Regularized Gradient Clipping Provably Trains Wide and Deep Neural Networks

Matteo Tucat* and Anirbit Mukherjee†

Abstract. In this work, we instantiate a regularized form of the gradient clipping algorithm and prove that it can converge to the global minima of deep neural network loss functions provided that the net is of sufficient width. We present empirical evidence that our theoretically founded regularized gradient clipping algorithm is also competitive with the state-of-the-art deep-learning heuristics. Hence the algorithm presented here constitutes a new approach to rigorous deep learning. The modification we do to standard gradient clipping is designed to leverage the PL* condition, a variant of the Polyak-Łojasiewicz inequality which was recently proven [LZB20], to be true for various neural networks for any depth within a neighbourhood of the initialisation.

Key words. adaptive gradients, gradient clipping, neural network optimization, Polyak-Łojasiewicz inequality

1. Introduction. In various disciplines, ranging from control theory to machine learning theory there has been a long history of trying to understand the nature of convergence on non-convex objectives for first order optimization algorithms i.e algorithms which only have access to an (estimate of) the gradient of the objective [MC01, FGQ97]. The new avatar of this question in large dimension optimization problems that arise in modern machine learning applications (like with neural networks) motivate the need for finite time analysis of such algorithms. A challenging aspect of these modern use cases is their essential need to finely tune the hyper-parameters in there like, the step-size, momentum, and batch size. In the wake of this, the "adaptive gradient" algorithms such as Adam [KB14] (and its special case RMSProp) have become essentially indispensable for doing deep-learning, [SS19, MDB17, BAP+17]. A reason for the widespread popularity of RMSProp and Adam stems from the fact that it seems easy to find task-specific and useful neural nets where the default settings of these algorithms already work well. Adam-like methods use as their update direction a vector which is the image of a linear combination of some (or all) of the gradients seen until now, under a linear transformation (often called the "diagonal pre-conditioner") constructed out of the history of the gradients. It is generally believed that this "pre-conditioning" makes these algorithms much less sensitive to the selection of its hyper-parameters. A precursor to RMSProp and Adam was the AdaGrad algorithm, [DHS11].

Motivated by their far-reaching usefulness in the deep-learning community, adaptive gradients methods like RMSProp and Adam have attracted significant attempts at their theoretical justifications in the non-convex setting. But, to the best of our knowledge, there has never been a theoretical guarantee for any adaptive gradient algorithm to converge to the global minima of deep neural nets.

*Work done while a student at the Department of Computer Science, University of Manchester (matteotucat@gmail.com)

†Department of Computer Science, The University of Manchester, UK (anirbit.mukherjee@manchester.ac.uk).

In contrast to the above, in recent times a number of motivations have come to light to consider training algorithms beyond these conventional adaptive gradient algorithms [BWAA18]. In works like [SSG19, ZKV+20] a number of reasons have been pointed out as to how gradient clipping based adaptivity is better suited for deep-learning. In this kind of adaptivity we primarily seek for mechanisms to prevent the algorithm from using arbitrarily large gradients. Gradient clipping has been successfully deployed in a wide range of cases, particularly in natural language processing tasks such as GPTs [BMR+20] and LSTMs [MKS17], and more recently in computer vision tasks [BDSS21]. Clipping the gradient is also known to alleviate the exploding gradients problem in recurrent neural networks [PMB12], as well as help provide privacy guarantees in differentially private machine learning [ACG+16]; [MKH23].

Inspired by the above, in this work, we initiate a form of gradient clipping algorithm which in experiments we demonstrate to be competitive with Adam, stochastic gradient descent and standard gradient clipping – while also being guaranteed to train neural nets of arbitrary depth - when training on the squared loss and when sufficiently wide.

Summary of Results. In [ZHSJ19], the authors study the following specific form of gradient clipping (which from here onwards we will refer to as "standard gradient clipping" or "GClip")

Definition 1.1 **(GClip).** *For any $\eta, \gamma > 0$, the* GClip *algorithm for a differentiable objective function f is defined as,*

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - h(\boldsymbol{x}_t) \cdot \nabla f(\boldsymbol{x}_t), \text{ where } h(\boldsymbol{x}_t) \coloneqq \eta \cdot \min\left\{1, \frac{\gamma}{\|\nabla f(\boldsymbol{x}_t)\|}\right\}. \tag{1.1}$$

The γ term acts as the threshold gradient norm. To the best of knowledge the above has no known convergence guarantees for deep-learning and thus motivated we present a modification of GClip – which we refer to as δ–Regularized-GClip (or δ-GClip).

Definition 1.2 (δ–**Regularized-GClip).** *The* δ–Regularized-GClip *algorithm for a differentiable objective function f would be defined as,*

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - h(\boldsymbol{x}_t) \cdot \nabla f(\boldsymbol{x}_t), \text{ where } h(\boldsymbol{x}_t) \coloneqq \eta \cdot \min\left\{1, \max\left\{\delta, \frac{\gamma}{\|\nabla f(\boldsymbol{x}_t)\|}\right\}\right\} \tag{1.2}$$

for any $\eta, \gamma > 0$ and $\delta \in (0, 1)$.

Note that setting $\delta = 0$ in above recovers standard gradient clipping. *A stochastic version of the above would also be considered when under a certain noisy gradient setup we state a convergence result for it in Theorem 2.9.*

Note that the critical $\max\{\delta, ...\}$ term ensures $h(\boldsymbol{x}_t) \geq \eta\delta$, and thus preventing $h(\boldsymbol{x}_t)$ from vanishing as $\|\nabla f(\boldsymbol{x})\| \to \infty$. It is important to note that due to this modification, the distance between any two iterates $\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\|$ is not bounded as the gradient norm grows. In Section 2.2 we shall show that in practice δ can be chosen small enough such that the lower bound is never hit - but its presence is critical for the following convergence guarantee for deep-learning that we shall establish.

Theorem 1.3 **(Informal Theorem About δ–Regularized-GClip).** *Given a deep neural network that is sufficiently wide (parametric in δ), δ–Regularized-GClip will minimise the square loss to find a zero-loss solution at an exponential convergence rate, for any training data.*

To the best of our knowledge, the above establishes the first instance of an adaptive gradient algorithm that provably trains nets at any depth. Additionally, our experiments will also show that δ–Regularized-GClip is competitive with the state-of-the-art deep-learning optimizers.

Notation. We denote a Euclidian ball centered at $\boldsymbol{w}_0 \in \mathbb{R}^m$ with radius R as $B(\boldsymbol{w}_0, R) \coloneqq \{\boldsymbol{w} \in \mathbb{R}^m : \|\boldsymbol{w}_0 - \boldsymbol{w}\|_2 \leq R\}$. Unless otherwise stated, $\|.\|$ denotes the ℓ_2-norm for vectors and the spectral norm for matrices.

2. The Main Results.

Towards stating the main results we recall the following definition,

Definition 2.1 (μ-PL* Condition). *A non-negative loss function $\mathcal{L}$ is said to satisfy μ-PL* on a set $\mathcal{S} \subset \mathbb{R}^m$ if $\exists \mu > 0$ such that $\forall \boldsymbol{w} \in \mathcal{S} : \|\nabla \mathcal{L}(\boldsymbol{w})\|^2 \geq \mu \mathcal{L}(\boldsymbol{w})$.*

Further, we recall the following L-hidden layer feed-forward neural network architectures and their loss setups which were within the ambit of considerations in [LZB20].

Definition 2.2.

$$f(\boldsymbol{W}; \boldsymbol{x}) = \alpha^{(L+1)}, \quad \alpha^{(l)} = \sigma_l\left(\frac{1}{\sqrt{m_{l-1}}} \cdot W^{(l)} \alpha^{(l-1)}\right) \text{ for } l \in [1, L+1], \quad \alpha^{(0)} = \boldsymbol{x} \tag{2.1}$$

where m_l is the width of the lth layer, $\alpha^{(l)}$ is the output from the l-th layer. $W^l \in \mathbb{R}^{m_l \times m_{l-1}}$ represents the weights for the l-th layer and $m_{L+1} = 1$. σ_l is the activation function for the l-th layer. We assume that the last layer activation $\sigma^{(L+1)}$ is L_σ-Lipschitz continuous, β_σ-Lipschitz smooth (β_σ-smooth) and satisfies $|\sigma'_{L+1}(z)| \geq \rho > 0$.

We train f as in equation 2.1 using an n-sample training dataset, $\{\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i) \mid i = 1, ..., n\}$, and we denote the vector of outputs for all training samples as,
$\mathcal{F}(\boldsymbol{W}) = (f(\boldsymbol{W}; \boldsymbol{x}_1), ..., f(\boldsymbol{W}; \boldsymbol{x}_n)) \in \mathbb{R}^n$. We utilise the square loss, $\mathcal{L}(\boldsymbol{W}) = \frac{1}{2}\|\mathcal{F}(\boldsymbol{W}) - \boldsymbol{y}\|^2$.

Now we have all the requisite background to state the key theorem we will present in this work.

Theorem 2.3 **(δ–Regularized-GClip Provably Trains Wide and Deep Neural Nets).** *Suppose an overparametrised neural network $\mathcal{F}$ is being trained using the square loss, $\mathcal{L}(\boldsymbol{w})$, as specified in Definition 2.2. Then $\exists\ \lambda_0 > 0$ s.t for any $\eta, \mu, \delta > 0$ appropriately small enough, if the minimum width of the network satisfies,*

$$m = \tilde{\Omega}\left(\frac{nR^{6L+2}}{(\lambda_0 - \mu\rho^{-2})^2}\right) \text{ where } R = \frac{\eta\sqrt{2\beta}\sqrt{\mathcal{L}(\boldsymbol{w}_0)}}{1 - \sqrt{1 - \frac{1}{2} \cdot \eta\delta\mu}}. \tag{2.2}$$

then one can initialize the net s.t w.h.p over initialization the above loss is μ-PL* in the ball $B(\boldsymbol{w}_0, R)$ around initialization $\boldsymbol{w}_0$. Further, let $\beta_{\mathcal{F}}$ be s.t $\mathcal{F}(\boldsymbol{w})$ is locally $\beta_{\mathcal{F}}$-smooth in

$B(\boldsymbol{w}_0, R)$. *Then, training such a network using* $\delta-$*Regularized-GClip with* $\eta < \min\{\frac{1}{\beta_{\mathcal{F}}}, \frac{1}{\mu}\}$ *and* $\delta \in (0,1)$*, will result in geometric convergence to a global minimiser of* $\mathcal{L}$*,* $\boldsymbol{w}_* \in B(\boldsymbol{w}_0, R)$*, such that* $\mathcal{L}(\boldsymbol{w}_*) = 0$*. Furthermore,* $\delta-$*Regularized-GClip will converge with convergence rate,*

$$\mathcal{L}(\boldsymbol{w}_t) \leq \mathcal{L}(\boldsymbol{w}_0)(1 - \frac{1}{2} \cdot \eta\delta\mu)^t. \tag{2.3}$$

***Remark* 2.4.** *The assumptions of* $\eta < 1/\mu$ *and* $\delta < 1$ *imply* $(1 - \frac{1}{2} \cdot \eta\delta\mu) \in (\frac{1}{2}, 1)$*, hence* $\lim_{t\to\infty} \mathcal{L}(\boldsymbol{w}_t) = 0$.

The proof of the above theorem can be found in Section 4. In subsection 2.1 we state the lemmas that are required to prove this.

Next, we consider a stochastic version of our algorithm, defined as follows,

Definition 2.5 **(Stochastic** δ**–Regularized-GClip).** *The Stochastic* $\delta-$*Regularized-GClip algorithm for a differentiable function* $\mathcal{L}$ *would be defined as,*

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - h(\boldsymbol{g}_t)\cdot\boldsymbol{g}_t, \text{ where } h(\boldsymbol{g}_t) = \eta \min\left\{1, \max\left\{\delta, \frac{\gamma}{\|\boldsymbol{g}_t\|}\right\}\right\} \text{ and, } \mathbb{E}[\boldsymbol{g}_t \mid \boldsymbol{w}_t] = \nabla\mathcal{L}(\boldsymbol{w}_t) \tag{2.4}$$

for any $\eta, \gamma > 0$ *and* $\delta \in (0,1)$ *and an arbitrary choice of* $\boldsymbol{w}_1$*, the initial point.*

Towards analyzing the above we make the following assumptions,

***Assumption* 2.6.** $\exists\ \theta \geq 0$ *s.t.* $\forall \boldsymbol{w},\ \|\boldsymbol{g}(\boldsymbol{w}) - \nabla\mathcal{L}(\boldsymbol{w})\| \leq \theta$

***Assumption* 2.7.** $\mathcal{L}$ *is non-negatively lower bounded i.e.* $\min_{\boldsymbol{w}} \mathcal{L} = \mathcal{L}_* \geq 0$

***Assumption* 2.8.** $\mathcal{L}$ *is* β*-smooth*

Thus we have the following convergence theorem,

Theorem 2.9. *Given Assumptions 2.6, 2.7 and 2.8, and for an arbitrary choice of* $\epsilon > 0$*, let* $\epsilon' := \frac{\epsilon}{\theta}$*. Then for* $\beta = 1, \delta = \frac{1+2\epsilon'^2}{1+3\epsilon'^2}, \eta = \frac{1}{4} \cdot \frac{\epsilon'^2}{1+\epsilon'^2}$*, stochastic* $\delta-$*Regularized-GClip iterates satisfy the following inequality,*

$$\text{for, } T = \frac{\theta^4}{\epsilon^4}, \ \min_{t=1,\ldots,T} \mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2\right] \leq \mathcal{O}(\epsilon^2)$$

The proof of the above is given in Appendix A - where we first prove a slightly more general result than the above. We note that this convergence guarantee above does not need the gradient norms to be bounded as was also the case for standard stochastic gradient clipping, Theorem 7 in [ZHSJ19]. But, the convergence guarantee for standard stochastic clipping does not immediately hold as stated in [ZHSJ19] for the standard smoothness assumption that is used here. Additionally, unlike standard clipping, here we can get convergence guarantees in the deterministic ("full gradient") setting (Theorem 2.3) as well as the noisy setting – for the same clipping algorithm.

2.1. Intermediate Lemmas for Theorem 2.3.

Lemma 2.10. *Corresponding to constants $a, b > 0$ and $a\mu < 1$ suppose a loss function $\mathcal{L}$ is β-smooth, $\min \mathcal{L} = 0$, and satisfies the $\mu - \mathrm{PL}^*$ condition within a Euclidean ball $B(\boldsymbol{w}_0, R)$, with $R \geq \frac{b\sqrt{2\beta}\sqrt{\mathcal{L}(\boldsymbol{w}_0)}}{1-\sqrt{1-\frac{1}{2}\cdot a\mu}}$. Then there exists a global minimiser of $\mathcal{L}$, $\boldsymbol{w}_* \in B(\boldsymbol{w}_0, R)$ such that $\mathcal{L}(\boldsymbol{w}_*) = 0$. Furthermore, given a first order adaptive step size algorithm of the form,*

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - h(\boldsymbol{w}_t) \cdot \nabla \mathcal{L}(\boldsymbol{w}_t), \tag{2.5}$$

where $h(\boldsymbol{w}_t)$ is a time/iterate-dependent function such that $0 < a \leq h(\boldsymbol{w}_t) \leq b < \min\{\frac{1}{\beta}, \frac{1}{\mu}\}$, then the algorithm will converge with convergence rate,

$$\mathcal{L}(\boldsymbol{w}_t) \leq \mathcal{L}(\boldsymbol{w}_0)(1 - \frac{1}{2} \cdot a\mu)^t \tag{2.6}$$

Lemma 2.11. *The $\delta-$Regularized-GClip step size $h(\boldsymbol{w})$ is bounded $\eta\delta \leq h(\boldsymbol{w}) \leq \eta$, given that $0 < \delta < 1$.*

Lemma 2.12. *($\delta-$Regularized-GClip Converges on smooth PL^* functions) Corresponding to positive constants η, δ, β, μ s.t $\eta < \min\{1/\beta, 1/\mu\}$ and $0 < \delta < 1$, suppose there exists a loss function $\mathcal{L}$ that is β-smooth, lower bounded by 0, and satisfies the $\mu-\mathrm{PL}^*$ condition within an Euclidean ball $B(\boldsymbol{w}_0, R)$ where $R \geq \frac{\eta\sqrt{2\beta}\sqrt{\mathcal{L}(\boldsymbol{w}_0)}}{1-\sqrt{1-\frac{1}{2}\cdot\eta\delta\mu}}$. Then there exists a global minimiser of $\mathcal{L}$, $\boldsymbol{w}_* \in B(\boldsymbol{w}_0, R)$ such that $\mathcal{L}(\boldsymbol{w}_*) = 0$. Furthermore, $\delta-$Regularized-GClip will converge at rate,*

$$\mathcal{L}(\boldsymbol{w}_t) \leq \mathcal{L}(\boldsymbol{w}_0)(1 - \frac{1}{2} \cdot \eta\delta\mu)^t \tag{2.7}$$

The proofs for Lemmas 2.10, 2.11 and 2.12 can be found in Subsection 4.1.

2.2. Experimental Evidence for The Performance of $\delta-$Regularized-GClip. In this section we aim to demonstrate that the regularization term in $\delta-$Regularized-GClip helps improve the performance of standard gradient clipping – which anyway outperforms stochastic gradient descent (SGD) – and is in fact competitive when compared against the most popular optimizers such as Adam, even superseding it at times. We test in supervised classification as well as unsupervised distribution learning settings.

We perform four experiments, the first set is on the standard benchmark of a ResNet-18 [HZRS15] being trained on the CIFAR-10 [Kri09] dataset - which we recall is a 10−class image classification task with $50,000$ training images and $10,000$ test images. The second set of experiments is training a VAE model on the Fashion-MNIST dataset - with $60,000$ training samples and $10,000$ for testing. Further, we test both with learning rate scheduling – whereby η (or the learning rate) is reduced at certain points in the training – and without (constant η throughout).

Note that, in the (supervised) classification experiment the training is done on the cross-entropy loss and on ReLU gate nets and while using weight-decay (of $5e{-}4$). And the VAE

setup does not have a loss function in the same conventional sense as considered in the theorem earlier. *Hence these experiments demonstrate the efficacy of regularised gradient clipping beyond the ambit of the current theory.*

The code for the experiments can be found in our GitHub repository[1]. We built basic custom implementations of $\delta-$Regularized-GClip and standard GClip and used the standard Pytorch optimizers for SGD and Adam - which we recall is highly optimized. Hence we would be demonstrating performance of our modification in competitions which are a priori skewed in favour of the existing benchmarks.

In the legends of the figures, a notation of, SGD (0.1) stands for stochastic gradient descent with $\eta = 0.1$, $\delta-$GClip $(1; 1; 1e{-}8)$ is $\delta-$Regularized-GClip with $\eta = 1, \gamma = 1, \delta = 1e-8$, GClip $(5; 1)$ for standard gradient clipping with $\eta = 5, \gamma = 1$ and Adam (1) is notation for Adam with $\eta = 1$ – and similarly for other hyperparameter choices.

2.2.1. The Setup of the Experiments with ResNet-18 and CIFAR-10. The ResNet-18 was trained using the full training set using mini-batches of size 512. We tested all the following hyperparameter combinations, $\eta \in \{0.0001, 0.001, 0.01, 0.1, 1, 5\}$, $\gamma \in \{0.25, 1, 5, 10\}$ and $\delta \in \{1e{-}3, 1e{-}8\}$ for each optimizer. For Adam, only the learning rate (η) was modified, the rest were left at the PyTorch defaults ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1e{-}8$). In the case with scheduling the η value quoted in the legend denotes the η value at epoch 0 – i.e before any reductions by the scheduling algorithm are done.

Experiments Without Learning Rate Scheduling. In Figure 2.1 we only plot the best-performing (in terms of test accuracy) hyperparameter selection for each algorithm.

Experiments With Learning Rate Scheduling. In Figure 2.2 we show a repeat of the above experiments and again plot the best-performing hyperparameters. In here we start at larger η values and divide η by 10 at epochs 100 and 150, following the setup from [ZHSJ19]. See Appendix B, for a version of this experiment with no weight-decay.

We draw two primary conclusions from the above results. *Firstly,* that a very small value of δ in $\delta-$Regularized-GClip does not seem to have a significant effect either for loss minimization or test accuracy. The results for $\delta-$Regularized-GClip and standard GClip set to similar η and γ values, are practically identical in both scenarios for all small enough values of δ tried. As alluded to in the previous sections, the gradient norm would have to be larger than $\eta\gamma/\delta$ for the lower bound on $h(\boldsymbol{w}_t)$ to be attained - and even for the larger setting of $\delta = 1e{-}3$ and a typical $\gamma = 0.25$ setting requires a gradient norm of over 250, which is only infrequently seen along the optimization trajectory.

Secondly, though Adam attained the best test accuracy without learning rate scheduling by a margin of about $\sim 1\%$ compared to both $\delta-$Regularized-GClip and standard gradient clipping – but all other optimizers superseded it by $\sim 3\%$ when learning rate scheduling was used. *The best performance with scheduling (which is by our regularized gradient clipping) is better than for any algorithm without scheduling.* Interestingly, with learning rate scheduling Adam

[1]Experiment code is available at https://github.com/matteo-tucat/delta-gradient-clipping.

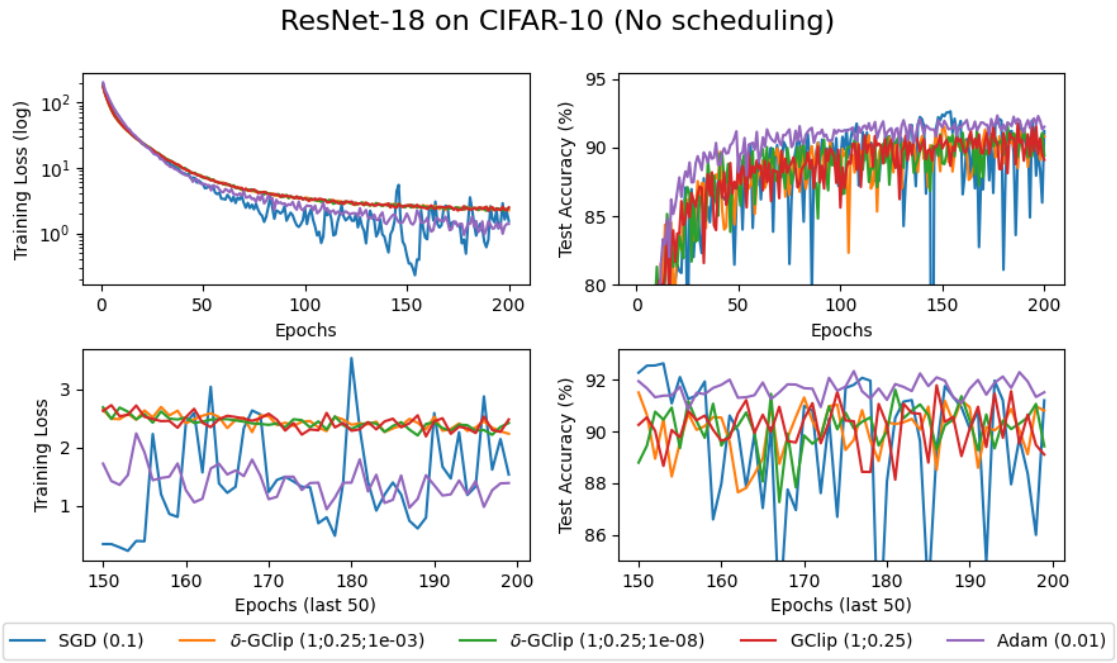

Figure 2.1: $\delta-$Regularized-GClip ($\delta-$GClip) is competitive against SOTA heuristics for training ResNet-18 on CIFAR-10 without learning-rate scheduling

performed the best in terms of minimizing the training loss while SGD performed the worst, even though SGD's solution seems to generalize significantly better (as shown by the ~ 3 percentage point higher test accuracy).

The significant ability of δ-regularised gradient clipping to exploit learning rate scheduling motivates an interesting direction for future exploration in theory.

2.2.2. VAE on Fashion-MNIST. We performed the VAE training experiment both with and without scheduling when training on the Fashion MNIST dataset. We tested the following hyperparmeter choices $\eta \in \{1e{-}5, 1e{-}4, 1e{-}3, 1e{-}2\}$, $\gamma \in \{10, 50, 200, 500\}$, $\delta \in \{0.01, 0.1, 1\}$. We utilize the same scheduling as in the ResNet experiment (η division by 10 at epochs 100 and 150) - and the results are given in Figure 2.3.

The VAE results with (and without - though not shown here) learning rate scheduling supports our earlier observations that the added regularization term of δ helps the performance w.r.t that of GClip at the same values of step-length and clipping threshold which anyway outperforms SGD. And it is only mildly underperforming with respect to Adam.

We therefore conclude from our experiments that $\delta-$Regularized-GClip clipping remains competitive with current optimizers, while offering the significant benefit of provable deep neural network training.

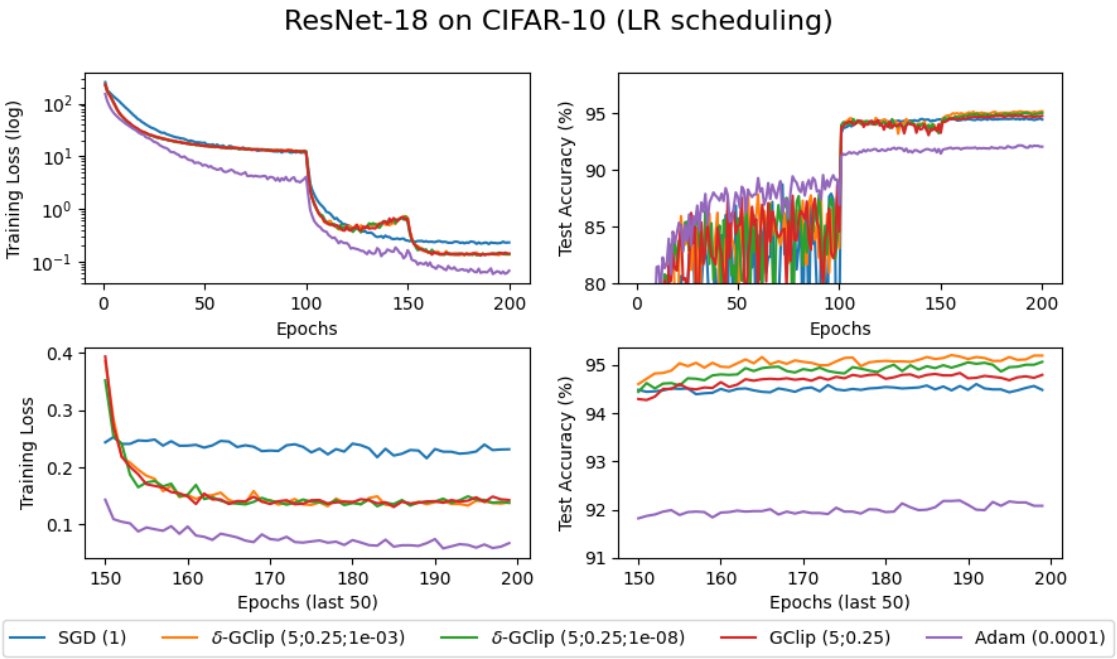

Figure 2.2: $\delta-$Regularized-GClip ($\delta-$GClip) outperforms other optimizers for training ResNet-18 on CIFAR-10 with learning-rate scheduling

3. Discussion. In this section, we will give a summary of the state-of-the-art literature about provable deep-learning algorithms - particularly focusing on the theoretical attempts that have been made so far in settings that are closest to real-world implementations.

Literature Review of Theory for Adam. Adam was proposed in [KB14] and in [RKK18] it was proved that for common hyperparameter choices ($\beta_1 < \sqrt{\beta_2}$), there exists a stochastic convex optimisation problem where Adam does not converge. They presented a modification to Adam that provably converges for online convex optimization. In [DMU18] the authors analyse the convergence of Adam in the deterministic case, without the use of convexity, but leveraging Lipschitz smoothness and a bounded gradient norm they gave the first proof of Adam's convergence to an ε-stationary point for such non-convex functions.

For the same optimization target as above, in [CLSH18], a convergence rate of $O(\log T/\sqrt{T})$ was shown for Adam-like adaptive gradient algorithms under the assumption of a bounded gradient oracle. Later, a burn-in stage was added in [SRK+19] to prove a $O(1/\sqrt{T})$ convergence rate. In [CG18], the authors introduced a partial adaptive parameter and proved convergence to criticality for a class of adaptive gradient algorithms, which does not include RMSProp. It was shown in [ZSJ+19] that generic Adam (including RMSProp) converges with high probability under certain decaying conditions on β_2 and step size - in contrast to the usual implementations. In [WWB19] the authors proved similar convergence results for AdaGrad which is a special case of RMSProp.

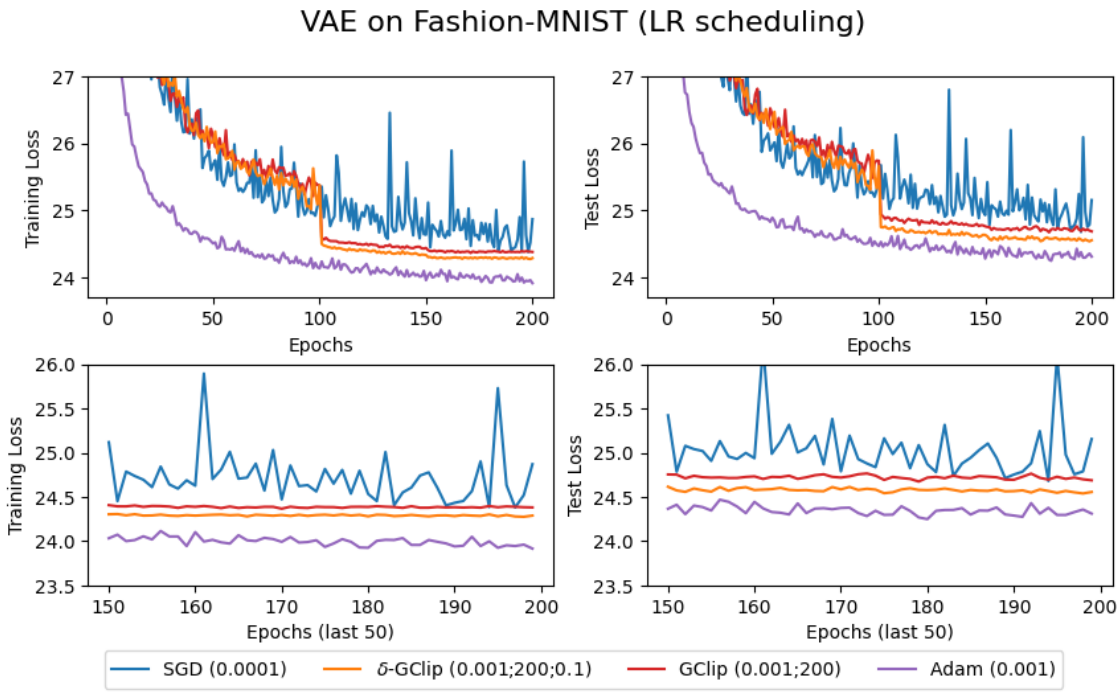

Figure 2.3: $\delta-$Regularized-GClip ($\delta-$ GClip) is competitive against SOTA heuristics for training a VAE on the Fashion MNIST dataset with learning-rate scheduling

Review of Theory for Adaptive Gradient Methods Training Neural Nets. In contrast to the convergence to criticality results mentioned above - there have also been works providing guarantees of convergence to global minima for adaptive methods in shallow neural network training scenarios. [WDW19] provides a proof of the convergence of the AdaLoss adaptive algorithm to global minima on two-layer network, with widths large enough to be in the NTK regime. [ZCLG21] provides a proof of Adam's global convergence on two-layer convolutional neural networks to a zero error solution whilst utilising weight decay regularisation. The authors further provide evidence that although both GD and Adam converge to zero error solutions, GD's solution generalises significantly better. In the context of Generative Adversarial Networks (GANs), [DL22] analysed the performance of Adam-like algorithms and proved the convergence of Extra Gradient AMSGrad to an ε-stationary point under novel assumptions they motivated.

Literature Review of Gradient Clipping. In the smooth non-convex case, [ZHSJ19] proved the convergence of deterministic gradient clipping to an ε-stationary point under a new smoothness assumption that is strictly weaker than standard Lipschitz smoothness. Their provided iteration complexity implies that gradient clipping can converge faster than gradient descent (in constants), while achieving $\mathcal{O}(\varepsilon)-$criticality in $\mathcal{O}(\varepsilon^{-2})$ steps. The authors provide a similar analysis in the stochastic case, with the additional assumption of either a bound on the noise of the stochastic gradient or its distribution being symmetric sub-Gaussian. It is important

to note that the provided stochastic iteration complexity does not supersede that of SGD in the general case. They had pointed out, possibly for the first time, that gradient clipping can converge, in deterministic as well as noisy settings, on smooth functions without the need for gradients to be bounded.

In [ZKV+20] the authors utilise Lipschitz smoothness while working with non-convex targets and having heavy-tailed gradient stochasticity to achieve $\mathcal{O}(1/t^{\frac{1}{4}})$ close convergence to criticality in $t-$steps – which matches that of SGD in the non-heavy tailed setting. In there the authors gave a lower bound in the same setting, which matches upto constants the run-time given above and thus proving that their convergence rate is worst-case optimal. Furthermore, the authors also consider non-smooth but strongly convex functions with a bound on the expected norm of the stochastic gradients – which we recall had appeared earlier in [SZ13] for non-heavy tailed settings – and achieve the same convergence, implying that the convergence rate is optimal even in the Lipschitz smooth and strongly convex setting.

We posit that from above kinds of analysis of adaptive algorithms (including GClip), either for depth 2 neural networks or in the more general (non-)convex settings, there is no obvious path towards provable convergence guarantees in deep neural network training for adaptive gradient algorithms. But, recently, in [LZB20], convergence guarantees where proven for (S)GD for sufficiently wide, and arbitrarily deep neural networks, by leveraging the novel PL* condition that the authors proved to be true for squared losses for such nets. Next we will briefly review those results.

Review of the PL* *Condition.* Our motivation behind studying the convergence characteristics of algorithms under the PL* condition comes from the paper [LZB20], where the authors prove that overparametrised feedforward, convolutional and residual (ResNet) neural networks can all satisfy the PL* condition within a finite radius of the initialisation, given that they are sufficiently wide. In partiicular, they showed that,

Theorem 3.1. *Any neural network of the form described in Definition 2.2, if randomly initialized s.t.* $\boldsymbol{W}_0^l \sim \mathcal{N}(0, I_{m_l \times m_{l-1}})$ *for* $l \in [0, L+1]$ *and defining* $\lambda_0 \coloneqq \lambda_{\min}(K_{\mathcal{F}}(\boldsymbol{W}_0)) > 0$ *where* $K_{\mathcal{F}}(\boldsymbol{W}) = \mathcal{D}\mathcal{F}(\boldsymbol{W})\mathcal{D}\mathcal{F}(\boldsymbol{W})^\top$, *then for any* $\mu \in (0, \lambda_0\rho^2)$ *and the minimum layer width of the network being,*

$$m = \tilde{\Omega}\left(\frac{nR^{6L+2}}{(\lambda_0 - \mu\rho^{-2})^2}\right), \tag{3.1}$$

the μ*-PL* condition holds for the square loss in the ball* $B(\boldsymbol{W}_0, R)$ *where* R *is a finite radius.*

Therefore, a path opened up, that by proving that the iterates of our $\delta-$Regularized-GClip algorithm never leave a ball of finite radius, and proving the convergence of $\delta-$Regularized-GClip on locally smooth μ-PL* functions we can argue for the algorithm's convergence to the loss global minima in such neural networks.

Conclusion. In this work, we have presented a new adaptive algorithm, $\delta-$Regularized-GClip, as a modification to the standard gradient clipping algorithm. In contrast to *all* previous attempts at finding good adaptive gradient methods, we proved that our $\delta-$Regularized-GClip algorithm can train deep neural networks (at any depth) with arbitrary data and while

training on the squared loss. Additionally, we have also given experimental evidence that our algorithm can compete and sometimes outperform the deep-learning algorithms in current use. Our proof critically hinges on the interplay between the modification we do to standard gradient clipping and the $\mu - \mathrm{PL}^*$ condition that has previously been shown to be true for squared losses on deep nets of sufficient width.

Our work suggests an immediate direction of future research into establishing convergence guarantees for regularized gradient clipping on various other standard losses in use like cross-entropy and for nets with ReLU activation. We note that recently reported heuristics which are particularly good for LLM training, [LLH+23] can also be seen as modifications of the clipping algorithm. We envisage that exciting lines of investigation could open up in trying explore the efficacy of these new developments crossed with the provably good modifications of gradient clipping that we instantiated here.

4. Methods. In this section we will give the proofs for the main theorems presented in this work.

Proof. **(of Theorem 2.3)**

Firstly, we invoke the assumption that the initialization is s.t that the conditions of Theorem 3.1 apply - which we know from therein to be a high-probability event. In particular we conclude that $\mathcal{L}$ satisfies μ–PL^* within a finite ball $\mathrm{B}(\boldsymbol{w}_0, R)$ for some $R > 0$ and that the tangent kernel at initialization is positive definite.

If L_σ and β_σ are the Lipschitz constant and the Lipschitz smoothness coefficients for the activation σ then it was shown in [LZB20], that we have for the prediction map $\mathcal{F}$, its Lipschitz constant $L_{\mathcal{F}} \leq L_\sigma \left(\sqrt{\|K_{\mathcal{F}}(\boldsymbol{w}_0)\|} + R\sqrt{n} \cdot O(R^{3L}/\sqrt{m}) \right)$ as well as its smoothness constant $\beta_{\mathcal{F}} \leq \beta_\sigma L_\sigma \left(\sqrt{\|K_{\mathcal{F}}(\boldsymbol{w}_0)\|} + R\sqrt{n} \cdot O(R^{3L}/\sqrt{m}) \right) + L_\sigma \cdot O(R^{3L}/\sqrt{m})$. Where $K_{\mathcal{F}}$ is the neural tangent kernel (recall that $K_{\mathcal{F}} = \mathcal{D}\mathcal{F}(\boldsymbol{w})\mathcal{D}\mathcal{F}(\boldsymbol{w})^\top$).

By plugging in the lowerbound on m specified in the theorem, we get that both $L_{\mathcal{F}}$ and $\beta_{\mathcal{F}}$ are upper bounded by a constant and thus m independent. If $H_{\mathcal{L}}$ is the Hessian of the loss function, then by [LZB20] we also have that,

$$\beta_{\mathcal{L}} = \sup_{\boldsymbol{w} \in B(\boldsymbol{w}_0, R)} \|H_{\mathcal{L}}(\boldsymbol{w})\| \leq L_{\mathcal{F}}^2 + \beta_{\mathcal{F}} \cdot \|\mathcal{F}(\boldsymbol{w}_0) - \boldsymbol{y}\| \tag{4.1}$$

By [JGH18], we have that $\|\mathcal{F}(\boldsymbol{w}_0) - \boldsymbol{y}\|$ is also m independent with high probability for the given size of the net. Therefore, $\mathcal{L}$ can be said to be $\beta_{\mathcal{L}}$-smooth within $\mathrm{B}(\boldsymbol{w}_0, R)$, where $\beta_{\mathcal{L}}$ is m and thus R independent. Hence, we can say that for every $R > 0$, for some width which satisfies the given condition, the loss function is β-smooth (and by Theorem 3.1, μ-PL^*) in $\mathrm{B}(\boldsymbol{w}_0, R)$ with high probability.

Thus far the argument above was parametric in R. But given that we satisfy all the conditions to invoke Lemma 2.12 we can compute from it the minimum R value required such that the iterates of regularized gradient clipping never leave $\mathrm{B}(\boldsymbol{w}_0, R)$ i.e $R = \frac{\eta\sqrt{2\beta}\sqrt{\mathcal{L}(\boldsymbol{w}_0)}}{1-\sqrt{1-\frac{1}{2}\cdot\eta\delta\mu}}$, and conclude

that $\delta-$Regularized-GClip converges to a zero-loss solution within $\mathrm{B}(\boldsymbol{w}_0, R)$ at a convergence rate of $\mathcal{L}(\boldsymbol{w}_t) \leq \mathcal{L}(\boldsymbol{w}_0)(1-\eta\delta\mu)^t$. ∎

4.1. Proofs of the Lemmas .

Proof. **(of Lemma 2.10)** We shall prove the theorem by induction and our hypothesis is that, upto step t, $\boldsymbol{w}_t \in B(\boldsymbol{w}_0, R)$ for the given R, $\mathcal{L}(\boldsymbol{w}_t) \leq \mathcal{L}(\boldsymbol{w}_0)(1-\frac{1}{2}\cdot a\mu)^t$ and thus up to t the algorithm explored a region where the $\mu-$PL* condition holds. The base case is trivial, when $t=0$ then $\boldsymbol{w}_0 \in B(\boldsymbol{w}_0, R)$. Now we set out to prove that these continue to hold at $t+1$ too.

From the assumptions that, $\mathcal{L}$ is β-smooth, we have,

$$\mathcal{L}(\boldsymbol{w}_{t+1}) - \mathcal{L}(\boldsymbol{w}_t) - \nabla\mathcal{L}(\boldsymbol{w}_t)^\top(\boldsymbol{w}_{t+1}-\boldsymbol{w}_t) \leq \frac{\beta}{2}\|\boldsymbol{w}_{t+1}-\boldsymbol{w}_t\|^2. \tag{4.2}$$

As $h(\boldsymbol{w}_t) < \min\{\frac{1}{\beta}, \frac{1}{\mu}\}$, we have that $\frac{1}{h(\boldsymbol{w}_t)} > \beta$, hence we relax the above inequality to,

$$\mathcal{L}(\boldsymbol{w}_{t+1}) - \mathcal{L}(\boldsymbol{w}_t) - \nabla\mathcal{L}(\boldsymbol{w}_t)^\top(\boldsymbol{w}_{t+1}-\boldsymbol{w}_t) \leq \frac{1}{2h(\boldsymbol{w}_t)}\|\boldsymbol{w}_{t+1}-\boldsymbol{w}_t\|^2 \tag{4.3}$$

Using the definition of the algorithm, that $\boldsymbol{w}_{t+1}-\boldsymbol{w}_t = -h(\boldsymbol{w}_t)\nabla\mathcal{L}(\boldsymbol{w}_t)$ and we can rearrange the above to get,

$$\mathcal{L}(\boldsymbol{w}_{t+1}) - \mathcal{L}(\boldsymbol{w}_t) \leq -\frac{h(\boldsymbol{w}_t)}{2}\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2 \tag{4.4}$$

Further, we can use the induction hypothesis for the μ-PL* condition at the current iterate, $\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2 \geq \mu\mathcal{L}(\boldsymbol{w}_t)$, to get,

$$\mathcal{L}(\boldsymbol{w}_{t+1}) - \mathcal{L}(\boldsymbol{w}_t) \leq -\frac{h(\boldsymbol{w}_t)}{2}\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2 \leq -\frac{h(\boldsymbol{w}_t)\mu}{2}\mathcal{L}(\boldsymbol{w}_t) \tag{4.5}$$

And the above can be rearranged to,

$$\mathcal{L}(\boldsymbol{w}_{t+1}) \leq (1-\frac{1}{2}\cdot h(\boldsymbol{w}_t)\mu)\mathcal{L}(\boldsymbol{w}_t) \tag{4.6}$$

Note that for the convergence rate to hold, $h(\boldsymbol{w}_t)$ must be bounded such that $\forall t$, $(1-\frac{1}{2}\cdot h(\boldsymbol{w}_t)\mu)$ is always positive and less than 1, both of which follow from the bounds on a, b. We then unroll the recursion to get,

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}_{t+1}) &\leq \mathcal{L}(\boldsymbol{w}_0)\cdot\prod_{i=0}^{t}(1-\frac{h(\boldsymbol{w}_i)\mu}{2}) \\ &\leq \mathcal{L}(\boldsymbol{w}_0)(1-\frac{1}{2}\cdot a\mu)^{t+1}\end{aligned} \tag{4.7}$$

where the last inequality comes from $0 < a \leq h(\boldsymbol{w}_t)$. Therefore assuming that the convergence rate holds till time t implies that it also holds till $t+1$.

Next we embark on proving that $\boldsymbol{w}_{t+1} \in B(\boldsymbol{w}_0, R)$. From the algorithm's update equation, the triangle inequality and recalling that $h(\boldsymbol{w}_t) \leq b$ we get

$$\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_0\| \leq \sum_{i=0}^{t} \|h(\boldsymbol{w}_i) \cdot \nabla \mathcal{L}(\boldsymbol{w}_i)\| \leq b \sum_{i=0}^{t} \|\nabla \mathcal{L}(\boldsymbol{w}_i)\| \tag{4.8}$$

We can rearrange the β-smoothness inequality from equation 4.2 and apply Cauchy-Schwarz,

$$0 \leq \frac{\beta}{2}\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_t\|^2 + \|\nabla \mathcal{L}(\boldsymbol{w}_t)\|\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_t\| + \mathcal{L}(\boldsymbol{w}_t) - \mathcal{L}(\boldsymbol{w}_{t+1}) \tag{4.9}$$

We can relax the above inequality dropping the $\mathcal{L}(\boldsymbol{w}_{t+1})$ term and treat the above as a quadratic in $\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_t\|$ and conclude that the inequality only holds if the discriminant is non-positive, $\|\nabla \mathcal{L}(\boldsymbol{w}_t)\| \leq \sqrt{2\beta \mathcal{L}(\boldsymbol{w}_t)}$. Substituting this inequality into equation 4.8 we get,

$$\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_0\| \leq b \sum_{i=0}^{t} \sqrt{2\beta \mathcal{L}(\boldsymbol{w}_i)} \tag{4.10}$$

Using the assumed convergence rate till the current iterate we get,

$$\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_0\| \leq b\sqrt{2\beta}\sqrt{\mathcal{L}(\boldsymbol{w}_0)} \cdot \left(\sum_{i=0}^{t} \prod_{j=0}^{i} (1 - \frac{1}{2} \cdot h(\boldsymbol{w}_j)\mu)^{1/2} \right) \tag{4.11}$$

Since $a \leq h(\boldsymbol{w}_t) < 1/\mu$, we have, $0 < 1 - \frac{1}{2} \cdot h(\boldsymbol{w}_t)\mu < 1 - \frac{1}{2} \cdot a\mu < 1$. Thus we get,

$$\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_0\| \leq b\sqrt{2\beta}\sqrt{\mathcal{L}(\boldsymbol{w}_0)} \cdot \left(\sum_{i=0}^{t} (1 - \frac{1}{2} \cdot a\mu)^{i/2} \right) \tag{4.12}$$

Upper bounding the above by the closed form expression for the infinite geometric series, we get,

$$\|\boldsymbol{w}_{t+1} - \boldsymbol{w}_0\| \leq \frac{b\sqrt{2\beta}\sqrt{\mathcal{L}(\boldsymbol{w}_0)}}{1 - \sqrt{1 - \frac{1}{2} \cdot a\mu}} \leq R \tag{4.13}$$

The last inequality follows by the definition of R and hence we have proven that $\boldsymbol{w}_{t+1} \in B(\boldsymbol{w}_0, R)$ - and hence up to time $t+1$ the algorithm is still exploring the region within which the μ–PL* condition holds.

Thus induction follows and we have that $\forall t$, $\boldsymbol{w}_t \in B(\boldsymbol{w}_0, R)$ and $\mathcal{L}(\boldsymbol{w}_t) \leq \mathcal{L}(\boldsymbol{w}_0)(1 - \frac{1}{2} \cdot a\mu)^t$. ■

Proof of δ–Regularized-GClip Having a Bounded Step Size.

Proof. **(of Lemma 2.11)** Utilising δ–Regularized-GClip's definition for h, we get that if $\|\nabla \mathcal{L}(\boldsymbol{w}_t)\| \geq \gamma/\delta$, then, $h(\boldsymbol{w}_t) = \min\{\eta, \eta\delta\}$ Otherwise, if $\|\nabla \mathcal{L}(\boldsymbol{w}_t)\| < \gamma/\delta$ then, $h(\boldsymbol{w}_t) = \min\left\{\eta, \frac{\eta\gamma}{\|\nabla \mathcal{L}(\boldsymbol{w}_t)\|}\right\}$ The smallest possible h for the above would be if $\|\nabla \mathcal{L}(\boldsymbol{w}_t)\|$ was as large as it could be, which would result in $h(\boldsymbol{w}_t) = \min\{\eta, \eta\delta\}$. As $\delta < 1$, we conclude $0 < \eta\delta \leq h(\boldsymbol{w}_t) \leq \eta$. ■

*Proof of δ–***Regularized-GClip** *Convergence on Smooth* PL* *Functions.*

Proof. **(of Lemma 2.12)** From Lemma 2.11, we know that δ–Regularized-GClip satisfies the condition $0 < \eta\delta \leq h(\boldsymbol{w}_t) \leq \eta$. Therefore, by setting $\eta < \min\{\frac{1}{\beta}, \frac{1}{\mu}\}$ and $\delta < 1$, we can apply Lemma 2.10 and obtain the convergence rate,

$$\mathcal{L}(\boldsymbol{w}_t) \leq \mathcal{L}(\boldsymbol{w}_0)(1 - \frac{1}{2} \cdot \eta\delta\mu)^t, \tag{4.14}$$

as well as that the PL* condition must hold within a ball $\mathrm{B}(\boldsymbol{w}_0, \mathrm{R})$ where,
$R \geq \dfrac{\eta\sqrt{2\beta}\sqrt{\mathcal{L}(\boldsymbol{w}_0)}}{1 - \sqrt{1 - \frac{1}{2} \cdot \eta\delta\mu}}$. ∎

REFERENCES

[ACG+16] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, oct 2016.

[BAP+17] Parnia Bahar, Tamer Alkhouli, Jan-Thorsten Peter, Christopher Jan-Steffen Brix, and Hermann Ney. Empirical investigation of optimization algorithms in neural machine translation. *The Prague Bulletin of Mathematical Linguistics*, 108(1):13–25, 2017.

[BDSS21] Andrew Brock, Soham De, Samuel L. Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. *CoRR*, abs/2102.06171, 2021.

[BMR+20] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[BWAA18] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd: compressed optimisation for non-convex problems. *arXiv preprint arXiv:1802.04434*, 2018.

[CG18] Jinghui Chen and Quanquan Gu. Closing the generalization gap of adaptive gradient methods in training deep neural networks. *arXiv preprint arXiv:1806.06763*, 2018.

[CLSH18] Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. *arXiv preprint arXiv:1808.02941*, 2018.

[DHS11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[DL22] Zehao Dou and Yuanzhi Li. On the one-sided convergence of adam-type algorithms in non-convex non-concave min-max optimization, 2022.

[DMU18] Soham De, Anirbit Mukherjee, and Enayat Ullah. Convergence guarantees for rmsprop and adam in non-convex optimization and an empirical comparison to nesterov acceleration, 2018.

[FGQ97] Haitao Fang, Guanglu Gong, and Minping Qian. Annealing of iterative stochastic schemes. *SIAM journal on control and optimization*, 35(6):1886–1907, 1997.

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[JGH18] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *CoRR*, abs/1806.07572, 2018.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arxiv. org, 2014.

[Kri09] Alex Krizhevsky. Learning multiple layers of features from tiny images. Masters thesis, University of Toronto, Toronto, Canada, 2009.

[LLH+23] Hong Liu, Zhiyuan Li, David Leo Wright Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *The Twelfth International Conference on Learning Representations*, 2023.

[LZB20] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. February 2020.

[MC01] John L Maryak and Daniel C Chin. Global random optimization by simultaneous perturbation stochastic approximation. In *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, volume 2, pages 756–762. IEEE, 2001.

[MDB17] Gábor Melis, Chris Dyer, and Phil Blunsom. On the state of the art of evaluation in neural language models. *arXiv preprint arXiv:1707.05589*, 2017.

[MKH23] Chunmei Ma, Xiangshan Kong, and Baogui Huang. Image classification based on layered gradient clipping under differential privacy. *IEEE Access*, 11:20150–20158, 2023.

[MKS17] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing LSTM language models. *CoRR*, abs/1708.02182, 2017.

[PMB12] Razvan Pascanu, Tomás Mikolov, and Yoshua Bengio. Understanding the exploding gradient problem. *CoRR*, abs/1211.5063, 2012.

[RKK18] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[SRK+19] Matthew Staib, Sashank J Reddi, Satyen Kale, Sanjiv Kumar, and Suvrit Sra. Escaping saddle points with adaptive gradient methods. *arXiv preprint arXiv:1901.09149*, 2019.

[SS19] S Sun and J.C Spall. Spsa method using diagonalized hessian estimate. *Proceedings of the IEEE Conference on Decision and Control*, page 4922–4927, 2019.

[SSG19] U. Simsekli, L. Sagun, and M. Gurbuzbalaban. A tail-index analysis of stochastic gradient noise in deep neural networks. In *Proceedings of the 36th International Conference on Machine Learning, (ICML) 2019*, 2019.

[SZ13] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 71–79, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[WDW19] Xiaoxia Wu, Simon S. Du, and Rachel Ward. Global convergence of adaptive gradient methods for an over-parameterized neural network. *CoRR*, abs/1902.07111, 2019.

[WWB19] Rachel Ward, Xiaoxia Wu, and Leon Bottou. AdaGrad stepsizes: Sharp convergence over non-convex landscapes. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6677–6686, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[ZCLG21] Difan Zou, Yuan Cao, Yuanzhi Li, and Quanquan Gu. Understanding the generalization of adam in learning neural networks with proper regularization. *CoRR*, abs/2108.11371, 2021.

[ZHSJ19] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity, 2019.

[ZKV+20] Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank J Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models?, 2020.

[ZSJ+19] Fangyu Zou, Li Shen, Zequn Jie, Weizhong Zhang, and Wei Liu. A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Appendix A. A Proof of Convergence for Stochastic δ–Regularized-GClip.

We start by proving a more general result as follows,

Theorem A.1. *Given Assumptions 2.6, 2.7 and 2.8, and for an arbitrary choice of* $\epsilon > 0$, *consider* $1 > \delta > \frac{\left(1+\left(\frac{\epsilon}{\theta}\right)^2\right)}{\left(1+3\left(\frac{\epsilon}{\theta}\right)^2\right)}$ *and* $0 < \eta < \frac{\delta\left(1+3\left(\frac{\epsilon}{\theta}\right)^2\right)-\left(1+\left(\frac{\epsilon}{\theta}\right)^2\right)}{2\beta\left(1+\left(\frac{\epsilon}{\theta}\right)^2\right)}$, *stochastic* δ–*Regularized-GClip satisfies the following inequality over any* $T > 0$ *iterations,*

$$\min_{t=1,\ldots,T} \mathbb{E}\left[\|\nabla \mathcal{L}(\boldsymbol{w}_t)\|^2\right] \leq \epsilon^2 + \frac{1}{T} \cdot \frac{\mathcal{L}(\boldsymbol{w}_1)}{\left(\frac{\eta}{2}(3\delta - 1) - \beta\eta^2\right)}$$

It's clear from above that we can choose any $\epsilon > 0$ howsoever small and $T > 0$ howsoever large and have the minimum value over iterates of the expected gradient norm be similarly small. To prove Theorem A.1 we need the following two lemmas.

Lemma A.2.

$$\mathbb{E}\left[h(\boldsymbol{g}_t)^2\langle \boldsymbol{g}_t - \nabla \mathcal{L}(\boldsymbol{w}_t), \nabla \mathcal{L}(\boldsymbol{w}_t)\rangle \mid \boldsymbol{w}_t\right] \leq \eta^2\theta\|\nabla \mathcal{L}(\boldsymbol{w}_t)\|. \tag{A.1}$$

Proof. We begin by employing Cauchy-Schwarz and Assumption 2.6 to get,

$$\begin{aligned}\mathbb{E}\left[h(\boldsymbol{g}_t)^2\langle \boldsymbol{g}_t - \nabla \mathcal{L}(\boldsymbol{w}_t), \nabla \mathcal{L}(\boldsymbol{w}_t)\rangle \mid \boldsymbol{w}_t\right] &\leq \mathbb{E}\left[h(\boldsymbol{g}_t)^2\|\boldsymbol{g}_t - \nabla \mathcal{L}(\boldsymbol{w}_t)\| \mid \boldsymbol{w}_t\right]\|\nabla \mathcal{L}(\boldsymbol{w}_t)\| \\ &\leq \mathbb{E}\left[h(\boldsymbol{g}_t)^2 \mid \boldsymbol{w}_t\right]\|\nabla \mathcal{L}(\boldsymbol{w}_t)\|\theta \\ &\leq \eta^2\|\nabla \mathcal{L}(\boldsymbol{w}_t)\|\theta \qquad \text{(A.2)}\end{aligned}$$

where in the last inequality we invoked the fact that $h(\boldsymbol{g}_t) \leq \eta$. ∎

Lemma A.3.

$$\mathbb{E}\left[(-h(\boldsymbol{g}_t))\langle \boldsymbol{g}_t - \nabla \mathcal{L}(\boldsymbol{w}_t), \nabla \mathcal{L}(\boldsymbol{w}_t)\rangle \mid \boldsymbol{w}_t\right] \leq (\eta - \eta\delta)\cdot\theta\cdot\|\nabla \mathcal{L}(\boldsymbol{w}_t)\| \tag{A.3}$$

Proof. Because of $\boldsymbol{g}_t$ being an unbiased gradient estimate we have,

$$\mathbb{E}[(-h(\boldsymbol{g}_t))\cdot\langle \boldsymbol{g}_t - \nabla \mathcal{L}(\boldsymbol{w}_t), \nabla \mathcal{L}(\boldsymbol{w}_t)\rangle|\boldsymbol{w}_t] = \mathbb{E}[(\eta - h(\boldsymbol{g}_t))\cdot\langle \boldsymbol{g}_t - \nabla \mathcal{L}(\boldsymbol{w}_t), \nabla \mathcal{L}(\boldsymbol{w}_t)\rangle|\boldsymbol{w}_t] \tag{A.4}$$

Noting that $0 \leq \eta - h(\boldsymbol{g}_t) \leq \eta - \eta\delta$, we get,

$$\mathbb{E}[(-h(\boldsymbol{g}_t))\cdot\langle \boldsymbol{g}_t - \nabla \mathcal{L}(\boldsymbol{w}_t), \nabla \mathcal{L}(\boldsymbol{w}_t)\rangle|\boldsymbol{w}_t] \leq (\eta - \eta\delta)\cdot\theta\cdot\|\nabla \mathcal{L}(\boldsymbol{w}_t)\| \tag{A.5}$$

∎

A.1. Proof of Theorem A.1.

Proof. We parameterize the line from $\boldsymbol{w}_t$ to $\boldsymbol{w}_{t+1}$ as $\kappa(t) = t\boldsymbol{w}_t + (1-t)\boldsymbol{w}_{t+1}$ and applying the Taylor's expansion and then Cauchy-Schwartz formula for the loss evaluated at its end-point we get,

$$\begin{aligned}
&\mathbb{E}[\mathcal{L}(\boldsymbol{w}_{t+1}) \mid \boldsymbol{w}_t] \\
&\leq \mathbb{E}\left[\mathcal{L}(\boldsymbol{w}_t) - h(\boldsymbol{g}_t)\langle \boldsymbol{g}_t, \nabla\mathcal{L}(\boldsymbol{w}_t)\rangle + \frac{1}{2}\int_0^1 (\boldsymbol{w}_{t+1}-\boldsymbol{w}_t)^\top \nabla^2\mathcal{L}(\kappa(s))(\boldsymbol{w}_{t+1}-\boldsymbol{w}_t)\, ds \mid \boldsymbol{w}_t\right] \\
&\leq \mathcal{L}(\boldsymbol{w}_t) - \mathbb{E}[h(\boldsymbol{g}_t)\langle \boldsymbol{g}_t, \nabla\mathcal{L}(\boldsymbol{w}_t)\rangle \mid \boldsymbol{w}_t] \\
&+ \frac{\mathbb{E}[\|\boldsymbol{w}_{t+1}-\boldsymbol{w}_t\|^2 \mid \boldsymbol{w}_t]}{2}\int_0^1 \left\|\nabla^2\mathcal{L}(\kappa(s))\right\| ds
\end{aligned}$$

Invoking $\|\boldsymbol{w}_{t+1}-\boldsymbol{w}_t\| = h(\boldsymbol{g}_t)\|\boldsymbol{g}_t\|$ and $\left\|\nabla^2\mathcal{L}(\kappa(s))\right\| \leq \beta$ we have,

$$\mathbb{E}[\mathcal{L}(\boldsymbol{w}_{t+1}) \mid \boldsymbol{w}_t] \leq \mathcal{L}(\boldsymbol{w}_t) - \mathbb{E}[h(\boldsymbol{g}_t)\langle \boldsymbol{g}_t, \nabla\mathcal{L}(\boldsymbol{w}_t)\rangle \mid \boldsymbol{w}_t] + \frac{\beta}{2}\mathbb{E}\left[h(\boldsymbol{g}_t)^2\|\boldsymbol{g}_t\|^2 \mid \boldsymbol{w}_t\right] \tag{A.6}$$

Substituting $\nabla\mathcal{L}(\boldsymbol{w}_t) + \boldsymbol{g}_t - \nabla\mathcal{L}(\boldsymbol{w}_t)$ for $\boldsymbol{g}_t$ in the second and the third term above, we get,

$$\begin{aligned}
\mathbb{E}[\mathcal{L}(\boldsymbol{w}_{t+1}) \mid \boldsymbol{w}_t] \leq \mathcal{L}(\boldsymbol{w}_t) &\\
- \mathbb{E}[h(\boldsymbol{g}_t)\langle \boldsymbol{g}_t - \nabla\mathcal{L}(\boldsymbol{w}_t), \nabla\mathcal{L}(\boldsymbol{w}_t)\rangle \mid \boldsymbol{w}_t] - \mathbb{E}[h(\boldsymbol{g}_t) \mid \boldsymbol{w}_t]\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2 &\\
+ \frac{\beta}{2}\mathbb{E}\left[h(\boldsymbol{g}_t)^2\left(\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2 + \|\boldsymbol{g}_t - \nabla\mathcal{L}(\boldsymbol{w}_t)\|^2 + 2\langle \nabla\mathcal{L}(\boldsymbol{w}_t), \boldsymbol{g}_t - \nabla\mathcal{L}(\boldsymbol{w}_t)\rangle\right) \mid \boldsymbol{w}_t\right] &
\end{aligned} \tag{A.7}$$

Recalling that $\eta\delta \leq h(\boldsymbol{g}_t) \leq \eta$ and given that $\delta \in (0,1)$ we get,

$$\begin{aligned}
\mathbb{E}[\mathcal{L}(\boldsymbol{w}_{t+1}) \mid \boldsymbol{w}_t] \leq \mathcal{L}(\boldsymbol{w}_t) &\\
- \mathbb{E}[h(\boldsymbol{g}_t)\langle \boldsymbol{g}_t - \nabla\mathcal{L}(\boldsymbol{w}_t), \nabla\mathcal{L}(\boldsymbol{w}_t)\rangle \mid \boldsymbol{w}_t] - \eta\delta\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2 &\\
+ \frac{\beta\eta^2}{2}\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2 + \frac{\beta\eta^2\theta^2}{2} &\\
+ \beta\mathbb{E}\left[h(\boldsymbol{g}_t)^2\langle \boldsymbol{g}_t - \nabla\mathcal{L}(\boldsymbol{w}_t), \nabla\mathcal{L}(\boldsymbol{w}_t)\rangle \mid \boldsymbol{w}_t\right] &
\end{aligned} \tag{A.8}$$

Now we invoke Lemma A.3 on the second term above and Lemma A.2 on the last term of the RHS above and take total expectations to get,

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\boldsymbol{w}_{t+1})] \leq \mathbb{E}[\mathcal{L}(\boldsymbol{w}_t)] + \left\{\eta(1-\delta)\theta + \beta\eta^2\theta\right\}\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|] \\ - \left(\eta\delta - \frac{\beta\eta^2}{2}\right)\mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2\right] \\ + \frac{\beta\eta^2\theta^2}{2}\end{aligned} \tag{A.9}$$

Given a $T \in \mathbb{Z}^+$ and summing the above over all $t = 1, \ldots, T$ and recalling that $\boldsymbol{w}_1$ is an arbitrary non-random initialization, we get,

$$\begin{aligned}\left(\eta\delta - \frac{\beta\eta^2}{2}\right)\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2\right] \leq & \mathcal{L}(\boldsymbol{w}_1) - \mathbb{E}[\mathcal{L}(\boldsymbol{w}_{T+1})] \\ & + \left\{\eta(1-\delta)\theta + \beta\eta^2\theta\right\}\sum_{t=1}^{T}\mathbb{E}[\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|] + \frac{\beta\eta^2\theta^2}{2}T\end{aligned} \tag{A.10}$$

Invoking the inequality, $\theta \cdot \|\nabla\mathcal{L}(\boldsymbol{w}_t)\| \leq \frac{1}{2}\cdot(\theta^2 + \|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2)$ and that $\mathcal{L} \geq 0$ we get,

$$\begin{aligned}&\left(\eta\delta - \frac{\beta\eta^2}{2}\right)\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2\right] \\ &\leq \mathcal{L}(\boldsymbol{w}_1) + \left\{\eta(1-\delta) + \beta\eta^2\right\}\sum_{t=1}^{T}\mathbb{E}[\frac{1}{2}\cdot\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2] \\ &+ \left(\frac{\beta\eta^2 + \eta(1-\delta) + \beta\eta^2}{2}\right)\theta^2 T\end{aligned} \tag{A.11}$$

The above implies,

$$\left(\eta\delta - \frac{\beta\eta^2}{2} - \frac{\eta(1-\delta) + \beta\eta^2}{2}\right)\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2\right] \leq \mathcal{L}(\boldsymbol{w}_1) + \left(\frac{2\beta\eta^2 + \eta(1-\delta)}{2}\right)\theta^2 T$$

Invoking the assumption that $\delta > \frac{\left(1+\left(\frac{\epsilon}{\theta}\right)^2\right)}{\left(1+3\left(\frac{\epsilon}{\theta}\right)^2\right)} > \frac{1}{3}$ and $\eta < \frac{\delta\left(1+3\left(\frac{\epsilon}{\theta}\right)^2\right)-\left(1+\left(\frac{\epsilon}{\theta}\right)^2\right)}{2\beta\left(1+\left(\frac{\epsilon}{\theta}\right)^2\right)} < \frac{3\delta-1}{2\beta}$ we get,

$$\begin{aligned}\min_{t=1,\ldots,T}\mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2\right] &\leq \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla\mathcal{L}(\boldsymbol{w}_t)\|^2\right] \\ &\leq \frac{\mathcal{L}(\boldsymbol{w}_1)}{T\cdot\left(\frac{\eta}{2}(3\delta-1) - \beta\eta^2\right)} + \left(\frac{2\beta\eta^2 + \eta(1-\delta)}{2\cdot\left(\frac{\eta}{2}(3\delta-1) - \beta\eta^2\right)}\right)\theta^2\end{aligned} \tag{A.12}$$

Now for an arbitrary $\epsilon > 0$. we can solve the inequation,

$$\frac{\eta(1-\delta)+2\beta\eta^2}{\eta(3\delta-1)-2\beta\eta^2} < \left(\frac{\epsilon}{\theta}\right)^2 \implies \eta \in \left(0, \frac{\delta\left(1+3\left(\frac{\epsilon}{\theta}\right)^2\right)-\left(1+\left(\frac{\epsilon}{\theta}\right)^2\right)}{2\beta(1+\left(\frac{\epsilon}{\theta}\right)^2)}\right)$$

Note that the above upperbound on η is the range of η chosen in the statement. And thus we get the desired theorem statement. ∎

A.2. Proof of Theorem 2.9.

Proof. Substituting the given choices of η, δ and β we get,

$$\frac{1}{\eta\cdot\left(\frac{3\delta-1}{2}-\beta\eta\right)} = \frac{16(1+\epsilon'^2)^2(1+3\epsilon'^2)}{\epsilon'^2(3\epsilon'^4+9\epsilon'^2+4)} = \frac{4}{\epsilon'^2}+11+\frac{\epsilon'^2}{4}+\frac{51\epsilon'^4}{16}+\mathcal{O}(\epsilon'^6)$$

Substituting the above into the guarantee of Theorem A.1 along with $T = \frac{1}{\epsilon'^4}$ we get the result claimed. ∎

Appendix B. ResNet-18 on CIFAR-10 Without Weight Decay. For completeness, in Figure B.1 we present a version of the experiments ran in Section 2.2, but without weight-decay for any of the algorithms considered.

The performance of the gradient clipping based algorithms, as well as Adam, do not show significant changes with the removal of weight decay, however, SGD performs significantly worse. In summary, the discussion around the effectiveness of $\delta-$Regularized-GClip still stands as given in the main text.

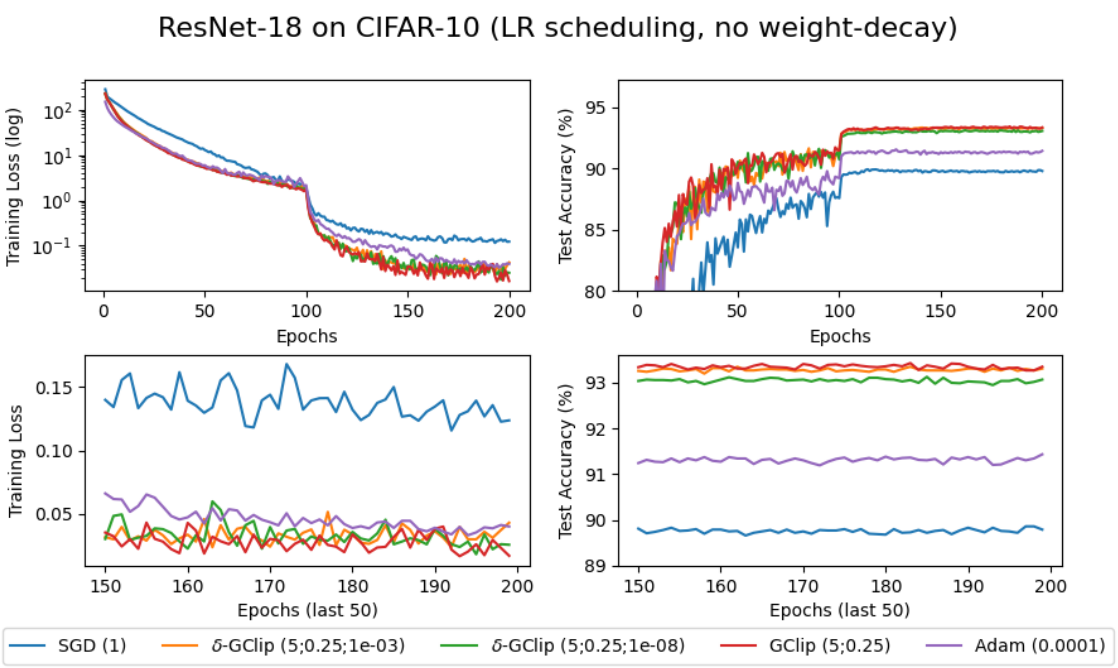

Figure B.1: $\delta-$Regularized-GClip ($\delta-$GCLip) matches the best heuristics for training a ResNet-18 on CIFAR-10 with learning-rate scheduling, but no weight-decay