E}_q \left[\mathcal{L}(h\left(\mathrm{x}; \boldsymbol{\theta}\right), \mathrm{y})\right] - \dfrac{1}{\beta} D_\phi(q \| \hat{p}_{\mathrm{data}})\right) = \mathbb{E}_{\hat{p}_{\mathrm{data}}} \left[\mathcal{L}(h\left(\mathrm{x}; \boldsymbol{\theta}\right), \mathrm{y})\right] \\ \qquad\qquad + \dfrac{\beta}{2\phi''(1)} \mathbb{V}_{\hat{p}_{\mathrm{data}}} \left[\mathcal{L}(h\left(\mathrm{x}; \boldsymbol{\theta}\right), \boldsymbol{y})\right] + o(\beta) \\ \max\limits_{q \in \Delta_n} \left(\mathbb{E}_q \left[\mathcal{L}(h\left(\mathrm{x}; \boldsymbol{\theta}\right), \mathrm{y})\right] - \dfrac{1}{\beta} D_\phi(q \| \hat{p}_{\mathrm{data}})\right) \xrightarrow[\beta \to +\infty]{} \max\limits_i \mathcal{L}(h\left(\mathrm{x}_i; \boldsymbol{\theta}\right), \mathrm{y}_i) \end{cases} \tag{11}$$

where $\mathbb{V}_{\hat{p}_{\mathrm{data}}}$ is the empirical variance. In addition, we can see that when β is small, DRO results in optimizing the bias-variance tradeoff. This result is due to (Gotoh et al., 2018, Theorem 3.2). Furthermore, we observe that the distributionally robust loss (9) is an upper bound to the mean loss (4) (independently to the choice of ϕ and β), i.e. for all ϕ-divergence and all $\beta > 0$

$$\forall \boldsymbol{\theta}, \quad M(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))) \leq R(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))) \tag{12}$$

We now make assumptions for the ϕ-divergence to simplify the derivations in the remainder of the paper. Let D_ϕ be a ϕ-Divergence, and $c \in \mathbb{R}$, one can note that for $\phi_c : z \mapsto \phi(z) + c\,(1-z)$, we have $D_{\phi_c} = D_\phi$. As a result, if ϕ is differentiable in 1, we can assume without loss of generality that $\phi'(1) = 0$.

Assumption 3.1 (Regularity of ϕ). *$\phi : \mathbb{R}_+ \to \mathbb{R}$ is two times continuously differentiable on $[0, n]$, ρ-strongly convex, for $\rho > 0$, i.e.:*

$$\forall z, z' \in [0, n],\ \phi(z') \geq \phi(z) + \phi'(z)(z' - z) + \frac{\rho}{2}(z - z')^2$$

and ϕ satisfies:

$$\begin{cases} \forall z \in \mathbb{R},\ \phi(z) \geq \phi(1) = 0 \\ \phi'(1) = 0 \end{cases}$$

These assumptions are verified by most of the ϕ-divergence used in practice (e.g. the KL divergence and Pearson χ^2 divergence).

4 SGD WITH HARDNESS WEIGHTED SAMPLING

4.1 DISTRIBUTIONALLY ROBUST OPTIMIZATION WITH SGD AND ADAPTIVE SAMPLING

Existing optimization methods for DRO with a non-convex predictor h alternate between approximate minimization and maximization steps (Rafique et al., 2018; Jin et al., 2019; Lin et al., 2019), requiring the introduction of additional hyperparameters compared to SGD. These are difficult to tune in practice and convergence has not been proven for deep networks with ReLU activations.

In this section, we highlight properties that allows to link Distributionally Robust Optimization with SGD combined with adaptive sampling. Our analysis relies mainly on Fenchel duality (Moreau, 1965) and the notion of Fenchel conjugate (Fenchel, 1949) that we now define.

Definition 4.1 (Fenchel Conjugate Function). *Let $f : \mathbb{R}^m \to R \cup \{+\infty\}$ a proper function. The Fenchel conjugate of f is defined as $\forall \boldsymbol{v} \in \mathbb{R}^m,\ f^*(\boldsymbol{v}) = \max_{\boldsymbol{x} \in \mathbb{R}^m} \langle \boldsymbol{v}, \boldsymbol{x} \rangle - f(\boldsymbol{x})$*

Let

$$\forall p \in \mathbb{R}^n, \quad G(p) = \frac{1}{\beta} D_{\phi}(p \| p_{train}) + \delta_{\Delta_n}(p) \tag{13}$$

where δ_{Δ_n} is the characteristic function of the closed convex set Δ_n, i.e.

$$\forall p \in \mathbb{R}^n, \quad \delta_{\Delta_n}(p) = \begin{cases} 0 & \text{if } p \in \Delta_n \\ +\infty & \text{otherwise} \end{cases} \tag{14}$$

One can remark that the distributionally robust loss R (9) can be rewritten using the Fenchel conjugate function of G. This allows to obtain regularity properties for R.

Lemma 4.1 (Regularity of R). *If ϕ satisfies Assumption 3.1, then G and R satisfy the following:*

$$G \text{ is } \left(\frac{n\rho}{\beta}\right) \text{-strongly convex} \tag{15}$$

$$\forall \boldsymbol{\theta}, \quad R(\mathcal{L}(h(\boldsymbol{\theta}))) = \max_{q \in \mathbb{R}^n} \left(\langle \mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta})), q \rangle - G(q) \right) = G^* \left(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta})) \right) \tag{16}$$

$$R \text{ is } \left(\frac{\beta}{n\rho}\right) \text{-gradient Lipschitz continuous.} \tag{17}$$

Equation (16) follows from Definition 4.1. The proof of (15) and (17) is found in Appendix C.2.

According to (15), the optimization problem (16) is strictly convex and admit a unique solution in Δ_n. Let us denote this solution

$$\bar{p}(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))) := \arg\max_{q \in \mathbb{R}^n} \left(\langle \mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta})), q \rangle - G(q) \right) \tag{18}$$

The following lemma shows that the gradient, with respect to $\boldsymbol{\theta}$, of the distributionally robust loss (9) at a given $\boldsymbol{\theta}$ can be rewritten as the expectation, with respect to the weighted empirical distribution $\bar{p}(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta})))$, of the per-example loss gradient. We further show that straightforward analytical formulas exist for $\bar{p}$ when relying on classical ϕ-divergences. This result motivates our Algorithm 4.1 for efficient training with the distributionally robust loss.

Lemma 4.2 (Stochastic Gradient of the Distributionally Robust Loss)**.** *For all $\boldsymbol{\theta}$, we have*

$$\begin{aligned} \bar{p}(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))) &= \nabla_{\boldsymbol{v}} R(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))) \\ \nabla_{\boldsymbol{\theta}}(R \circ \mathcal{L} \circ \boldsymbol{h})(\boldsymbol{\theta}) &= \mathbb{E}_{\bar{p}(\mathcal{L}(h(\boldsymbol{\theta})))}\left[\nabla_{\boldsymbol{\theta}}\, \mathcal{L}\left(h(\mathrm{x}; \boldsymbol{\theta}), y\right)\right] \end{aligned} \tag{19}$$

where $\nabla_{\boldsymbol{v}} R$ is the gradient of R with respect to its input.

The proof is found in Appendix C.3. It is apparent from (19) that, given $\bar{p}$, an estimate of $\nabla_{\boldsymbol{\theta}} R$ could easily be provided by sampling a batch according to $\bar{p}$ and estimating the per-example loss gradients in the batch as per standard practice. We now provided closed-form formulas for $\bar{p}$ given $\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))$.

Example 4.1. *For the KL divergence (i.e. $\phi : z \mapsto z\log(z) - z + 1$), we have (see C.1 for a proof)*

$$\bar{p}(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))) = \mathrm{softmax}\left(\beta\, \mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))\right) \tag{20}$$

And for the Pearson χ^2 divergence (i.e. $\phi : z \mapsto (z-1)^2$), we have:

$$\forall i, \quad \bar{p}_i(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))) = \mathrm{ReLU}\left(\frac{1}{n}\left(1 + \frac{\beta}{2}\left(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))_i - \frac{1}{n}\sum_{j=1}^{n}\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))_j\right)\right)\right) \tag{21}$$

In both cases, we can verify consistency with (11) *as*

$$\forall i \in \{1, \dots, n\}, \quad \bar{p}_i(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))) \xrightarrow[\beta \to 0]{} \frac{1}{n} \tag{22}$$

Algorithm 1 SGD-HWS: SGD with Hardness Weighted Sampling for Kullback-Leibler DRO

Input: Training data $\{(x_i, y_i)\}_{i=1}^{n}$, number of epochs $T > 1$, robustness parameter $\beta > 0$, learning rate $\eta > 0$, batch size $b \in \{1, \dots, n\}$.
Initialization:
Initialise $\boldsymbol{\theta}$ randomly
Initialise the loss history $\tilde{\mathcal{L}} = -1$
Warm start:
// Split the training data into batches $\mathcal{B}$ and run 1 epoch with classic SGD
for $\{(x_i, y_i)\}_{i \in I}$ in $\mathcal{B}$ **do**
    // Run forward pass and store losses for all the samples in the batch
    for $i \in I$ **do**
        $\tilde{\mathcal{L}}_i \leftarrow \mathcal{L}(h(x_i; \boldsymbol{\theta}), y_i)$
    // Run backward pass and update the parameters of the model
    $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \frac{1}{b} \sum_{i \in I} \nabla_{\boldsymbol{\theta}}\, \mathcal{L}(h(x_i; \boldsymbol{\theta}), y_i)$
SGD with dynamic hardness weighted sampling:
for epoch $= 2, \dots, T$ **do**
    for iteration $i = 1, \dots, \left(\left\lfloor \frac{n}{b} \right\rfloor + 1\right)$ **do**
        // Run $\mathrm{softmax}$ to update the sampling probabilities of the samples
        $\hat{p} = \mathrm{softmax}(\beta \tilde{\mathcal{L}})$
        // Draw a batch with replacement using the probability distribution $\hat{p}$
        $\{(x_i, y_i)\}_{i \in I}$ such that $I \overset{\text{i.i.d.}}{\sim} \tilde{p}$ and $|I| = b$
        // Run forward pass and update losses for all the samples in the batch
        for $i \in I$ **do**
            $\tilde{\mathcal{L}}_i \leftarrow \mathcal{L}(h(x_i; \boldsymbol{\theta}), y_i)$
        // Run backward pass and update the parameters of the model
        $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \frac{1}{b} \sum_{i \in I} \nabla_{\boldsymbol{\theta}}\, \mathcal{L}(h(x_i; \boldsymbol{\theta}), y_i)$
Output: θ

4.2 Efficient Algorithm for Distributionally Robust Deep Learning

The second equality in (19) implies that $\nabla_{\boldsymbol{\theta}}\, \mathcal{L}(h_i(\boldsymbol{\theta}), \boldsymbol{y}_i)$ is an unbiased estimator of the distributionally robust loss $R\left(\mathcal{L}(h(\boldsymbol{\theta}))\right)$ when i is sampled with respect to $\bar{p}(\mathcal{L}(h(\boldsymbol{\theta})))$. This suggests that the Distributionally robust loss can be minimized efficiently by Stochastic Gradient Descent by sampling mini-batches with respect to $\bar{p}(\mathcal{L}(h(\boldsymbol{\theta})))$ at each iteration. However, even though closed-form formulas were provided for $\bar{p}$, evaluating exactly $\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))$, i.e. doing one forward pass on the whole training set at each iteration, is computationally prohibitive for large dataset.

In practice, we propose to use a stale version of $\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))$ by maintaining online a history of the loss values of the training examples during training $\left(\mathcal{L}(h(\boldsymbol{x}_i;\boldsymbol{\theta}^{(t_i)}),\boldsymbol{y}_i)\right)$. Where for all i, t_i is the last iteration at which the per-example loss of example i has been computed. Using the the Kullback-Leibler divergence as ϕ-divergence, this leads to the Stochastic Gradient Descent with KL hardness weighted sampling algorithm proposed in Algorithm 4.1.

In contrast to alternate min-max optimization methods, our SGD with a adaptive sampling strategy is similar to the SGD-based optimizers used by the vast majority of deep learning practitioners (e.g. SGD, SGD with momentum, ADAM). Compared to standard SGD-based training optimizers for the mean loss, our algorithm requires only an additional $\mathrm{softmax}$ operation per iteration and to store an additional vector of size n (number of training examples), thereby making it ideally suited for deeep learning applications.

4.3 DRO AS PRINCIPLED HARD EXAMPLE MINING

In this section, we discuss the relationship between DRO and hard example mining. SGD with an ad hoc adaptive sampling strategy is already used in practice while starting from a mean loss optimization formulation in the hard example mining literature (Loshchilov & Hutter, 2015; Shrivastava et al., 2016). Similarly to our algorithm, in hard example mining heuristics, the *hard examples*, those training examples with high values of the loss are sampled more often. We formalize this in the following definition for hard example mining sampling strategies.

Definition 4.2 (Hard Example Mining Sampling)**.** *Any adaptive sampling method such that the probability p_i of sampling example x_i is an non-decreasing function of the (potentially stale) loss value associated with x_i.*

Theorem 4.1. *The proposed hardness weighted sampling is a hard example mining sampling for any ϕ-Divergence that satisfies Assumption 3.1. In addition, the probability p_i of sampling example x_i is an non-increasing function of the loss value associated with x_j for all $j \neq i$.*

See Appendix C.4 for the proof. The second part of Theorem 4.1 implies that as the loss of an example diminishes, the sampling probabilities of all the other examples increase. As a result, Distributionally Robust Optimization balances exploration and exploitation.

5 CONVERGENCE OF SGD WITH HARDNESS WEIGHTED SAMPLING FOR OVER-PARAMETERIZED DEEP NEURAL NETWORKS WITH ReLU

Convergence results for over-parameterized deep learning has recently been proposed in (Allen-Zhu et al., 2019). It gives convergence guarantees for deep neural networks h with any activation function (including ReLU), and with any (finite) number of layers L and parameters m, under the assumption that m is large enough. Although some results suggest that this theory cannot explain all the properties of deep learning observed in practice (Chizat et al., 2019), at the time of writing, this is the most realistic setting for which a convergence theory of deep learning exists.

In this section, we demonstrate the first convergence guarantees for deep neural networks with ReLU trained with DRO. Our analysis is based on the results developed in (Allen-Zhu et al., 2019) which is a simplified version of (Allen-Zhu et al., 2018). Improving on those theoretical results would automatically improves our results as well. We focus at providing theoretical tools that could be used to generalize any convergence result for ERM using SGD to DRO using Algorithm 4.1.

Let us first state our assumptions on the neural network h, and the per-example loss function $\mathcal{L}$.

Assumption 5.1 (Deep Neural Network)**.** *In this section, we use the following notations and assumptions similar to (Allen-Zhu et al., 2019):*

- *h is a fully connected neural network with $L+2$ layers, ReLU activation function, and m nodes in each hidden layers*

- *For all $i \in \{1, \dots, n\}$, we denote $h_i : \boldsymbol{\theta} \mapsto h_i(x_i; \boldsymbol{\theta})$ the output d dimensional scores of h applied to example x_i of dimension $\mathfrak{d}$.*
- *$\boldsymbol{\theta} = (\boldsymbol{\theta}_l)_{l=0}^{L+1}$ is the set of parameters of the neural network h, where $\boldsymbol{\theta}_l$ is the set of weights for layer l with $\boldsymbol{\theta}_0 \in \mathbb{R}^{\mathfrak{d} \times m}$, $\boldsymbol{\theta}_{L+1} \in \mathbb{R}^{m \times d}$, and $\boldsymbol{\theta}_l \in \mathbb{R}^{m \times m}$ for any other l.*
- *(Data separation) It exists $\delta > 0$ such that for all $i, j \in \{1, \dots, n\}$, if $i \neq j$, $\|x_i - x_j\| \geq \delta$.*
- *We assume $m \geq \Omega(d \times \text{poly}(n, L, \delta^{-1}))$ for some sufficiently large polynomial* poly*, and $\delta \geq O\left(\frac{1}{L}\right)$. We refer the reader to (Allen-Zhu et al., 2019) for details about* poly.
- *The parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_l)_{l=0}^{L+1}$ are initialized at random such that:*
  - $[\boldsymbol{\theta}_0]_{i,j} \sim \mathcal{N}\left(0, \frac{2}{m}\right)$ *for every* $(i, j) \in \{1, \dots, m\} \times \{1, \dots, \mathfrak{d}\}$
  - $[\boldsymbol{\theta}_l]_{i,j} \sim \mathcal{N}\left(0, \frac{2}{m}\right)$ *for every* $(i, j) \in \{1, \dots, m\}^2$ *and* $l \in \{1, \dots, L\}$
  - $[\boldsymbol{\theta}_{L+1}]_{i,j} \sim \mathcal{N}\left(0, \frac{1}{d}\right)$ *for every* $(i, j) \in \{1, \dots, d\} \times \{1, \dots, m\}$

Assumption 5.2 (Regularity of $\mathcal{L}$)**.** *For all i, $\mathcal{L}_i$ is a $C(\nabla \mathcal{L})$-gradient Lipschitz continous, $C(\mathcal{L})$-Lipschitz continous, and bounded (potentially non-convex) function.*

We first generalize the converce of SGD in (Allen-Zhu et al., 2019, Theorem 2) to the minimization of the distributionally robust loss using SGD and an *exact* hardness weighted sampling (19), i.e. with an exact non-stale loss history.

Theorem 5.1 (Convergence of Robust SGD with exact Loss History)**.** *Let batch size $1 \leq b \leq n$, and $\epsilon > 0$. Suppose there exists constants C_1, C_2, $C_3 > 0$ such that the number of hidden units satisfies $m \geq C_1(d\epsilon^{-1} \times \text{poly}(n, L, \delta^{-1}))$, $\delta \geq \left(\frac{C_2}{L}\right)$, and the learning rate be $\eta_{exact} = C_3\left(\min\left(1, \frac{\alpha n^2 \rho}{\beta C(\mathcal{L})^2 + 2n\rho C(\nabla \mathcal{L})}\right) \times \frac{b\delta d}{\text{poly}(n, L) m \log^2(m)}\right)$. There exists constants C_4, $C_5 > 0$ such that with probability at least $1 - \exp\left(-C_4(\log^2(m))\right)$ over the randomness of the initialization and the mini-batches, Robust SGD with exact loss vector finds $\|\nabla_{\boldsymbol{\theta}}(R \circ f \circ h)(\boldsymbol{\theta})\| \leq \epsilon$ after $T = C_5\left(\frac{Ln^3}{\eta_{exact}\delta\epsilon^2}\right)$ iterations.*

where $\alpha = \min_{\boldsymbol{\theta}} \min_i \bar{p}_i(\mathcal{L}(\boldsymbol{\theta}))$ is lower bound on the sampling probabilities. For the Kullback-Leibler ϕ-divergence, and for any ϕ-divergence satisfying assumption 3.1 with a robustness parameter β small enough, we have $\alpha > 0$. We refer the reader to (Allen-Zhu et al., 2019, Theorem 2) for the values of the constants C_1, C_2, C_3, C_4, C_5 and the definitions of the polynoms. Compared to (Allen-Zhu et al., 2019, Theorem 2) only the learning rate differs. The $\min(1, .)$ operation in the formula for η_{exact} allows to guarantee that $\eta_{exact} \leq \eta'$ where η' is the learning rate of in (Allen-Zhu et al., 2019, Theorem 2). The proof can be found in Appendix C.6.3.

It is worth noting that for the KL ϕ-divergence, $\rho = \frac{1}{n}$. In addition, in the limit $\beta \to 0$, which corresponds to ERM, we have $\alpha \to \frac{1}{n}$. As a result, we recover exactly Theorem 2 of (Allen-Zhu et al., 2019) as extended in their Appendix A for any $\mathcal{L}$ that satisfies assumption 5.2 with $C(\nabla \mathcal{L}) = 1$.

When the amount of distributionally robustness increases the sampling differs more and more from the uniform sampling and becomes more sensitive to changes of the loss distribution. One way to mitigate this issue is to reduce the learning rate. The conditions of Theorem 5.1 are consistent with this observation since when β increases, α and η_{exact} decreases.

In practice in algorithm 4.1, we have access only to a stale loss history. We know restate the convergence of Robust SGD with a stale loss history and a warm-up as in Algorithm 4.1.

Theorem 5.2 (Convergence of Robust SGD with Stale Loss History and warm-up)**.** *Let batch size $1 \leq b \leq n$, and $\epsilon > 0$. Under the conditions of Theorem 5.1, the same notations, and with the learning rate $\eta_{stale} = C_6 \min\left(1, \frac{\alpha\rho d^{3/2}\delta b \log\left(\frac{1}{1-\alpha}\right)}{\beta C(\mathcal{L}) A(\nabla \mathcal{L}) L m^{3/2} n^{3/2} \log^2(m)}\right) \times \eta_{exact}$ for a constant $C_6 > 0$. With probability at least $1 - \exp\left(-C_4(\log^2(m))\right)$ over the randomness of the initialization and the mini-batches, Robust SGD with exact loss vector finds $\|\nabla_{\boldsymbol{\theta}}(R \circ f \circ h)(\boldsymbol{\theta})\| \leq \epsilon$ after $T = C_5\left(\frac{Ln^3}{\eta_{stale}\delta\epsilon^2}\right)$ iterations.*

Figure 1: Comparison of learning curves for ERM with SGD (blue) and DRO with our SGD with hardness weighted sampling (orange: $\beta = 0.1$, green: $\beta = 0.3$) on CIFAR10. The models are trained on an imbalanced CIFAR10 dataset (only 10% of the cats kept in the training dataset) and evaluated on the original CIFAR10 testing dataset.

Where $C(\mathcal{L}) > 0$ is a constant such that $\mathcal{L}$ is $C(\mathcal{L})$-Lipschitz continuous, and $A(\nabla \mathcal{L}) > 0$ is a constant that bound the gradient of $\mathcal{L}$ with respect to its input. $C(\mathcal{L})$ and $A(\nabla \mathcal{L})$ are guaranteed to exist under assumptions 5.1. The proof can be found in Appendix C.7.

Compared to Theorem 5.1 only the learning rate differs. Similarly to Theorem 5.1, when β tends to zero we recover Theorem 2 of (Allen-Zhu et al., 2019).

It is worth noting that when β increases, $\frac{\alpha \rho d^{3/2} \delta b \log\left(\frac{1}{1-\alpha}\right)}{\beta C(\mathcal{L}) A(\nabla \mathcal{L}) L m^{3/2} n^{3/2} \log^2(m)}$ decreases. This implies that η_{stale} decreases faster than η_{exact} when β increases. This was to be expected since the error that is made by using the stale loss history instead of the exact loss increases when β increases.

6 Experiments

We now illustrate the properties of our SGD with hardness weighted sampling described in Algorithm 4.1 for training deep neural networks with ReLU activation functions with DRO (10).

6.1 Experiments on MNIST

We create a bias between training and testing data distribution of MNIST (LeCun, 1998) by keeping only 1% of the digits 3 in the training dataset, while the testing dataset remains unchanged. Implementation details can be found in Appendix A.1.

The learning curves in figure 2 of Appendix A.1, computed using the original testing MNIST dataset, shows that our method outperforms ERM for a large range of values.

Furthermore, the variations of learning curves with β are consistent with our theoretical insight in Theorem 5.2. As β decreases to 0, the learning curves of DRO with our method converges to the learning curve of ERM with SGD. For large value of β the learning curve becomes instable. This is because we use the same learning rate for all our experiments, but according to Theorem 5.2, the learning rate should be reduced as β increases.

6.2 Experiments on CIFAR10

We now show that our method for training deep neural network with DRO is not only more robust to bias in the training dataset, but can also be used with a higher learning rate than ERM.

We create a bias between training and testing data distribution in CIFAR10 (Krizhevsky et al., 2010) by keeping only 10% of the cats in the CIFAR10 training dataset, while the CIFAR10 testing dataset remains unchanged.

We used the state-of-the-art WRN-28-10 deep neural network architecture proposed in (Zagoruyko & Komodakis, 2016). We kept the hyperparameters and data augmentation, except that we used a higher learning rate equal to 1 (see Appendix A.2 for more details).

The learning curves of figure 1, computed on the CIFAR10 testing dataset, shows that our methods outperforms the baseline. This suggests that replacing SGD-based optimizer by our Algorithm 4.1 in an existing deep learning pipeline is straightforward, and that our method is robust to the choice of the learning rate.

7 Conclusion and Discussion

We have shown that efficient training of deep neural networks with Distributionally Robust Optimization (DRO) with a ϕ-divergence (10) is possible. Our Stochastic Gradient Descent (SGD) with hardness weighted sampling is a principled hard example mining method. It is as straightforward to implement, and as computationally efficient as SGD for Empirical Risk Minimization. It can be used for deep neural networks with any activation function (including ReLU), and with any per-example loss function. In addition, we prove the convergence of our method for over-parameterized deep neural networks. This is, to the best of our knowledge, the first convergence result for training a deep neural network based on DRO.

Our experiments on MNIST illustrate several behaviours that were predicted by our Theorem 5.2. Our experiments on CIFAR10 illustrates that our SGD with hardness weighted sampling can be used successfully as a replacement of SGD in a state-of-the-art deep learning pipeline that was tuned for SGD. In addition, our results on CIFAR10 suggest that our method is more robust than SGD to bias in the training data distribution and to the choice of the learning rate. The fact that our algorithm can be used with a higher learning rate than SGD goes beyond the prediction of our theoretical results. Investigating this advantage is left for future work.

References

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. On the convergence rate of training recurrent neural networks. *arXiv preprint arXiv:1810.12065*, 2018.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *ICML*, pp. 242–252, 2019.

Spyridon Bakas, Mauricio Reyes, Andras Jakab, Stefan Bauer, Markus Rempfler, Alessandro Crimi, Russell Takeshi Shinohara, Christoph Berger, Sung Min Ha, Martin Rozycki, et al. Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge. *arXiv preprint arXiv:1811.02629*, 2018.

Lorenz Berger, Hyde Eoin, M Jorge Cardoso, and Sébastien Ourselin. An adaptive sampling scheme to efficiently train fully convolutional networks for semantic segmentation. In *Annual Conference on Medical Image Understanding and Analysis*, pp. 277–286. Springer, 2018.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

Guillaume Bouchard, Théo Trouillon, Julien Perez, and Adrien Gaidon. Online learning to sample. *arXiv preprint arXiv:1506.09016*, 2015.

Yuan Cao and Quanquan Gu. A generalization theory of gradient descent for learning over-parameterized deep relu networks. *arXiv preprint arXiv:1902.01384*, 2019.

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems*, pp. 1002–1012, 2017.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *arXiv preprint arXiv:1812.07956*, 2019.

Emilie Chouzenoux, Henri Gérard, and Jean-Christophe Pesquet. General risk measures for robust machine learning. *arXiv preprint arXiv:1904.11707*, 2019.

Imre Csiszár, Paul C Shields, et al. Information theory and statistics: A tutorial. *Foundations and Trends® in Communications and Information Theory*, 1(4):417–528, 2004.

John Duchi, Peter Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv preprint arXiv:1610.03425*, 2016.

Werner Fenchel. On conjugate convex functions. *Canadian Journal of Mathematics*, 1(1):73–77, 1949.

Eli Gibson, Wenqi Li, Carole Sudre, Lucas Fidon, Dzhoshkun I Shakir, Guotai Wang, Zach Eaton-Rosen, Robert Gray, Tom Doel, Yipeng Hu, et al. Niftynet: a deep-learning platform for medical imaging. *Computer methods and programs in biomedicine*, 158:113–122, 2018.

Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

Jun-ya Gotoh, Michael Jong Kim, and Andrew EB Lim. Robust empirical optimization is almost the same as mean–variance optimization. *Operations research letters*, 46(4):448–452, 2018.

Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex analysis and minimization algorithms I: Fundamentals*, volume 305. Springer science & business media, 2013.

Chi Jin, Praneeth Netrapalli, and Michael I Jordan. Minmax optimization: Stable limit points of gradient descent ascent are locally optimal. *arXiv preprint arXiv:1902.00618*, 2019.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 8, 2010.

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

Tianyi Lin, Chi Jin, and Michael I Jordan. On gradient descent ascent for nonconvex-concave minimax problems. *arXiv preprint arXiv:1906.00331*, 2019.

Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. *arXiv preprint arXiv:1511.06343*, 2015.

Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Advances in Neural Information Processing Systems*, pp. 2208–2216, 2016.

Hassan Rafique, Mingrui Liu, Qihang Lin, and Tianbao Yang. Non-convex min-max optimization: Provable algorithms and applications in machine learning. *arXiv preprint arXiv:1810.02060*, 2018.

Shai Shalev-Shwartz and Yonatan Wexler. Minimizing the maximal loss: How and why. In *ICML*, pp. 793–801, 2016.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 761–769, 2016.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

Matthew Staib and Stefanie Jegelka. Distributionally robust deep learning as a generalization of adversarial training. In *NIPS workshop on Machine Learning and Computer Security*, 2017.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Difan Zou and Quanquan Gu. An improved analysis of training over-parameterized deep neural networks. *arXiv preprint arXiv:1906.04688*, 2019.

A MORE ON OUR EXPERIMENTS

A.1 EXPERIMENTS ON MNIST

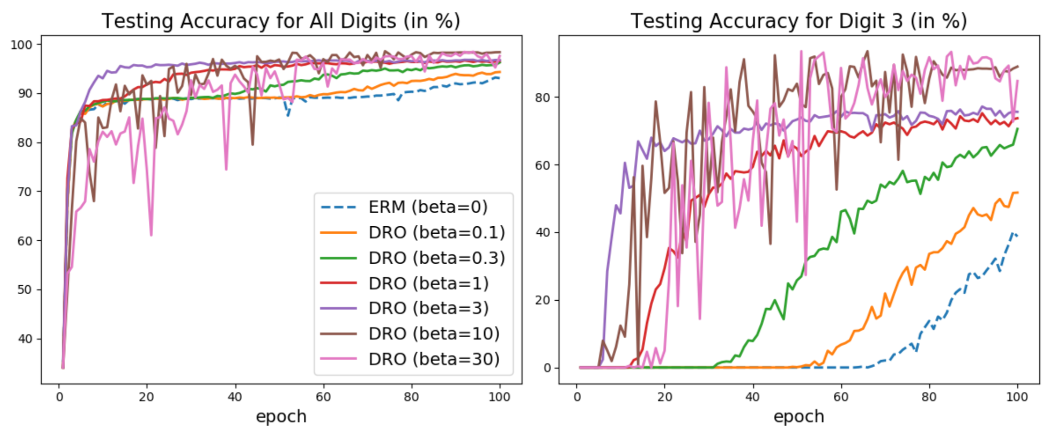

Figure 2: Comparison of learning curves for ERM with SGD (blue) and DRO with our SGD with hardness weighted sampling for different values of β on MNIST ($\beta = 0.1$, $\beta = 0.3$, $\beta = 1$, $\beta = 3$, $\beta = 10$, $\beta = 30$). The models are trained on an imbalanced MNIST dataset (only 1% of the digits 3 kept in the training datatset) and evaluated on the original MNIST testing dataset.

A.1.1 IMPLEMENTATION DETAILS

For our experiments on MNIST, we used a Wide Residual Network (WRN) (Zagoruyko & Komodakis, 2016). The familly of WRN models has proved to be very efficient and flexible, achieving state-of-the-art accuracy on several dataset. More specifically, we used WRN-16-1 (see Zagoruyko & Komodakis, 2016, section 2.3).

For the optimization we used a learning rate of 0.01. No momentum or weight decay were used. No data augmentation was used.

A.1.2 COMMENT ON EARLY-STOPPING WITH ERM AND DRO

Figure 2 suggests that if we train ERM long enough it will converge to the same accuracy as DRO. It is worth noting that we don't use the knowledge of the training dataset bias in our method. For a less obvious bias, we would have access only to the global measure of accuracy (in our case the left panel figure 2). In this situation, with early-stopping we would stop the training of ERM when it plateaus, i.e. we would obtain a model that achieved in accuracy of 0 on an under-representated part of the training dataset.

This suggests two things. First, the mean accuracy is not a good criteria to decide when to stop the training of ERM for safety-critical systems. Second, our SGD with hardness weighted sampling converge faster than SGD to a safe solution, and is more robust to early-stopping.

A.2 EXPERIMENTS ON CIFAR10

A.3 IMPLEMENTATION DETAILS

For our experiments on the CIFAR10 dataset (Krizhevsky et al., 2010), we used the WideResNet WRN-28-10 as described in (Zagoruyko & Komodakis, 2016).

We used the same pipeline as in (Zagoruyko & Komodakis, 2016), that we summarized here:

- we used SGD with momentum $\beta_1 = 0.9$

- we trained for 200 epochs and multiply the learning rate by 0.2 at the beginning of epochs 60, 120 and 160
- we used a weight decay of 0.0005
- we used a batch size of 128
- dropout was not used
- we normalized the images with ZCA whitening as in (Goodfellow et al., 2013)
- we used horizontal flip with probability 0.5 and random crops applied to the image after padded by 4 pixels

The only difference is that we used a larger initial learning rate $lr = 1$, rather than $lr = 0.1$ in (Zagoruyko & Komodakis, 2016).

We based our implementation on the code provided by the authors of (Zagoruyko & Komodakis, 2016) which can be found at `https://github.com/szagoruyko/wide-residual-networks`.

B NOTATIONS

For the ease of following the proofs we first summarize our notations.

B.1 PROBABILITY THEORY NOTATIONS

- $\Delta_n = \{p = (p_i)_{i=1}^n \in [0,1]^n, \ \sum_i p_i = 1\}$
- Let $q = (q_i) \in \Delta_n$, and f a function, we denote $\mathbb{E}_q[f(\mathrm{x})] := \sum_{i=1}^n q_i f(x_i)$.
- Let $q \in \Delta_n$, and f a function, we denote $\mathbb{V}_q[f(\mathrm{x})] := \sum_{i=1}^n q_i \left\| f(x_i) - \mathbb{E}_q[f(\mathrm{x})] \right\|^2$.
- $\hat{p}_{\mathrm{data}}$ is the uniform training data distribution, i.e. $\hat{p}_{\mathrm{data}} = \left(\frac{1}{n}\right)_{i=1}^n \in \Delta_n$

B.2 MACHINE LEARNING NOTATIONS

- n is the number of training examples
- d is the dimension of the output
- $\mathfrak{d}$ is the dimension of the input
- training data: $\{(x_i, y_i)\}_{i=1}^n$, where for all $i \in \{1, \ldots, n\}$, $x_i \in \mathbb{R}^{\mathfrak{d}}$ and $y_i \in \mathbb{R}^d$
- $h : \boldsymbol{x} \mapsto \boldsymbol{y}$ is the predictor
- $\boldsymbol{\theta}$ is the set of parameters of the predictor
- For all i, $h_i : \boldsymbol{\theta} \mapsto h(\mathrm{x}_i; \boldsymbol{\theta})$ is the output of the network for example i as a function of $\boldsymbol{\theta}$
- $\mathcal{L}$ is the objective function
- $\mathcal{L}_i : h_i \mapsto \mathcal{L}(h_i, y_i)$ is the objective function for example i.
- By abuse of notation we also denote by $\mathcal{L}$ the function $\mathcal{L} : (h_i)_{i=1}^n \mapsto (\mathcal{L}_i(h_i))_{i=1}^n$
- $b \in \{1, \ldots, n\}$ is the batch size
- $\eta > 0$ is the learning rate
- EMR is short for Empirical Risk Minimization

B.3 DISTRIBUTIONALLY ROBUST OPTIMISATION NOTATIONS

- DRO is short for Distributionally Robust Optimisation

B.4 MISCELLANEOUS

- By abuse of notation, and similarly to (Allen-Zhu et al., 2019), we use the Bachmann-Landau notations to hide constants that do not depend on our main hyper-parameters. Let f and g be two scalars, we note:

$$\begin{cases} f \leq O(g) & \iff & \exists c > 0 & \text{s.t.} & f \leq cg \\ f \geq \Omega(g) & \iff & \exists c > 0 & \text{s.t.} & f \geq cg \\ f = \Theta(g) & \iff & \exists c_1 > 0 \text{ and } \exists c_2 > c_1 & \text{s.t.} & c_1 g \leq f \leq c_2 g \end{cases}$$

C PROOFS

C.1 PROOF OF EXAMPLE 4.1: FORMULA OF THE SAMPLING PROBABILITIES FOR THE KL DIVERGENCE

We give here a simple proof of the formula of the sampling probabilities for the KL divergence as ϕ-divergence (i.e. $\phi : z \mapsto z \log(z) - z + 1$)

$$\forall \boldsymbol{\theta}, \quad \bar{p}(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))) = \operatorname{softmax}\left(\beta\, \mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))\right)$$

For any $\boldsymbol{\theta}$, the distributionally robust loss (9) for the KL divergence at $\boldsymbol{\theta}$ is given by

$$\begin{aligned} R \circ \mathcal{L} \circ \boldsymbol{h}(\boldsymbol{\theta}) &= \max_{q \in \Delta_n} \left(\sum_{i=1}^{n} q_i \underset{i}{\mathcal{L}} \circ h_i(\boldsymbol{\theta}) - \frac{1}{\beta} \sum_{i=1}^{n} q_i \log\left(n q_i\right) \right) \\ &= \max_{q \in \Delta_n} \sum_{i=1}^{n} \left(q_i \underset{i}{\mathcal{L}} \circ h_i(\boldsymbol{\theta}) - \frac{1}{\beta} q_i \log\left(n q_i\right) \right) \end{aligned}$$

To simplify the notations, let us denote $\boldsymbol{v} = (v_i)_{i=1}^n = \mathcal{L} \circ \boldsymbol{h}(\boldsymbol{\theta}) = (\mathcal{L}_i \circ h_i(\boldsymbol{\theta}))_{i=1}^n$, and $\bar{p} = (\bar{p}_i)_{i=1}^n = \bar{p}(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta})))$.

Thus $\bar{p}(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta})))$ is, by definition, solution of the optimization problem

$$\underset{q \in \Delta_n}{\arg\max} \sum_{i=1}^{n} \left(q_i v_i - \frac{1}{\beta} q_i \log\left(n q_i\right) \right) \tag{23}$$

First, let us remark that the function $q \mapsto \sum_{i=1}^{n} q_i \log\left(n q_i\right)$ is strictly convex on the non empty closed convex set Δ_n as a sum of strictly convex functions. This implies that the optimization (23) has a unique solution and as a result $\bar{p}(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta})))$ is well defined.

We know reformulate the optimization problem (23) as a convex smooth constrained optimization problem by writing the condition $q \in \Delta_n$ as constraints.

$$\begin{aligned} &\underset{q \in \mathbb{R}_+^n}{\arg\max} \sum_{i=1}^{n} \left(q_i v_i - \frac{1}{\beta} q_i \log\left(n q_i\right) \right) \\ &\text{s.t.} \sum_{i=1}^{n} q_i = 1 \end{aligned} \tag{24}$$

There exists a Lagrange multiplier $\lambda \in \mathbb{R}$, such that the solution $\bar{p}$ of (24) is characterized by

$$\begin{aligned} \forall i \in \{1, \ldots, n\}, \quad & v_i - \frac{1}{\beta}\left(\log\left(n \bar{p}_i\right) + 1\right) + \lambda = 0 \\ & \sum_{i=1}^{n} \bar{p}_i = 1 \end{aligned} \tag{25}$$

Which we can rewrite as

$$\begin{aligned} \forall i \in \{1, \ldots, n\}, \quad & \bar{p}_i = \frac{1}{n} \exp\left(\beta\left(v_i + \lambda\right) - 1\right) \\ & \frac{1}{n} \sum_{i=1}^{n} \exp\left(\beta\left(v_i + \lambda\right) - 1\right) = 1 \end{aligned} \tag{26}$$

The last equality gives

$$\exp(\beta\lambda - 1) = \frac{n}{\sum_{i=1}^{n} \exp(\beta v_i)}$$

And by replacing in the formula of the $\bar{p}_i$

$$\begin{aligned} \forall i \in \{1, \dots, n\}, \quad \bar{p}_i =& \frac{1}{n} \exp(\beta v_i) \exp(\beta\lambda - 1) \\ =& \frac{\exp(\beta v_i)}{\sum_{j=1}^{n} \exp(\beta v_j)} \end{aligned}$$

Which corresponds exactly to

$$\bar{p} = \mathrm{softmax}(\beta \boldsymbol{v})$$

C.2 Proof of lemma 4.1: regularity properties of R

For the ease of reading, let us first recall that given a ϕ-Divergence that satisfies assumptions 3.1, we have defined in (9)

$$\begin{aligned} R : \mathbb{R}^n &\to \mathbb{R} \\ v &\mapsto \max_{q \in \Delta_n} \sum_i q_i v_i - \frac{1}{\beta} D_\phi(q \| p_{train}) \end{aligned} \tag{27}$$

And in (13)

$$\begin{aligned} G : \mathbb{R}^n &\to \mathbb{R} \\ p &\mapsto \frac{1}{\beta} D_\phi(p \| p_{train}) + \delta_{\Delta_n}(p) \end{aligned} \tag{28}$$

where δ_{Δ_n} is the characteristic function of the closed convex set Δ_n, i.e.

$$\forall p \in \mathbb{R}^n, \; \delta_{\Delta_n}(p) = \begin{cases} 0 & \text{if } p \in \Delta_n \\ +\infty & \text{otherwise} \end{cases} \tag{29}$$

We now prove Lemma 4.1 on the regularity of R.

Lemma C.1 (Regularity of R – Restated from Lemma 4.1)**.** *Let ϕ that satisfies Assumption 3.1, G and R satisfy*

$$G \textit{ is } \left(\frac{n\rho}{\beta}\right) \textit{-strongly convex} \tag{30}$$

$$R(\mathcal{L}(h(\boldsymbol{\theta}))) = \max_{q \in \mathbb{R}^n} \left(\langle \mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta})), q \rangle - G(q) \right) = G^* \left(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta})) \right) \tag{31}$$

$$R \textit{ is } \left(\frac{\beta}{n\rho}\right) \textit{-gradient Lipschitz continuous.} \tag{32}$$

ϕ is ρ-strongly convex on $[0, n]$ so

$$\forall x, y \in [0, n]^2, \forall \lambda \in [0, 1], \phi(\lambda x + (1-\lambda) y) \leq \lambda \phi(x) + (1-\lambda)\phi(y) - \frac{\rho \lambda (1-\lambda)}{2} |y - x|^2 \tag{33}$$

Let $p = (p_i)_{i=1}^n$, $q = (q_i)_{i=1}^n \in \Delta_n$, and $\lambda \in [0, 1]$, using (33) and the convexity of δ_{Δ_n}, we obtain:

$$\begin{aligned} G(\lambda p + (1-\lambda) q) &= \frac{1}{\beta n} \sum_{i=1}^{n} \phi(n\lambda p_i + n(1-\lambda) q_i) + \delta_{\Delta_n}(\lambda p + (1-\lambda) q) \\ &\leq \lambda G(p) + (1-\lambda) G(q) - \frac{1}{\beta n} \sum_{i=1}^{n} \frac{\rho \lambda (1-\lambda)}{2} |n q_i - n p_i|^2 \\ &\leq \lambda G(p) + (1-\lambda) G(q) - \frac{n\rho}{\beta} \frac{\lambda(1-\lambda)}{2} \|q - p\|^2 \end{aligned} \tag{34}$$

This proves that G is $\frac{n\rho}{\beta}$-strongly convex.

Since G is convex, $R = G^*$ is also convex, and $R^* = (G^*)^* = G$ (Hiriart-Urruty & Lemaréchal, 2013).

We obtain (31) using Definition 4.1.

We now show that R is Frechet differentiable on $\mathbb{R}^n$. Let $v \in \mathbb{R}^n$.

G is strongly-convex, so in particular G is strictly convex. This implies that the following optimization problem has a unique solution that we denote $\hat{p}(v)$.

$$\arg\max_{q \in \mathbb{R}^n} \left(\langle v, q \rangle - G(q)\right) \tag{35}$$

In addition

$$\begin{aligned}
\hat{p} \in \Delta_n \text{ solution of (35)} &\iff 0 \in v - \partial G(\hat{p}) \\
&\iff v \in \partial G(\hat{p}) \\
&\iff \hat{p} \in \partial G^*(v) \\
&\iff \hat{p} \in \partial R(v)
\end{aligned}$$

where we have used (Hiriart-Urruty & Lemaréchal, 2013, Proposition 6.1.2 p.39) for the third equivalence, and (31) for the last equivalence.

As a result, $\partial R(v) = \{\hat{p}(v)\}$. this implies that R admit a gradient at v, and

$$\nabla_v R(v) = \hat{p}(v) \tag{36}$$

Since this holds for any $v \in \mathbb{R}^n$, we deduce that R is Frechet differentiable on $\mathbb{R}^n$.

We are now ready to show that R is $\frac{\beta}{n\rho}$-gradient Lipchitz continuous by using the following lemma (Hiriart-Urruty & Lemaréchal, 2013, Theorem 6.1.2 p.280).

Lemma C.2. *A necessary and sufficient condition for a convex function $f : \mathbb{R}^n \to \mathbb{R}$ to be c-strongly convex on a convex set C is that for all $x_1, x_2 \in C$*

$$\langle s_2 - s_1, x_2 - x_1 \rangle \geq c \left\| x_2 - x_1 \right\|^2 \quad \textit{for all } s_i \in \partial f(x_i), i = 1, 2.$$

Using this lemma for $f = G$, $c = \frac{n\rho}{\beta}$, and $C = \Delta_n$, we obtain:

For all $p_1, p_2 \in \Delta_n$, for all $v_1 \in \partial G(p_1)$, $v_2 \in \partial G(p_2)$,

$$\langle v_2 - v_1, p_2 - p_1 \rangle \geq \frac{n\rho}{\beta} \left\| p_2 - p_1 \right\|^2$$

In addition, for $i \in \{1,\, 2\}$, $v_i \in \partial G(p_i) \iff p_i \in \partial R(v_1) = \{\nabla_v R(v_i)\}$.

And using Cauchy Schwarz inequality

$$\left\| v_2 - v_1 \right\| \left\| p_2 - p_1 \right\| \geq \langle v_2 - v_1, p_2 - p_1 \rangle$$

We conclude that

$$\frac{n\rho}{\beta} \left\| \nabla_v R(v_2) - \nabla_v R(v_1) \right\| \leq \left\| v_2 - v_1 \right\|$$

Which implies that R is $\frac{\beta}{n\rho}$-gradient Lipchitz continuous.

C.3 Proof of Lemma 4.2: Formula of the distributionally robust loss gradient

We prove Lemma 4.2 that we restate here for the ease of reading.

Lemma C.3 (Stochastic Gradient of the Distributionally Robust Loss – Restated from Lemma 4.2)**.** *For all $\boldsymbol{\theta}$, we have*

$$\bar{p}(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))) = \nabla_{\boldsymbol{v}} R(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))) \tag{37}$$

$$\nabla_{\boldsymbol{\theta}} (R \circ \mathcal{L} \circ \boldsymbol{h})(\boldsymbol{\theta}) = \mathbb{E}_{\bar{p}(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta})))} \left[\nabla_{\boldsymbol{\theta}} \mathcal{L} \left(h(\mathrm{x}; \boldsymbol{\theta}), y \right) \right] \tag{38}$$

where $\nabla_{\boldsymbol{v}} R$ is the gradient of R with respect to its input.

For a given $\boldsymbol{\theta}$, equality (37) is a special case of (36) for $\boldsymbol{v} = \mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta}))$.

Then using the chain rule and (37),

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}}(R \circ \mathcal{L} \circ \boldsymbol{h})(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \frac{\partial R}{\partial v_i}(\mathcal{L} \circ \boldsymbol{h}(\boldsymbol{\theta}))\nabla_{\boldsymbol{\theta}}(\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}) \\
&= \sum_{i=1}^{n} \bar{p}_i(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta})))\nabla_{\boldsymbol{\theta}}(\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}) \\
&= \mathbb{E}_{\bar{p}(\mathcal{L}(\boldsymbol{h}(\boldsymbol{\theta})))}\left[\nabla_{\boldsymbol{\theta}}\, \mathcal{L}\left(h(\mathrm{x}; \boldsymbol{\theta}), y\right)\right]
\end{aligned}$$

C.4 Proof of Theorem 4.1: Distributionally Robust Optimization as Principled Hard Example Mining

Let D_ϕ an ϕ-divergence satisfying Assumption 3.1, and $v = (v_i)_{i=1}^n \in \mathbb{R}^n$. v will play the role of a generic loss vector.

ϕ is strongly convex, and Δ^n is closed and convex, so the following optimization problem has one and only one solution:

$$\max_{p=(p_i)_{i=1}^n \in \Delta^n} \langle v, p\rangle - \frac{1}{\beta n}\sum_{i=1}^{n} \phi(np_i) \tag{39}$$

Making the constraints associated with $p \in \Delta_n$ explicit, this can be rewritten as

$$\begin{aligned}
&\max_{p=(p_i)_{i=1}^n \in \mathbb{R}^n} \langle v, p\rangle - \frac{1}{\beta n}\sum_{i=1}^{n} \phi(np_i) \\
&\text{s.t. } \forall i \in \{1, \dots, n\},\ p_i \geq 0 \\
&\text{s.t. } \sum_{i=1}^{n} p_i = 1
\end{aligned} \tag{40}$$

There exists KKT multipliers $\lambda \in \mathbb{R}$ and $\forall i,\ \mu_i \geq 0$ such that the solution $\bar{p} = (\bar{p}_i)_{i=1}^n$ satisfies:

$$\begin{cases}
\forall i \in \{1, \dots, n\}, & v_i - \frac{1}{\beta}\phi'(n\bar{p}_i) + \lambda - \mu_i = 0 \\
\forall i \in \{1, \dots, n\}, & \mu_i p_i = 0 \\
\forall i \in \{1, \dots, n\}, & p_i \geq 0 \\
& \sum_{i=1}^{n} \bar{p}_i = 1
\end{cases} \tag{41}$$

Since ϕ is continuously differentiable and strongly convex, we have $(\phi')^{-1} = (\phi^*)'$, where ϕ^* is the Fenchel conjugate of ϕ (see Hiriart-Urruty & Lemaréchal, 2013, Proposition 6.1.2). As a result, (41) can be rewritten has:

$$\begin{cases}
\forall i \in \{1, \dots, n\}, & \bar{p}_i = \frac{1}{n}\left(\phi^*\right)'\left(\beta(v_i + \lambda - \mu_i)\right) \\
\forall i \in \{1, \dots, n\}, & \mu_i p_i = 0 \\
\forall i \in \{1, \dots, n\}, & p_i \geq 0 \\
& \frac{1}{n}\sum_{i=1}^{n} \left(\phi^*\right)'\left(\beta(v_i + \lambda - \mu_i)\right) = 1
\end{cases} \tag{42}$$

We now show that the KKT multipliers are uniquely defined.

The μ_i's are uniquely defined by v and λ:
Since $\forall i \in \{1, \ldots, n\},\ \mu_i p_i = 0,\ p_i \geq 0$ and $\mu_i \geq 0$, for all $\forall i \in \{1, \ldots, n\}$, either $p_i = 0$ or $\mu_i = 0$.

In the case $p_i = 0$, using (42) it comes $(\phi^*)'\left(\beta(v_i + \lambda - \mu_i)\right) = 0$.

According to assumption 3.1, ϕ is strongly convex and continuously differentiable, so ϕ' and $(\phi^*)' = (\phi')^{-1}$ are continuous and strictly increasing functions. As a result, it exists a unique μ_i (dependent to v and λ) such that:

$$(\phi^*)'\left(\beta(v_i + \lambda - \mu_i)\right) = 0$$

And (42) can be rewritten as:

$$\begin{cases} \forall i \in \{1, \ldots, n\}, \quad \bar{p}_i = \mathrm{ReLU}\left(\dfrac{1}{n}(\phi^*)'\left(\beta(v_i + \lambda)\right)\right) = \dfrac{1}{n}\mathrm{ReLU}\left((\phi^*)'\left(\beta(v_i + \lambda)\right)\right) \\ \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \mathrm{ReLU}\left((\phi^*)'\left(\beta(v_i + \lambda)\right)\right) = 1 \end{cases} \tag{43}$$

λ is uniquely defined by v and a continuous function of v:
Let $\lambda \in \mathbb{R}$ that satisfies (43).

We have $\frac{1}{n}\sum_{i=1}^{n} \mathrm{ReLU}\left((\phi^*)'\left(\beta(v_i + \lambda)\right)\right) = 1$. So there exists at least one index i_0 such that

$$\mathrm{ReLU}\left((\phi^*)'\left(\beta(v_{i_0} + \lambda)\right)\right) = (\phi^*)'\left(\beta(v_{i_0} + \lambda)\right) \geq 1$$

Since $(\phi^*)^{-1}$ is continuous and striclty increasing, $\lambda' \mapsto \mathrm{ReLU}\left((\phi^*)'\left(\beta(v_{i_0} + \lambda')\right)\right)$ is continuous and strictly increasing on a neighborhood of λ.

In addition ReLU is continuous and increasing, so for all $i \in \{1, \ldots, n\}$, $\lambda' \mapsto \mathrm{ReLU}\left((\phi^*)'\left(\beta(v_i + \lambda')\right)\right)$ is a continuous and increasing function.

As a result, $\lambda' \mapsto \frac{1}{n}\sum_{i=1}^{n} \mathrm{ReLU}\left((\phi^*)'\left(\beta(v_i + \lambda')\right)\right)$ is a continous function that is increasing on $\mathbb{R}$, and strictly increasing on a neighborhood of λ.

This implies that λ is uniquely defined by v, and that $v \mapsto \lambda(v)$ is continuous.

Hard Example Mining Sampling:

For any pseudo loss vector $v = (v_i)_{i=1}^{n} \in \mathbb{R}^n$, there exists a unique λ and a unique $\bar{p}$ that satisfies (43), so we can define the mapping:

$$\begin{aligned} \bar{p} :\ & \mathbb{R}^n \to \Delta_n \\ & v \mapsto \bar{p}(v; \lambda(v)) \end{aligned} \tag{44}$$

where for all v, $\lambda(v)$ is the unique $\lambda \in \mathbb{R}$ satisfying (43).

We will now demonstrate that each $\bar{p}_{i_0}(v)$ for $i_0 \in \{1, \ldots, n\}$ is an increasing function of v_i and a decreasing function of the v_i for $i \neq i_0$. Without loss of generality we assume $i_0 = 1$.

Let $v = (v_i)_{i=1}^{n} \in \mathbb{R}^n$, and $\epsilon > 0$.

Let us define $v' = (v_i')_{i=1}^{n} \in \mathbb{R}^n$, such that $v_1' = v_1 + \epsilon$ and $\forall i \in \{2, \ldots, n\},\ v_i' = v_i$.

Similarly as in the proof of the uniqueness of λ above, we can show that there exists $\eta > 0$ such that the function

$$F : \lambda' \mapsto \frac{1}{n}\sum_{i=1}^{n} \mathrm{ReLU}\left((\phi^*)'\left(\beta(v_i + \lambda')\right)\right)$$

is continuous and strictly increasing on $[\lambda(v) - \eta, \lambda(v) + \eta]$, and $F(\lambda(v)) = 1$.

$v \mapsto \lambda(v)$ is continuous, so for ϵ small enough $\lambda(v') \in [\lambda(v) - \eta, \lambda(v) + \eta]$.

Let us now prove by contradiction that $\lambda(v') \leq \lambda(v)$. Therefore, let us assume that $\lambda(v') > \lambda(v)$. Then, as $\mathrm{ReLU} \circ (\phi^*)'$ is an increasing function and F is strictly increasing on $[\lambda(v) - \eta, \lambda(v) + \eta]$, and $\epsilon > 0$ we obtain

$$\begin{aligned} 1 &= \frac{1}{n}\sum_{i=1}^{n} \mathrm{ReLU}\left((\phi^*)'\left(\beta(v_i' + \lambda(v'))\right)\right) \\ &\geq \frac{1}{n}\sum_{i=1}^{n} \mathrm{ReLU}\left((\phi^*)'\left(\beta(v_i + \lambda(v'))\right)\right) \\ &\geq F(\lambda(v')) \\ &> F(\lambda(v)) \\ &> 1 \end{aligned}$$

which is a contradiction. As a result

$$\lambda(v') \leq \lambda(v) \tag{45}$$

Using (45), (43), and the fact that $\mathrm{ReLU} \circ (\phi^*)'$ is an increasing function, we obtain for all $i \in \{2, \ldots, n\}$

$$\begin{aligned} \bar{p}_i(v') &= \frac{1}{n}\mathrm{ReLU}\left((\phi^*)'\left(\beta(v_i' + \lambda(v'))\right)\right) \\ &= \frac{1}{n}\mathrm{ReLU}\left((\phi^*)'\left(\beta(v_i + \lambda(v'))\right)\right) \\ &\leq \frac{1}{n}\mathrm{ReLU}\left((\phi^*)'\left(\beta(v_i + \lambda(v))\right)\right) \\ &\leq \bar{p}_i(v) \end{aligned} \tag{46}$$

In addition

$$\sum_{i=1}^{n} \bar{p}_i(v') = 1 = \sum_{i=1}^{n} \bar{p}_i(v)$$

So necessarily

$$\bar{p}_1(v') \geq \bar{p}_1(v) \tag{47}$$

This holds for any i_0 and any v, which concludes the proof.

C.5 Proof that $R \circ \mathcal{L}$ is one-sided gradient Lipchitz

This property that $R \circ \mathcal{L}$ is one-sided gradient Lipschitz is a key element for the proof of the semi-smoothness theorem for the distributionally robust loss Theorem C.1.

Under assumption 3.1, we have shown that R^* is $\frac{\beta}{n\rho}$-gradient Lipchitz continuous. And under assumption 5.2, for all i, $\mathcal{L}_i$ is $C(\mathcal{L})$-Lipschitz continuous and $C(\nabla \mathcal{L})$-gradient Lipschitz continuous.

Let $z = (z_i)_{i=1}^n, z' = (z_i')_{i=1}^n \in \mathbb{R}^{dn}$.

We want to show that $R \circ \mathcal{L}$ is one-sided gradient Lipschitz, i.e. we want to prove the existence of a constant $C > 0$, independent to z and z', such that:

$$\langle \nabla_z (R \circ \mathcal{L})(z) - \nabla_z (R \circ \mathcal{L})(z'), z - z' \rangle \leq C \left\| z - z' \right\|^2$$

We have

$$\begin{aligned} &\langle \nabla_z (R \circ \mathcal{L})(z) - \nabla_z (R \circ \mathcal{L})(z'), z - z' \rangle \\ &= \sum_{i=1}^{n} \langle \nabla_{z_i} (R \circ \mathcal{L})(z) - \nabla_{z_i} (R \circ \mathcal{L})(z'), z_i - z_i' \rangle \\ &= \sum_{i=1}^{n} \langle \bar{p}_i(\mathcal{L}(z)) \nabla_{z_i} \underset{i}{\mathcal{L}}(z_i) - \bar{p}_i(\mathcal{L}(z')) \nabla_{z_i} \underset{i}{\mathcal{L}}(z_i'), z_i - z_i' \rangle \\ &= \sum_{i=1}^{n} \bar{p}_i(\mathcal{L}(z)) \langle \nabla_{z_i} \underset{i}{\mathcal{L}}(z_i) - \nabla_{z_i} \underset{i}{\mathcal{L}}(z_i'), z_i - z_i' \rangle \\ &\quad + \sum_{i=1}^{n} \left(\bar{p}_i(\mathcal{L}(z)) - \bar{p}_i(\mathcal{L}(z'))\right) \langle \nabla_{z_i} \underset{i}{\mathcal{L}}(z_i'), z_i - z_i' \rangle \end{aligned} \tag{48}$$

Where for all $i \in \{1, \ldots, n\}$ we have used the chain rule

$$\nabla_{z_i}(R \circ \mathcal{L})(z) = \sum_{j=1}^{n} \frac{\partial R^*}{\partial v_j}(\mathcal{L}(z)) \nabla_{z_i} \underset{j}{\mathcal{L}}(z_j) = \bar{p}_i(\mathcal{L}(z)) \nabla_{z_i} \underset{i}{\mathcal{L}}(z_i)$$

Let

$$A = \left| \sum_{i=1}^{n} \bar{p}_i(\mathcal{L}(z)) \langle \nabla_{z_i} \underset{i}{\mathcal{L}}(z_i) - \nabla_{z_i} \underset{i}{\mathcal{L}}(z_i'), z_i - z_i' \rangle \right|$$

For all i, $\mathcal{L}_i$ is $C(\nabla \mathcal{L})$-gradient Lipchitz continuous, so using Cauchy-Schwarz inequality

$$A \leq \sum_{i=1}^{n} C(\nabla \mathcal{L}) \left\| z_i - z_i' \right\|^2 = C(\nabla \mathcal{L}) \left\| z - z' \right\|^2 \tag{49}$$

Let

$$B = \left| \sum_{i=1}^{n} \left(\bar{p}_i(\mathcal{L}(z)) - \bar{p}_i(\mathcal{L}(z')) \right) \langle \nabla_{z_i} \underset{i}{\mathcal{L}}(z_i'), z_i - z_i' \rangle \right|$$

Using the triangular inequality:

$$\begin{aligned} B \leq & \left| \sum_{i=1}^{n} \left(\bar{p}_i(\mathcal{L}(z)) - \bar{p}_i(\mathcal{L}(z')) \right) \left(\underset{i}{\mathcal{L}}(z_i) - \underset{i}{\mathcal{L}}(z_i') \right) \right| \\ & + \left| \sum_{i=1}^{n} \left(\bar{p}_i(\mathcal{L}(z)) - \bar{p}_i(\mathcal{L}(z')) \right) \left(\underset{i}{\mathcal{L}}(z_i') + \langle \nabla_{z_i} \underset{i}{\mathcal{L}}(z_i'), z_i - z_i' \rangle - \underset{i}{\mathcal{L}}(z_i) \right) \right| \\ \leq & \langle \nabla(R^*)(\mathcal{L}(z)) - \nabla(R^*)(\mathcal{L}(z')), \mathcal{L}(z) - \mathcal{L}(z') \rangle \\ & + 2 \sum_{i=1}^{n} \left| \underset{i}{\mathcal{L}}(z_i') + \langle \nabla_{z_i} \underset{i}{\mathcal{L}}(z_i'), z_i - z_i' \rangle - \underset{i}{\mathcal{L}}(z_i) \right| \\ \leq & \frac{\beta}{n\rho} \left\| \mathcal{L}(z) - \mathcal{L}(z') \right\|^2 + 2 \frac{C(\nabla \mathcal{L})}{2} \left\| z - z' \right\|^2 \\ \leq & \left(\frac{\beta C(\mathcal{L})^2}{n\rho} + C(\nabla \mathcal{L}) \right) \left\| z - z' \right\|^2 \end{aligned} \tag{50}$$

Combining (48), (49) and (50) we finally obtain:

$$\langle \nabla_z (R \circ \mathcal{L})(z) - \nabla_z (R \circ \mathcal{L})(z'), z - z' \rangle \leq \left(\frac{\beta C(\mathcal{L})^2}{n\rho} + 2C(\nabla \mathcal{L}) \right) \left\| z - z' \right\|^2 \tag{51}$$

From there, we can obtain the following inequality that will be used for the proof of the semi-smoothness property in Theorem C.1:

$$\begin{aligned} & R(\mathcal{L}(z')) - R(\mathcal{L}(z)) - \langle \nabla_z (R \circ \mathcal{L})(z), z' - z \rangle \\ & = \int_{t=0}^{1} \langle \nabla_z (R \circ \mathcal{L}) \left(z + t(z' - z) \right) - \nabla_z (R \circ \mathcal{L})(z), z' - z \rangle dt \\ & \leq \frac{1}{2} \left(\frac{\beta C(\mathcal{L})^2}{n\rho} + 2C(\nabla \mathcal{L}) \right) \left\| z - z' \right\|^2 \end{aligned} \tag{52}$$

C.6 Proof of the convergence of Robust SGD

In this part, we prove the results of Therem 5.1 and 5.2.

They are generalizations of the convergence result for SGD presented in Theorem 2 of (Allen-Zhu et al., 2019).

For the ease of reading the proof, we remind here the chain rules for the distributionally robust loss (9) that we are going to use intensively in the following proofs.

Chain rule for the derivative of $R \circ \mathcal{L}$ with respect to the network outputs h:

$$\begin{aligned} \nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta})) &= \left(\nabla_{h_i} (R \circ \mathcal{L})(h(\boldsymbol{\theta}))\right)_{i=1}^n \\ \forall i \in \{1, \dots n\}, \quad \nabla_{h_i} (R \circ \mathcal{L})(h(\boldsymbol{\theta})) &= \sum_{j=1}^n \frac{\partial R}{\partial v_j}(\mathcal{L}(h(\boldsymbol{\theta}))) \nabla_{h_i} \underset{j}{\mathcal{L}}(h_j(\boldsymbol{\theta})) \\ &= \bar{p}_i(\mathcal{L}(h(\boldsymbol{\theta}))) \nabla_{h_i} \underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta})) \end{aligned} \tag{53}$$

Chain rule for the derivative of $R \circ \mathcal{L} \circ h$ with respect to the network parameters $\boldsymbol{\theta}$:

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} (R \circ \mathcal{L} \circ h)(\boldsymbol{\theta}) &= \sum_{i=1}^n \nabla_\theta h_i(\boldsymbol{\theta}) \nabla_{h_i} (R \circ \mathcal{L})(h(\boldsymbol{\theta})) \\ &= \sum_{i=1}^n \bar{p}_i(\mathcal{L}(h(\boldsymbol{\theta}))) \nabla_\theta h_i(\boldsymbol{\theta}) \nabla_{h_i} \underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta})) \\ &= \sum_{i=1}^n \bar{p}_i(\mathcal{L}(h(\boldsymbol{\theta})) \nabla_{\boldsymbol{\theta}} (\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta})) \end{aligned} \tag{54}$$

where for all $i \in \{1, \dots n\}$, $\nabla_\theta h_i(\boldsymbol{\theta})$ is the transpose of the Jacobian matrix of h_i as a function of $\boldsymbol{\theta}$.

C.6.1 Semi-smoothness property for the Distributionally Robust Loss

We prove the following lemma which is a generalization of Theorem 4 in (Allen-Zhu et al., 2019) for the distributionally robust loss (9).

Theorem C.1 (Semi-smoothness of the Distributionally Robust Loss)**.**
Let $\omega \in \left[\Omega\left(\frac{d^{3/2}}{m^{3/2} L^{3/2} \log^{3/2}(m)}\right), O\left(\frac{1}{L^{4.5} \log^3(m)}\right)\right]$, and the $\boldsymbol{\theta}^{(0)}$ being initialized randomly as described in assumption 5.1. With probability as least $1 - \exp\left(-\Omega(m\omega^{3/2} L)\right)$ over the initialization, we have for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in (\mathbb{R}^{m \times m})^L$ with $\left\|\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)}\right\|_2 \leq \omega$, and $\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2 \leq \omega$

$$\begin{aligned} R(\mathcal{L}(h(\boldsymbol{\theta}')) &\leq R(\mathcal{L}(h(\boldsymbol{\theta})) + \langle \nabla_{\boldsymbol{\theta}} (R \circ \mathcal{L} \circ h)(\boldsymbol{\theta}), \boldsymbol{\theta}' - \boldsymbol{\theta} \rangle \\ &\quad + \|\nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta}))\|_{2,1} O\left(\frac{L^2 \omega^{1/3} \sqrt{m \log(m)}}{\sqrt{d}}\right) \|\boldsymbol{\theta}' - \boldsymbol{\theta}\|_{2,\infty} \\ &\quad + O\left(\left(\frac{\beta C(\mathcal{L})^2}{n\rho} + 2C(\nabla \mathcal{L})\right) \frac{nL^2 m}{d}\right) \|\boldsymbol{\theta}' - \boldsymbol{\theta}\|_{2,\infty}^2 \end{aligned} \tag{55}$$

where for all layer $l \in \{1, \dots, L\}$, $\boldsymbol{\theta}_l$ is the vector of parameters for layer l, and

$$\begin{aligned} \|\boldsymbol{\theta}' - \boldsymbol{\theta}\|_{2,\infty} &= \max_l \|\boldsymbol{\theta}_l' - \boldsymbol{\theta}_l\|_2 \\ \|\boldsymbol{\theta}' - \boldsymbol{\theta}\|_{2,\infty}^2 &= \left(\max_l \|\boldsymbol{\theta}_l' - \boldsymbol{\theta}_l\|_2^2\right)^2 = \max_l \|\boldsymbol{\theta}_l' - \boldsymbol{\theta}_l\|_2^2 \\ \|\nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta}))\|_{2,1} &= \sum_{i=1}^n \|\nabla_{h_i} (R \circ \mathcal{L})(h(\boldsymbol{\theta}))\|_2 \\ &= \sum_{i=1}^n \left\|\bar{p}_i(\mathcal{L}(h(\boldsymbol{\theta}))) \nabla_{h_i} \underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}))\right\|_2 \quad \text{(chain rule (53))} \end{aligned}$$

To compare this semi-smoothness result to the one in (Allen-Zhu et al., 2019, Theorem 4), let us first remark that

$$\|\nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta}))\|_{2,1} \leq \sqrt{n} \|\nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta}))\|_{2,2}$$

As a result, our result is analogous to (Allen-Zhu et al., 2019, Theorem 4), up to an additional multiplicative factor $\left(\frac{\beta C(\mathcal{L})^2}{n\rho} + 2C(\nabla \mathcal{L})\right)$ in the last term of the right-hand side. It is worth noting that there is also implicitly an additional multiplicative factor $C(\nabla \mathcal{L})$ in Theorem 3 of (Allen-Zhu et al., 2019) since (Allen-Zhu et al., 2019) make the assumption that $C(\nabla \mathcal{L}) = 1$ (see Allen-Zhu et al., 2019, Appendix A).

Let $\boldsymbol{\theta}, \boldsymbol{\theta}' \in (\mathbb{R}^{m\times m})^L$ verifying the conditions of Theorem C.1.

Let $A = R(\mathcal{L}(h(\boldsymbol{\theta}')) - R(\mathcal{L}(h(\boldsymbol{\theta})) - \langle \nabla_{\boldsymbol{\theta}}(R \circ \mathcal{L} \circ h)(\boldsymbol{\theta}), \boldsymbol{\theta}' - \boldsymbol{\theta}\rangle$, the quantity we want to bound.

Using (52) for $z = h(\boldsymbol{\theta})$ and $z' = h(\boldsymbol{\theta}')$, we obtain

$$\begin{aligned} A \leq & \frac{1}{2}\left(\frac{\beta C(\mathcal{L})^2}{n\rho} + 2C(\nabla \mathcal{L})\right)\|h(\boldsymbol{\theta}') - h(\boldsymbol{\theta})\|_2^2 \\ & + \langle \nabla_h (R\circ\mathcal{L})(h(\boldsymbol{\theta})), h(\boldsymbol{\theta}') - h(\boldsymbol{\theta})\rangle \\ & - \langle \nabla_{\boldsymbol{\theta}}(R \circ \mathcal{L} \circ h)(\boldsymbol{\theta}), \boldsymbol{\theta}' - \boldsymbol{\theta}\rangle \end{aligned} \tag{56}$$

Then using the chain rule (54)

$$\begin{aligned} A \leq & \frac{1}{2}\left(\frac{\beta C(\mathcal{L})^2}{n\rho} + 2C(\nabla \mathcal{L})\right)\|h(\boldsymbol{\theta}') - h(\boldsymbol{\theta})\|_2^2 \\ & + \sum_{i=1}^{n}\langle \nabla_{h_i}(R\circ\mathcal{L})(h(\boldsymbol{\theta})), h_i(\boldsymbol{\theta}') - h_i(\boldsymbol{\theta}) - \left(\nabla_{\boldsymbol{\theta}} h_i(\boldsymbol{\theta})\right)^T(\boldsymbol{\theta}' - \boldsymbol{\theta})\rangle \end{aligned} \tag{57}$$

For all $i \in \{1,\ldots,n\}$, let us denote $\breve{loss}_i := \nabla_{h_i}(R\circ\mathcal{L})(h(\boldsymbol{\theta}))$ to match the notations used in (Allen-Zhu et al., 2019) for the derivative of the loss with respect to the output of the network for example i of the training set.

With this notation, we obtain exactly equation (11.3) in (Allen-Zhu et al., 2019) up to the multiplicative factor $\left(\frac{\beta C(\mathcal{L})^2}{n\rho} + 2C(\nabla \mathcal{L})\right)$ for the distributionally robust loss.

From there the proof of Theorem 4 in (Allen-Zhu et al., 2019) being independent to the formula for $\breve{loss}_i$, we can conclude the proof of our Theorem C.1 (as in Allen-Zhu et al., 2019, Appendix A).

C.6.2 Gradient Bounds for the Distributionally Robust Loss

We prove the following lemma which is a generalization of Theorem 3 in (Allen-Zhu et al., 2019) for the distributionally robust loss (9).

Theorem C.2 (Gradient Bounds for the Distributionally Robust Loss)**.**
Let $\omega \in O\left(\frac{\delta^{3/2}}{n^{9/2}L^6\log^3(m)}\right)$, and $\boldsymbol{\theta}^{(0)}$ being initialized randomly as described in assumption 5.1. With probability as least $1 - \exp\left(-\Omega(m\omega^{3/2}L)\right)$ over the initialization, we have for all $\boldsymbol{\theta} \in (\mathbb{R}^{m\times m})^L$ with $\left\|\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)}\right\|_2 \leq \omega$

$$\begin{aligned} & \forall i \in \{1,\ldots,n\},\ \forall l \in \{1,\ldots,L\},\ \forall \hat{\mathcal{L}} \in \mathbb{R}^n \\ & \left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}_l}(\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta})\right\|_2^2 \leq O\left(\frac{m}{d}\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{h_i}\underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}))\right\|_2^2\right) \\ & \forall l \in \{1,\ldots,L\},\ \forall \hat{\mathcal{L}} \in \mathbb{R}^n \\ & \left\|\sum_{i=1}^{n}\bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}_l}(\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta})\right\|_2^2 \leq O\left(\frac{mn}{d}\sum_{i=1}^{n}\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{h_i}\underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}))\right\|_2^2\right) \\ & \left\|\sum_{i=1}^{n}\bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}_L}(\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta})\right\|_2^2 \geq \Omega\left(\frac{m\delta}{dn^2}\sum_{i=1}^{n}\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{h_i}\underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}))\right\|_2^2\right) \end{aligned} \tag{58}$$

It is worth noting that the loss vector $\hat{\mathcal{L}}$ used for computing the robust probabilities $\bar{p}(\hat{\mathcal{L}}) = \left(\bar{p}_i(\hat{\mathcal{L}})\right)_{i=1}^n$ does not have to be equal to $\mathcal{L}(h(\boldsymbol{\theta}))$.

We will use this for the proof of the Robust SGD with stale loss history.

The adaptation of the proof of Theorem 3 in (Allen-Zhu et al., 2019) is straightforward.

Let $\boldsymbol{\theta} \in \left(\mathbb{R}^{m\times m}\right)^L$ satisfying the conditions of Theorem C.2, and $\hat{\mathcal{L}} \in \mathbb{R}^n$.

Let us denote $v := \left(\bar{p}_i(\hat{\mathcal{L}})\nabla_{h_i} \mathcal{L}_i(h_i(\boldsymbol{\theta}))\right)_{i=1}^n$, applying the proof of Theorem 3 in (Allen-Zhu et al., 2019) to our v gives:

$$
\begin{aligned}
&\forall i \in \{1,\dots,n\},\ \forall l \in \{1,\dots,L\},\\
&\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}_l}(\underset{i}{\mathcal{L}}\circ h_i)(\boldsymbol{\theta})\right\|_2^2 \le O\left(\frac{m}{d}\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{h_i}\underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}))\right\|_2^2\right)\\
&\forall l \in \{1,\dots,L\},\ \forall \hat{\mathcal{L}} \in \mathbb{R}^n\\
&\left\|\sum_{i=1}^n \bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}_l}(\underset{i}{\mathcal{L}}\circ h_i)(\boldsymbol{\theta})\right\|_2^2 \le O\left(\frac{mn}{d}\sum_{i=1}^n\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{h_i}\underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}))\right\|_2^2\right)\\
&\left\|\sum_{i=1}^n \bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}_L}(\underset{i}{\mathcal{L}}\circ h_i)(\boldsymbol{\theta})\right\|_2^2 \ge \Omega\left(\frac{m\delta}{dn}\max_i\left(\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{h_i}\underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}))\right\|_2^2\right)\right)
\end{aligned}
$$

In addition

$$\max_i\left(\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{h_i}\underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}))\right\|_2^2\right) \ge \frac{1}{n}\sum_{i=1}^n\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{h_i}\underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}))\right\|_2^2$$

This allows to conclude the proof of our Theorem C.2.

C.6.3 Convergence of Robust SGD with Exact Loss History

We can now prove Theorem 5.1.

Theorem C.3 (Convergence of Robust SGD with exact Loss History – Restated from Theorem 5.1)**.** *Suppose batch size $1 \le b \le n$, number of hidden units $m \ge \Omega(d\epsilon^{-1} \times \text{poly}(n, L, \delta^{-1}))$, and $\delta \ge O\left(\frac{1}{L}\right)$. Let $\epsilon > 0$, and the learning rate be $\eta_{exact} = \Theta\left(\frac{\alpha n^2\rho}{\beta C(\mathcal{L})^2 + 2n\rho C(\nabla\,\mathcal{L})} \times \frac{b\delta d}{\text{poly}(n,L)m\log^2(m)}\right)$, with probability at least $1 - \exp\left(-\Omega(\log^2(m))\right)$ over the randomness of the initialization and the mini-batches, Robust SGD with exact loss vector finds $\|\nabla_{\boldsymbol{\theta}}(R\circ f\circ h)(\boldsymbol{\theta})\| \le \epsilon$ after $T = O\left(\frac{Ln^3}{\eta\delta\epsilon^2}\right)$ iterations.*

Similarly to the proof of the convergence of SGD for the mean loss (4) (Theorem 2 in (Allen-Zhu et al., 2019)), the convergence of SGD for the distributionally robust loss (9) will mainly rely on the semi-smoothness property (Theorem C.1) and the gradient bound (Theorem C.2) that we have proved previously for the distributionally robust loss.

Let $\boldsymbol{\theta} \in \left(\mathbb{R}^{m\times m}\right)^L$ satisfying the conditions of Theorem 5.1, and $\hat{\mathcal{L}}$ be the exact loss history at $\boldsymbol{\theta}$, i.e.

$$\hat{\mathcal{L}} = \left(\underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}))\right)_{i=1}^n \tag{59}$$

For the batch size $b \in \{1,\dots,n\}$, let $S = \{i_j\}_{j=1}^b$ a batch of indices drawn from $\bar{p}(\hat{\mathcal{L}})$ without replacement, i.e.

$$\forall j \in \{1,\dots b\},\ i_j \overset{\text{i.i.d.}}{\sim} \bar{p}(\hat{\mathcal{L}}) \tag{60}$$

Let $\boldsymbol{\theta}' \in \left(\mathbb{R}^{m\times m}\right)^L$ be the values of the parameters after a stochastic gradient descent step at $\boldsymbol{\theta}$ for the batch S, i.e.

$$\boldsymbol{\theta}' = \boldsymbol{\theta} - \eta\frac{1}{b}\sum_{i\in S}\nabla_{\boldsymbol{\theta}}(\underset{i}{\mathcal{L}}\circ h_i)(\boldsymbol{\theta}) \tag{61}$$

where $\eta > 0$ is the learning rate.

Assuming that $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ satisfies the conditions of Theorem C.1, we obtain

$$\begin{aligned}R(\mathcal{L}(h(\boldsymbol{\theta}')) \leq & R(\mathcal{L}(h(\boldsymbol{\theta})) - \eta\langle\nabla_{\boldsymbol{\theta}}(R \circ \mathcal{L} \circ h)(\boldsymbol{\theta}), \frac{1}{b}\sum_{i \in S}\nabla_{\boldsymbol{\theta}}(\mathcal{L}_i \circ h_i)(\boldsymbol{\theta})\rangle \\ & + \eta\sqrt{n}\left\|\nabla_h(R \circ \mathcal{L})(h(\boldsymbol{\theta}))\right\|_{2,2} O\left(\frac{L^2\omega^{1/3}\sqrt{m\log(m)}}{\sqrt{d}}\right)\left\|\frac{1}{b}\sum_{i \in S}\nabla_{\boldsymbol{\theta}}(\mathcal{L}_i \circ h_i)(\boldsymbol{\theta})\right\|_{2,\infty} \\ & + \eta^2 O\left(\left(\frac{\beta C(\mathcal{L})^2}{n\rho} + 2C(\nabla\mathcal{L})\right)\frac{nL^2m}{d}\right)\left\|\frac{1}{b}\sum_{i \in S}\nabla_{\boldsymbol{\theta}}(\mathcal{L}_i \circ h_i)(\boldsymbol{\theta})\right\|_{2,\infty}^2\end{aligned} \tag{62}$$

where we refer to (54) for the form of $\nabla_{\boldsymbol{\theta}}(R \circ \mathcal{L} \circ h)(\boldsymbol{\theta})$ and to (53) for the form of $\nabla_h(R \circ \mathcal{L})(h(\boldsymbol{\theta}))$.

In addition, we make the assumption that for the set of values of $\boldsymbol{\theta}$ considered the hardness weighted sampling probabilities admit an upper-bound

$$\alpha = \min_{\boldsymbol{\theta}}\min_i \bar{p}_i(\mathcal{L}(\boldsymbol{\theta})) > 0 \tag{63}$$

Which is always satisfied under assumption 5.2 for Kullback-Leibler ϕ-divergence, and for any ϕ-divergence satisfying assumption 3.1 with a robustness parameter β small enough.

Let $\mathbb{E}_S$ be the expectation with respect to S. Applying $\mathbb{E}_S$ to (62), we obtain

$$\begin{aligned}&\mathbb{E}_S\left[R(\mathcal{L}(h(\boldsymbol{\theta}'))\right] \\ &\leq R(\mathcal{L}(h(\boldsymbol{\theta})) - \eta\left\|\nabla_{\boldsymbol{\theta}}(R \circ \mathcal{L} \circ h)(\boldsymbol{\theta})\right\|_{2,2}^2 \\ &\quad + \eta\left\|\nabla_h(R \circ \mathcal{L})(h(\boldsymbol{\theta}))\right\|_{2,2} O\left(\frac{nL^2\omega^{1/3}\sqrt{m\log(m)}}{\sqrt{d}}\right)\sqrt{\sum_{i=1}^n \max_l\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}_l}(\mathcal{L}_i \circ h_i)(\boldsymbol{\theta})\right\|^2} \\ &\quad + \eta^2 O\left(\left(\frac{\beta C(\mathcal{L})^2}{n\rho} + 2C(\nabla\mathcal{L})\right)\frac{nL^2m}{d}\right)\frac{1}{\alpha}\sum_{i=1}^n \max_l\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}_l}(\mathcal{L}_i \circ h_i)(\boldsymbol{\theta})\right\|^2\end{aligned} \tag{64}$$

where we have used the following results:

- For any integer $k \geq 1$, and all $(a_i)_{i=1}^n \in \left(\mathbb{R}^k\right)^n$, we have (see the proof in C.6.4)

$$\mathbb{E}_S\left[\frac{1}{b}\sum_{i \in S} a_i\right] = \mathbb{E}_{\bar{p}(\hat{\mathcal{L}})}\left[a_i\right] \tag{65}$$

- Using (65) for $(a_i)_{i=1}^n = (\nabla_{\boldsymbol{\theta}}(\mathcal{L}_i \circ h_i)(\boldsymbol{\theta}))_{i=1}^n$, and the chain rule (54)

$$\mathbb{E}_S\left[\frac{1}{b}\sum_{i \in S}\nabla_{\boldsymbol{\theta}}(\mathcal{L}_i \circ h_i)(\boldsymbol{\theta})\right] = \sum_{i=1}^n \bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}}(\mathcal{L}_i \circ h_i)(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}}(R \circ \mathcal{L} \circ h)(\boldsymbol{\theta}) \tag{66}$$

- Using the triangular inequality

$$\left\|\frac{1}{b}\sum_{i \in S}\nabla_{\boldsymbol{\theta}}(\mathcal{L}_i \circ h_i)(\boldsymbol{\theta})\right\|_{2,\infty} \leq \frac{1}{b}\sum_{i \in S}\left\|\nabla_{\boldsymbol{\theta}}(\mathcal{L}_i \circ h_i)(\boldsymbol{\theta})\right\|_{2,\infty} \tag{67}$$

  And using (65) for $(a_i)_{i=1}^n = \left(\left\|\nabla_{\boldsymbol{\theta}}(\mathcal{L}_i \circ h_i)(\boldsymbol{\theta})\right\|_{2,\infty}\right)_{i=1}^n$,

$$\begin{aligned}\mathbb{E}_S\left[\left\|\frac{1}{b}\sum_{i \in S}\nabla_{\boldsymbol{\theta}}(\mathcal{L}_i \circ h_i)(\boldsymbol{\theta})\right\|_{2,\infty}\right] &\leq \sum_{i=1}^n \bar{p}_i(\hat{\mathcal{L}})\left\|\nabla_{\boldsymbol{\theta}}(\mathcal{L}_i \circ h_i)(\boldsymbol{\theta})\right\|_{2,\infty} \\ &\leq \sum_{i=1}^n \max_l\left\|\nabla_{\boldsymbol{\theta}_l}(\bar{p}_i(\hat{\mathcal{L}})\,\mathcal{L}_i \circ h_i)(\boldsymbol{\theta})\right\|_2 \\ &\leq \sqrt{n}\sqrt{\sum_{i=1}^n \max_l\left\|\nabla_{\boldsymbol{\theta}_l}(\bar{p}_i(\hat{\mathcal{L}})\,\mathcal{L}_i \circ h_i)(\boldsymbol{\theta})\right\|_2^2}\end{aligned} \tag{68}$$

where we have used Cauchy-Schwarz inequality for the last inequality.

- Using (67) and the convexity of the function $x \mapsto x^2$

$$\left\| \frac{1}{b} \sum_{i \in S} \nabla_{\boldsymbol{\theta}} (\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}) \right\|_{2,\infty}^2 \leq \frac{1}{b} \sum_{i \in S} \left\| \nabla_{\boldsymbol{\theta}} (\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}) \right\|_{2,\infty}^2 \tag{69}$$

And using (65) for $(a_i)_{i=1}^n = \left(\|\nabla_{\boldsymbol{\theta}} (\mathcal{L}_i \circ h_i)(\boldsymbol{\theta})\|_{2,\infty}^2 \right)_{i=1}^n$,

$$\begin{aligned}
\mathbb{E}_S \left[\left\| \frac{1}{b} \sum_{i \in S} \nabla_{\boldsymbol{\theta}} (\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}) \right\|_{2,\infty}^2 \right] &\leq \sum_{i=1}^n \bar{p}_i(\hat{\mathcal{L}}) \left\| \nabla_{\boldsymbol{\theta}} (\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}) \right\|_{2,\infty}^2 \\
&\leq \sum_{i=1}^n \frac{1}{\bar{p}_i(\hat{\mathcal{L}})} \max_l \left\| \nabla_{\boldsymbol{\theta}_l} (\bar{p}_i(\hat{\mathcal{L}}) \underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}) \right\|_2^2 \\
&\leq \frac{1}{\alpha} \sum_{i=1}^n \max_l \left\| \nabla_{\boldsymbol{\theta}_l} (\bar{p}_i(\hat{\mathcal{L}}) \underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}) \right\|_2^2
\end{aligned} \tag{70}$$

Important Remark: It is worth noting the apparition of α (63) in (70). If we were using a uniform sampling as for ERM (i.e. for DRO in the limit $\beta \to 0$), we would have $\alpha = \frac{1}{n}$. So although our inequality (70) may seem brutal, it is consistent with equation (13.2) in (Allen-Zhu et al., 2019) and the corresponding inequality in the case of ERM.

The rest of the proof of convergence will consist in proving that $\eta \left\|\nabla_{\boldsymbol{\theta}} (R \circ \mathcal{L} \circ h)(\boldsymbol{\theta})\right\|_{2,2}^2$ dominates the two last terms in (62). As a result, we can already state that either the robustness parameter β, or the learning rate η will have to be small enough to control α. This is consistent with what we observed in our experiments.

Indeed, combining (62) with the chain rule (54), and the gradient bound Theorem C.2 where we use our $\hat{\mathcal{L}}$ defined in (59)

$$\begin{aligned}
\mathbb{E}_S \left[R(\mathcal{L}(h(\boldsymbol{\theta}'))) \right] \leq R(\mathcal{L}(h(\boldsymbol{\theta})) &- \Omega\left(\frac{\eta m \delta}{dn^2} \right) \sum_{i=1}^n \left\| \bar{p}_i(\hat{\mathcal{L}}) \nabla_{h_i} \underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta})) \right\|_2^2 \\
&+ \eta O\left(\frac{nL^2 \omega^{1/3} \sqrt{m \log(m)}}{\sqrt{d}} \right) O\left(\sqrt{\frac{m}{d}} \right) \sum_{i=1}^n \left\| \bar{p}_i(\hat{\mathcal{L}}) \nabla_{h_i} \underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta})) \right\|_2^2 \\
&+ \eta^2 O\left(\left(\frac{\beta C(\mathcal{L})^2}{n\rho} + 2C(\nabla \mathcal{L}) \right) \frac{nL^2 m}{d} \right) O\left(\frac{m}{d\alpha} \right) \sum_{i=1}^n \left\| \bar{p}_i(\hat{\mathcal{L}}) \nabla_{h_i} \underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta})) \right\|_2^2 \\
\leq R(\mathcal{L}(h(\boldsymbol{\theta})) &- \Omega\left(\frac{\eta m \delta}{dn^2} \right) \sum_{i=1}^n \left\| \bar{p}_i(\hat{\mathcal{L}}) \nabla_{h_i} \underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta})) \right\|_2^2 \\
&+ O\left(\frac{\eta n L^2 m \omega^{1/3} \sqrt{\log(m)}}{d} + K \frac{\eta^2 (n/\alpha) L^2 m^2}{d^2} \right) \sum_{i=1}^n \left\| \bar{p}_i(\hat{\mathcal{L}}) \nabla_{h_i} \underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta})) \right\|_2^2
\end{aligned} \tag{71}$$

where we have used

$$K := \frac{\beta C(\mathcal{L})^2}{n\rho} + 2C(\nabla \mathcal{L}) \tag{72}$$

There are only two differences with equation (13.2) in (Allen-Zhu et al., 2019):

- in the last fraction we have n/α instead of n^2 (see remark C.6.3 for more details), and an additional multiplicative term K. So in total, this term differs by a multiplicative factor $\frac{\alpha n}{K}$ from the analogous term in the proof of (Allen-Zhu et al., 2019).
- we have $\sum_{i=1}^n \left\| \bar{p}_i(\hat{\mathcal{L}}) \nabla_{h_i} \mathcal{L}_i(h_i(\boldsymbol{\theta})) \right\|_2^2$ instead of $F(\mathbf{W}^{(t)})$. In fact they are analogous since in equation (13.2) in (Allen-Zhu et al., 2019), $F(\mathbf{W}^{(t)})$ is the squared norm of the

mean loss for the L^2 loss. We don't make such a strong assumption on the choice of $\mathcal{L}$ (see assumption 5.2). It is worth noting that the same analogy is used in (Allen-Zhu et al., 2019, Appendix A) where they extend their result to the mean loss with other objective function than the L^2 loss.

Our choice of learning rate in Theorem 5.2 can be rewritten as

$$\begin{aligned} \eta_{exact} &= \Theta\left(\frac{\alpha n^2 \rho}{\beta C(\mathcal{L})^2 + 2n\rho C(\nabla \mathcal{L})} \times \frac{b\delta d}{\text{poly}(n,L) m \log^2(m)}\right) \\ &= \Theta\left(\frac{\alpha n}{K} \times \frac{b\delta d}{\text{poly}(n,L) m \log^2(m)}\right) \\ &\leq \frac{\alpha n}{K} \times \eta' \end{aligned} \tag{73}$$

And we also have

$$\eta_{exact} \leq \eta' \tag{74}$$

where η' is the learning rate chosen in the proof of Theorem 2 in (Allen-Zhu et al., 2019). We refer the reader to (Allen-Zhu et al., 2019) for the details of the constant in "Θ" and the exact form of the polynom poly(n, L).

As a result, for $\eta = \eta_{exact}$, the term $\Omega\left(\frac{\eta m \delta}{dn^2}\right)$ dominates the other term of the right-hand side of inequality (71) as in the proof of Theorem 2 in (Allen-Zhu et al., 2019).

This implies that the conditions of Theorem C.2 are satisfied for all $\boldsymbol{\theta}^{(t)}$, and that we have for all iteration $t > 0$

$$\mathbb{E}_{S_t}\left[R(\mathcal{L}(h(\boldsymbol{\theta}^{(t+1)}))\right] \leq R(\mathcal{L}(h(\boldsymbol{\theta}^{(t)})) - \Omega\left(\frac{\eta m \delta}{dn^2}\right) \sum_{i=1}^{n} \left\|\bar{p}_i(\hat{\mathcal{L}}) \nabla_{h_i} \underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}^{(t)}))\right\|_2^2 \tag{75}$$

And using a result in Appendix A of (Allen-Zhu et al., 2019), since under assumption 5.2 the distributionally robust loss is non-convex and bounded, we obtain for all $\epsilon' > 0$

$$\left\|\nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta}^{(T)}))\right\|_{2,2} \leq \epsilon' \quad \text{if} \quad T = O\left(\frac{dn^2}{\eta \delta m \epsilon'^2}\right) \tag{76}$$

where according to (53)

$$\left\|\nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta}^{(T)}))\right\|_{2,2} = \sum_{i=1}^{n} \left\|\bar{p}_i(\hat{\mathcal{L}}) \nabla_{h_i} \underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}^{(t)}))\right\|_2^2 \tag{77}$$

However, we are interested in a bound on $\left\|\nabla_{\boldsymbol{\theta}} (R \circ \mathcal{L} \circ h)(\boldsymbol{\theta}^{(T)})\right\|_{2,2}$, rather than $\left\|\nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta}^{(T)}))\right\|_{2,2}$.

Using the gradient bound of Theorem C.2 and the chain rules (54) and (53)

$$\left\|\nabla_{\boldsymbol{\theta}} (R \circ \mathcal{L} \circ h)(\boldsymbol{\theta}^{(T)})\right\|_{2,2} \leq c_1 \sqrt{\frac{Lmn}{d}} \left\|\nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta}^{(T)}))\right\|_{2,2} \tag{78}$$

where $c_1 > 0$ is the constant hidden in $O\left(\sqrt{\frac{Lmn}{d}}\right)$.

So with $\epsilon' = \frac{1}{c_1}\sqrt{\frac{d}{Lmn}}\epsilon$, we finally obtain

$$\begin{aligned} \left\|\nabla_{\boldsymbol{\theta}} (R \circ \mathcal{L} \circ h)(\boldsymbol{\theta}^{(T)})\right\|_{2,2} &\leq c_1 \sqrt{\frac{Lmn}{d}} \left\|\nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta}^{(T)}))\right\|_{2,2} \\ &\leq c_1 \sqrt{\frac{Lmn}{d}} \epsilon' \\ &\leq \epsilon \end{aligned} \tag{79}$$

If

$$T = O\left(\frac{dn^2}{\eta \delta m \epsilon'^2}\right) = O\left(\frac{dn^2}{\eta \delta m} \frac{Lmn}{d\epsilon^2}\right) = O\left(\frac{Ln^3}{\eta \delta \epsilon^2}\right) \tag{80}$$

which concludes the proof.

C.6.4 PROOF OF TECHNICAL LEMMA 1

For any integer $k \geq 1$, and all $(a_i)_{i=1}^n \in \left(\mathbb{R}^k\right)^n$, we have

$$\begin{aligned}
\mathbb{E}_S\left[\frac{1}{b}\sum_{i\in S} a_i\right] &= \sum_{1\leq i_1,\dots,i_b\leq n}\left[\left(\prod_{k=1}^{n}\bar{p}_{i_k}(\hat{\mathcal{L}})\right)\frac{1}{b}\sum_{j=1}^{b} a_{i_j}\right] \\
&= \frac{1}{b}\sum_{1\leq i_1,\dots,i_b\leq n}\left[\sum_{j=1}^{b}\bar{p}_{i_j}(\hat{\mathcal{L}})\,a_{i_j}\left(\prod_{\substack{k=1\\k\neq j}}^{n}\bar{p}_{i_k}(\hat{\mathcal{L}})\right)\right] \\
&= \frac{1}{b}\sum_{j=1}^{b}\left[\sum_{1\leq i_1,\dots,i_b\leq n}\bar{p}_{i_j}(\hat{\mathcal{L}})\,a_{i_j}\left(\prod_{\substack{k=1\\k\neq j}}^{n}\bar{p}_{i_k}(\hat{\mathcal{L}})\right)\right] \\
&= \frac{1}{b}\sum_{j=1}^{b}\left[\left(\sum_{i_j=1}^{n}\bar{p}_{i_j}(\hat{\mathcal{L}})\,a_{i_j}\right)\prod_{\substack{k=1\\k\neq j}}^{n}\left(\sum_{i_k=1}^{n}\bar{p}_{i_k}(\hat{\mathcal{L}})\right)\right] \\
&= \frac{1}{b}\sum_{j=1}^{b}\left(\sum_{i=1}^{n}\bar{p}_{i}(\hat{\mathcal{L}})\,a_{i}\right) \\
&= \sum_{i=1}^{n}\bar{p}_{i}(\hat{\mathcal{L}})\,a_{i} \\
&= \mathbb{E}_{\bar{p}(\hat{\mathcal{L}})}\left[a_i\right]
\end{aligned} \tag{81}$$

C.7 CONVERGENCE OF ROBUST SGD WITH STALE LOSS HISTORY

The proof of the convergence of Algorithm 4.1 under the conditions of Theorem 5.2 follows the same structure as the proof of the convergence of Robust SGD with exact loss history C.6.3. We will reuse the intermediate results of C.6.3 when possible and focus on the differences between the two proofs due to the inexactness of the loss history.

Let an iteration number t, so that the warm-up of Algorithm 4.1 is already over at t.

Let $\boldsymbol{\theta}^{(t)} \in \left(\mathbb{R}^{m\times m}\right)^L$ the parameters of the deep neural network at iteration t.

We define the stale loss history at iteration t as

$$\hat{\mathcal{L}} = \left(\underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}^{(t_i(t))}))\right)_{i=1}^n \tag{82}$$

where for all i, $t_i(t) < t$ corresponds to the latest iteration before t at which the loss for example i has been updated. Or equivalently, it corresponds to the last iteration before t when example i was drawn to be part of a mini-batch.

Thanks to the warm-up stage of Algorithm 4.1, it is guaranteed that the loss value of every example has been computed at least once before we start using the adaptive sampling. As a result, for all iteration after the warm-up, the stale loss history $\hat{\mathcal{L}}$ is well defined.

We also define the exact loss history that is unknown in Algorithm 4.1, as

$$\breve{\mathcal{L}} = \left(\underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}^{(t)}))\right)_{i=1}^n \tag{83}$$

Remark on the warm-up stage of Algorithm 4.1: The iterations performed during the warm-up stage amounts to classic SGD to minimize the mean loss (4). As a result, the convergence results of (Allen-Zhu et al., 2019, Theorem 2) apply during the warm-up. This guarantees that the condition on $\boldsymbol{\theta}$ of Theorem 5.2 remains satisfied during the warm-up if it was satisfied by the initial parameters.

Similarly to (61) we define

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \frac{1}{b} \sum_{i \in S} \nabla_{\boldsymbol{\theta}} (\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}^{(t)}) \tag{84}$$

and using Theorem C.1, similarly to (62), we obtain

$$\begin{aligned} R(\mathcal{L}(h(\boldsymbol{\theta}^{(t+1)})) \leq & R(\mathcal{L}(h(\boldsymbol{\theta}^{(t)})) - \eta \langle \nabla_{\boldsymbol{\theta}} (R \circ \mathcal{L} \circ h)(\boldsymbol{\theta}^{(t)}), \frac{1}{b} \sum_{i \in S} \nabla_{\boldsymbol{\theta}} (\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}^{(t)}) \rangle \\ & + \eta \left\| \nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta}^{(t)})) \right\|_{1,2} O\left(\frac{L^2 \omega^{1/3} \sqrt{m \log(m)}}{\sqrt{d}} \right) \left\| \frac{1}{b} \sum_{i \in S} \nabla_{\boldsymbol{\theta}} (\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}^{(t)}) \right\|_{2,\infty} \\ & + \eta^2 O\left(\left(\frac{\beta C(\mathcal{L})^2}{n\rho} + 2C(\nabla \mathcal{L}) \right) \frac{nL^2 m}{d} \right) \left\| \frac{1}{b} \sum_{i \in S} \nabla_{\boldsymbol{\theta}} (\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}^{(t)}) \right\|_{2,\infty}^2 \end{aligned} \tag{85}$$

We can still define α as in (63)

$$\alpha = \min_{\boldsymbol{\theta}} \min_i \bar{p}_i(\mathcal{L}(\boldsymbol{\theta})) > 0 \tag{86}$$

where we are guaranteed that $\alpha > 0$ under assumptions 5.1.

Since Theorem C.2 is independent to the choice of $\hat{\mathcal{L}}$, taking the expectation with respect to S, similarly to (71), we obtain

$$\begin{aligned} \mathbb{E}_S \left[R(\mathcal{L}(h(\boldsymbol{\theta}^{(t+1)}))) \right] \leq & R(\mathcal{L}(h(\boldsymbol{\theta}^{(t)})) - \eta \langle \nabla_{\boldsymbol{\theta}} (R \circ \mathcal{L} \circ h)(\boldsymbol{\theta}^{(t)}), \sum_{i=1}^{n} \bar{p}_i(\hat{\mathcal{L}}) \nabla_{\boldsymbol{\theta}} (\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}^{(t)})) \rangle \\ & + \eta \left\| \nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta}^{(t)})) \right\|_{1,2} O\left(\frac{L^2 \omega^{1/3} \sqrt{nm \log(m)}}{\sqrt{d}} \right) \sqrt{\sum_{i=1}^{n} \left\| \bar{p}_i(\hat{\mathcal{L}}) \nabla_{h_i} \underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}^{(t)})) \right\|_2^2} \\ & + \eta^2 O\left(\left(\frac{\beta C(\mathcal{L})^2}{n\rho} + 2C(\nabla \mathcal{L}) \right) \frac{nL^2 m}{d} \right) O\left(\frac{m}{d\alpha} \right) \sum_{i=1}^{n} \left\| \bar{p}_i(\hat{\mathcal{L}}) \nabla_{h_i} \underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}^{(t)})) \right\|_2^2 \end{aligned} \tag{87}$$

where the differences with respect to (71) comes from the fact that $\hat{\mathcal{L}}$ is not the exact loss history here, i.e. $\hat{\mathcal{L}} \neq \breve{\mathcal{L}}$, which leads to

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} (R \circ \mathcal{L} \circ h)(\boldsymbol{\theta}^{(t)}) &= \sum_{i=1}^{n} \hat{p}_i(\breve{\mathcal{L}}) \nabla_{\boldsymbol{\theta}} (\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}^{(t)})) \\ &\neq \sum_{i=1}^{n} \bar{p}_i(\hat{\mathcal{L}}) \nabla_{\boldsymbol{\theta}} (\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}^{(t)})) \end{aligned} \tag{88}$$

And

$$\begin{aligned} \left\| \nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta}^{(t)})) \right\|_{1,2} &= \sum_{i=1}^{n} \left\| \hat{p}_i(\breve{\mathcal{L}}) \nabla_{h_i} \underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}^{(t)}))) \right\|_2 \\ &\neq \sum_{i=1}^{n} \left\| \hat{p}_i(\hat{\mathcal{L}}) \nabla_{h_i} \underset{i}{\mathcal{L}}(h_i(\boldsymbol{\theta}^{(t)}))) \right\|_2 \end{aligned} \tag{89}$$

Let

$$K' = C(\mathcal{L}) A(\nabla \mathcal{L}) \, O\left(\frac{\beta L m^{3/2} \log^2(m)}{\alpha n^{1/2} \rho d^{3/2} b \log\left(\frac{1}{1-\alpha}\right)} \right) \tag{90}$$

Where $C(\mathcal{L}) > 0$ is a constant such that $\mathcal{L}$ is $C(\mathcal{L})$-Lipschitz continuous, and $A(\nabla \mathcal{L}) > 0$ is a constant that bound the gradient of $\mathcal{L}$ with respect to its input.

$C(\mathcal{L})$ and $A(\nabla \mathcal{L})$ are guaranteed to exist under assumptions 5.1.

We can prove that, with probability at least $1 - \exp\left(-\Omega\left(\log^2(m)\right)\right)$,

- according to lemma C.7.1

$$\left\|\hat{p}(\hat{\mathcal{L}}) - \hat{p}(\breve{\mathcal{L}})\right\|_2 = \sqrt{\sum_{i=1}^{n}\left(\hat{p}_i(\hat{\mathcal{L}}) - \hat{p}_i(\breve{\mathcal{L}})\right)^2} \leq \eta\alpha K' \tag{91}$$

- according to lemma C.7.2

$$\begin{aligned}\Bigg|\langle\nabla_{\boldsymbol{\theta}}(R \circ \mathcal{L} \circ h)(\boldsymbol{\theta}^{(t)}) - \sum_{i=1}^{n}\bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}}(\mathop{\mathcal{L}}_i \circ h_i)(\boldsymbol{\theta}^{(t)})), \sum_{i=1}^{n}\bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}}(\mathop{\mathcal{L}}_i \circ h_i)(\boldsymbol{\theta}^{(t)}))\rangle\Bigg| \\ \leq \eta\frac{m}{d}K'\sum_{i=1}^{n}\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}}(\mathop{\mathcal{L}}_i \circ h_i)(\boldsymbol{\theta}^{(t)}))\right\|_2^2\end{aligned} \tag{92}$$

- according to lemma C.7.3

$$\left\|\nabla_h(R \circ \mathcal{L})(h(\boldsymbol{\theta}^{(t)}))\right\|_{1,2} \leq \left(\sqrt{n} + \eta K'\right)\sqrt{\sum_{i=1}^{n}\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}}(\mathop{\mathcal{L}}_i \circ h_i)(\boldsymbol{\theta}^{(t)}))\right\|_2^2} \tag{93}$$

Combining those three inequalities with (87) we obtain

$$\begin{aligned}&\mathbb{E}_S\left[R(\mathcal{L}(h(\boldsymbol{\theta}^{(t+1)})))\right] - R(\mathcal{L}(h(\boldsymbol{\theta}^{(t)})) \leq \\ &\qquad \eta\left[-\Omega\left(\frac{m\delta}{dn^2}\right) + O\left(\frac{nL^2m\omega^{1/3}\sqrt{\log(m)}}{d}\right)\right]\sum_{i=1}^{n}\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{h_i}\mathop{\mathcal{L}}_i(h_i(\boldsymbol{\theta}^{(t)}))\right\|_2^2 \\ &\qquad \eta^2 O\left(K\frac{(n/\alpha)L^2m^2}{d^2} + \left(1 + \frac{m}{d}\right)K'\right)\sum_{i=1}^{n}\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{h_i}\mathop{\mathcal{L}}_i(h_i(\boldsymbol{\theta}^{(t)}))\right\|_2^2\end{aligned} \tag{94}$$

One can see that compared to (71), there is only the additional term $\left(1 + \frac{m}{d}\right)K'$.

Using our choice of η,

$$\eta = \eta_{stale} \leq O\left(\frac{\delta}{n^2K'}\eta_{exact}\right) \tag{95}$$

where η_{exact} is the learning rate of Theorem 5.1, we have

$$\Omega\left(\frac{\eta m\delta}{dn^2}\right) \geq O\left(\eta^2\left(1 + \frac{m}{d}\right)K'\right) \tag{96}$$

As a result, $\eta^2\left(1 + \frac{m}{d}\right)K'$ is dominated by the term $\Omega\left(\frac{\eta m\delta}{dn^2}\right)$

In addition, since $\eta_{stale} \leq \eta_{exact}$, $\Omega\left(\frac{\eta m\delta}{dn^2}\right)$ still dominates also the ther terms as in the proof of Theorem 5.1.

As a consequence, we obtain as in (75) that for any iteration $t > 0$ (after the end of the warm-up)

$$\mathbb{E}_{S_t}\left[R(\mathcal{L}(h(\boldsymbol{\theta}^{(t+1)})))\right] \leq R(\mathcal{L}(h(\boldsymbol{\theta}^{(t)})) - \Omega\left(\frac{\eta m\delta}{dn^2}\right)\sum_{i=1}^{n}\left\|\bar{p}_i(\hat{\mathcal{L}})\nabla_{h_i}\mathop{\mathcal{L}}_i(h_i(\boldsymbol{\theta}^{(t)}))\right\|_2^2 \tag{97}$$

This concludes the proof using the same arguments as in the end of the proof of Theorem 5.1 starting from (75).

C.7.1 PROOF OF TECHNICAL LEMMA 2

Using Lemma 4.2 and Lemma 4.1 we obtain

$$\begin{aligned}\left\|\hat{p}(\hat{\mathcal{L}})-\hat{p}(\check{\mathcal{L}})\right\|_2 &= \left\|\nabla_v R(\hat{\mathcal{L}})-\nabla_v R(\check{\mathcal{L}})\right\|_2 \\ &\leq \frac{\beta}{n\rho}\left\|\hat{\mathcal{L}}-\check{\mathcal{L}}\right\|_2\end{aligned} \tag{98}$$

Using assumptions 5.2 and (Allen-Zhu et al., 2019, Claim 11.2)

$$\begin{aligned}\left\|\hat{p}(\hat{\mathcal{L}})-\hat{p}(\check{\mathcal{L}})\right\|_2 &\leq \frac{\beta}{n\rho}\sqrt{\sum_{i=1}^{n}\left(\underset{i}{\mathcal{L}}\circ h_i(\boldsymbol{\theta}^{(t)})-\underset{i}{\mathcal{L}}\circ h_i(\boldsymbol{\theta}^{(t_i(t))})\right)^2} \\ &\leq \frac{\beta}{n\rho}C(\mathcal{L})C(h)\sqrt{\sum_{i=1}^{n}\left\|\boldsymbol{\theta}^{(t)}-\boldsymbol{\theta}^{(t_i(t))}\right\|_{2,2}^2} \\ &\leq C(\mathcal{L})O\left(\frac{\beta L m^{1/2}}{n\rho d^{1/2}}\right)\sqrt{\sum_{i=1}^{n}\left\|\boldsymbol{\theta}^{(t)}-\boldsymbol{\theta}^{(t_i(t))}\right\|_{2,2}^2}\end{aligned} \tag{99}$$

Where $C(\mathcal{L})$ is the constant of Lipschitz continuity of the per-example loss $\mathcal{L}$ (see assumptions 5.2) and $C(h)$ is the constant of Lipschitz continuity of the deep neural network h with respect to its parameters $\boldsymbol{\theta}$.

By developing the recurrence formula of $\boldsymbol{\theta}^{(t)}$ (84), we obtain

$$\begin{aligned}\left\|\hat{p}(\hat{\mathcal{L}})-\hat{p}(\check{\mathcal{L}})\right\|_2 &\leq C(\mathcal{L})O\left(\frac{\beta L m^{1/2}}{n\rho d^{1/2}}\right)\sqrt{\sum_{i=1}^{n}\left\|\boldsymbol{\theta}^{(t_i(t))}-\left(\sum_{\tau=t_i(t)}^{t-1}\frac{\eta}{b}\sum_{j\in S_\tau}\nabla_{\boldsymbol{\theta}}(\underset{j}{\mathcal{L}}\circ h_j)(\boldsymbol{\theta}^{(\tau)})\right)-\boldsymbol{\theta}^{(t_i(t))}\right\|_{2,2}^2} \\ &\leq \eta C(\mathcal{L})O\left(\frac{\beta L m^{1/2}}{n\rho d^{1/2}}\right)\sqrt{\sum_{i=1}^{n}\left\|\sum_{\tau=t_i(t)}^{t-1}\frac{1}{b}\sum_{j\in S_\tau}\nabla_{\boldsymbol{\theta}}(\underset{j}{\mathcal{L}}\circ h_j)(\boldsymbol{\theta}^{(\tau)})\right\|_{2,2}^2}\end{aligned}$$

Let $A(\nabla\mathcal{L})$ a bound on the gradient of the per-example loss function. Using Theorem C.2 and the chain rule

$$\forall j,\,\forall\tau\quad \left\|\nabla_{\boldsymbol{\theta}}(\underset{j}{\mathcal{L}}\circ h_j)(\boldsymbol{\theta}^{(\tau)})\right\|_{2,2}\leq A(\nabla\mathcal{L})O\left(\frac{m}{d}\right) \tag{100}$$

And using the triangular inequality

$$\begin{aligned}\left\|\sum_{\tau=t_i(t)}^{t-1}\frac{1}{b}\sum_{j\in S_\tau}\nabla_{\boldsymbol{\theta}}(\underset{j}{\mathcal{L}}\circ h_j)(\boldsymbol{\theta}^{(\tau)})\right\|_{2,2} &\leq \sum_{\tau=t_i(t)}^{t-1}\frac{1}{b}\sum_{j\in S_\tau}\left\|\nabla_{\boldsymbol{\theta}}(\underset{j}{\mathcal{L}}\circ h_j)(\boldsymbol{\theta}^{(\tau)})\right\|_{2,2} \\ &\leq \sum_{\tau=t_i(t)}^{t-1}A(\nabla\mathcal{L})O\left(\frac{m}{d}\right) \\ &\leq A(\nabla\mathcal{L})O\left(\frac{m}{d}\right)(t-t_i(t))\end{aligned} \tag{101}$$

As a result, we obtain

$$\left\|\hat{p}(\hat{\mathcal{L}})-\hat{p}(\check{\mathcal{L}})\right\|_2 \leq \eta C(\mathcal{L})A(\nabla\mathcal{L})O\left(\frac{\beta L m^{3/2}}{n\rho d^{3/2}}\right)\sqrt{\sum_{i=1}^{n}(t-t_i(t))^2} \tag{102}$$

For all i and for any τ the probability that the sample i is not in batch S_τ is lesser than $(1-\alpha)^b$.

Therefore, for any $k \geq 1$ and for any t,

$$P\left(t - t_i(t) \geq k\right) \leq (1-\alpha)^{kb} \tag{103}$$

For $k \geq \frac{1}{b}\Omega\left(\frac{\log^2(m)}{\log\left(\frac{1}{1-\alpha}\right)}\right)$, we have $(1-\alpha)^{kb} \leq \exp\left(-\Omega\left(\log^2(m)\right)\right)$, and thus with probability at least $1 - \exp\left(-\Omega\left(\log^2(m)\right)\right)$,

$$\forall t, \quad t - t_i(t) \leq O\left(\frac{\log^2(m)}{b\log\left(\frac{1}{1-\alpha}\right)}\right) \tag{104}$$

As a result, we finally obtain that with probability at least $1 - \exp\left(-\Omega\left(\log^2(m)\right)\right)$,

$$\begin{aligned}\left\|\hat{p}(\hat{\mathcal{L}}) - \hat{p}(\breve{\mathcal{L}})\right\|_2 &\leq \eta C(\mathcal{L})A(\nabla\mathcal{L})O\left(\frac{\beta L m^{3/2}}{n\rho d^{3/2}}\right)\sqrt{n}O\left(\frac{\log^2(m)}{b\log\left(\frac{1}{1-\alpha}\right)}\right)\\ &\leq \eta\alpha O\left(\frac{\beta L m^{3/2}\log^2(m)}{\alpha n^{1/2}\rho d^{3/2} b\log\left(\frac{1}{1-\alpha}\right)}\right)\\ &\leq \eta\alpha K'\end{aligned} \tag{105}$$

C.7.2 Proof of technical lemma 3

Let us first denote

$$\begin{aligned}A &= \left|\langle\nabla_{\boldsymbol{\theta}}(R\circ\mathcal{L}\circ h)(\boldsymbol{\theta}^{(t)}) - \sum_{i=1}^n \bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}}(\underset{i}{\mathcal{L}}\circ h_i)(\boldsymbol{\theta}^{(t)})), \sum_{i=1}^n \bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}}(\underset{i}{\mathcal{L}}\circ h_i)(\boldsymbol{\theta}^{(t)}))\rangle\right|\\ &= \left|\langle\sum_{i=1}^n\left(\bar{p}_i(\breve{\mathcal{L}}) - \bar{p}_i(\hat{\mathcal{L}})\right)\nabla_{\boldsymbol{\theta}}(\underset{i}{\mathcal{L}}\circ h_i)(\boldsymbol{\theta}^{(t)})), \sum_{i=1}^n \bar{p}_i(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}}(\underset{i}{\mathcal{L}}\circ h_i)(\boldsymbol{\theta}^{(t)}))\rangle\right|\end{aligned} \tag{106}$$

Using Cauchy-Schwarz inequality

$$\begin{aligned}A &= \left|\sum_{i=1}^n\left(\bar{p}_i(\breve{\mathcal{L}}) - \bar{p}_i(\hat{\mathcal{L}})\right)\langle\nabla_{\boldsymbol{\theta}}(\underset{i}{\mathcal{L}}\circ h_i)(\boldsymbol{\theta}^{(t)})), \sum_{j=1}^n \bar{p}_j(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}}(\underset{j}{\mathcal{L}}\circ h_j)(\boldsymbol{\theta}^{(t)}))\rangle\right|\\ &\leq \left\|\hat{p}(\hat{\mathcal{L}}) - \hat{p}(\breve{\mathcal{L}})\right\|_2\sqrt{\sum_{i=1}^n\left(\langle\nabla_{\boldsymbol{\theta}}(\underset{i}{\mathcal{L}}\circ h_i)(\boldsymbol{\theta}^{(t)})), \sum_{j=1}^n \bar{p}_j(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}}(\underset{j}{\mathcal{L}}\circ h_j)(\boldsymbol{\theta}^{(t)}))\rangle\right)^2}\end{aligned} \tag{107}$$

Let

$$B = \langle\nabla_{\boldsymbol{\theta}}(\underset{i}{\mathcal{L}}\circ h_i)(\boldsymbol{\theta}^{(t)})), \sum_{j=1}^n \bar{p}_j(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}}(\underset{j}{\mathcal{L}}\circ h_j)(\boldsymbol{\theta}^{(t)}))\rangle \tag{108}$$

Using again Cauchy-Schwarz inequality

$$B \leq \left\|\nabla_{\boldsymbol{\theta}}(\underset{i}{\mathcal{L}}\circ h_i)(\boldsymbol{\theta}^{(t)}))\right\|_{2,2}\left\|\sum_{j=1}^n \bar{p}_j(\hat{\mathcal{L}})\nabla_{\boldsymbol{\theta}}(\underset{j}{\mathcal{L}}\circ h_j)(\boldsymbol{\theta}^{(t)}))\right\|_{2,2} \tag{109}$$

As a result, A becomes

$$
\begin{aligned}
A &\leq \left\|\hat{p}(\hat{\mathcal{L}}) - \hat{p}(\breve{\mathcal{L}})\right\|_2 \left\|\sum_{j=1}^{n} \bar{p}_j(\hat{\mathcal{L}}) \nabla_{\boldsymbol{\theta}}(\underset{j}{\mathcal{L}} \circ h_j)(\boldsymbol{\theta}^{(t)}))\right\|_{2,2} \sqrt{\sum_{i=1}^{n} \left\|\nabla_{\boldsymbol{\theta}}(\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}^{(t)}))\right\|_{2,2}^2} \\
&\leq \left\|\hat{p}(\hat{\mathcal{L}}) - \hat{p}(\breve{\mathcal{L}})\right\|_2 \left\|\sum_{j=1}^{n} \bar{p}_j(\hat{\mathcal{L}}) \nabla_{\boldsymbol{\theta}}(\underset{j}{\mathcal{L}} \circ h_j)(\boldsymbol{\theta}^{(t)}))\right\|_{2,2} \sqrt{\sum_{i=1}^{n} \frac{1}{\alpha^2} \left\|\bar{p}_j(\hat{\mathcal{L}}) \nabla_{\boldsymbol{\theta}}(\underset{i}{\mathcal{L}} \circ h_i)(\boldsymbol{\theta}^{(t)}))\right\|_{2,2}^2} \\
&\leq \frac{1}{\alpha} \left\|\hat{p}(\hat{\mathcal{L}}) - \hat{p}(\breve{\mathcal{L}})\right\|_2 \left\|\sum_{j=1}^{n} \bar{p}_j(\hat{\mathcal{L}}) \nabla_{\boldsymbol{\theta}}(\underset{j}{\mathcal{L}} \circ h_j)(\boldsymbol{\theta}^{(t)}))\right\|_{2,2}^2
\end{aligned} \tag{110}
$$

Using the triangular inequality, Theorem C.2, and Lemma C.7.1, we finally obtain

$$
\begin{aligned}
A &\leq \frac{m}{\alpha d} \left\|\hat{p}(\hat{\mathcal{L}}) - \hat{p}(\breve{\mathcal{L}})\right\|_2 \sum_{j=1}^{n} \left\|\bar{p}_j(\hat{\mathcal{L}}) \nabla_{h_j} \underset{j}{\mathcal{L}}(h_j(\boldsymbol{\theta}^{(t)}))\right\|_{2,2}^2 \\
&\leq \eta \frac{m}{d} K' \sum_{j=1}^{n} \left\|\bar{p}_j(\hat{\mathcal{L}}) \nabla_{h_j} \underset{j}{\mathcal{L}}(h_j(\boldsymbol{\theta}^{(t)}))\right\|_{2,2}^2
\end{aligned} \tag{111}
$$

C.7.3 Proof of technical lemma 4

We have

$$
\begin{aligned}
\left\|\nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta}^{(t)}))\right\|_{1,2} &= \sum_{j=1}^{n} \bar{p}_j(\breve{\mathcal{L}}) \left\|\nabla_{h_j} \underset{j}{\mathcal{L}}(h_j(\boldsymbol{\theta}^{(t)}))\right\|_{2,2} \\
&= \sum_{j=1}^{n} \bar{p}_j(\hat{\mathcal{L}}) \left\|\nabla_{h_j} \underset{j}{\mathcal{L}}(h_j(\boldsymbol{\theta}^{(t)}))\right\|_{2,2} \\
&+ \sum_{j=1}^{n} \left(\frac{\bar{p}_j(\breve{\mathcal{L}}) - \bar{p}_j(\hat{\mathcal{L}})}{\bar{p}_j(\hat{\mathcal{L}})}\right) \bar{p}_j(\hat{\mathcal{L}}) \left\|\nabla_{h_j} \underset{j}{\mathcal{L}}(h_j(\boldsymbol{\theta}^{(t)}))\right\|_{2,2}
\end{aligned} \tag{112}
$$

Using Cauchy-Schwarz inequality

$$
\left\|\nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta}^{(t)}))\right\|_{1,2} = \left(\sqrt{n} + \sqrt{\sum_{j=1}^{n} \left(\frac{\bar{p}_j(\breve{\mathcal{L}}) - \bar{p}_j(\hat{\mathcal{L}})}{\bar{p}_j(\hat{\mathcal{L}})}\right)^2}\right) \sqrt{\sum_{j=1}^{n} \left\|\bar{p}_j(\hat{\mathcal{L}}) \nabla_{h_j} \underset{j}{\mathcal{L}}(h_j(\boldsymbol{\theta}^{(t)}))\right\|_{2,2}^2} \tag{113}
$$

Using Lemma C.7.1

$$
\begin{aligned}
\sum_{j=1}^{n} \left(\frac{\bar{p}_j(\breve{\mathcal{L}}) - \bar{p}_j(\hat{\mathcal{L}})}{\bar{p}_j(\hat{\mathcal{L}})}\right)^2 &\leq \frac{1}{\alpha} \left\|\hat{p}(\hat{\mathcal{L}}) - \hat{p}(\breve{\mathcal{L}})\right\|_2 \\
&\leq \eta K'
\end{aligned} \tag{114}
$$

Therefor, we finally obtain

$$
\left\|\nabla_h (R \circ \mathcal{L})(h(\boldsymbol{\theta}^{(t)}))\right\|_{1,2} = \left(\sqrt{n} + \eta K'\right) \sqrt{\sum_{j=1}^{n} \left\|\bar{p}_j(\hat{\mathcal{L}}) \nabla_{h_j} \underset{j}{\mathcal{L}}(h_j(\boldsymbol{\theta}^{(t)}))\right\|_{2,2}^2} \tag{115}
$$