Appendix: A Generic Approach for Escaping Saddle points

A Proof of Theorem 1

The case of $\tau = \emptyset$ can be handled in a straightforward manner, so let us focus on the case where $\tau = \diamond$. We split our analysis into cases, each analyzing the change in objective function value depending on second-order criticality of y^t.

We start with the case where the gradient condition of second-order critical point is violated and then proceed to the case where the Hessian condition is violated.

Case I: $\mathbb{E}[\|\nabla f(y^t)\|] \geq \epsilon$ for some $t > 0$

We first observe the following: $\mathbb{E}[\|\nabla f(y^t)\|^2] \geq (\mathbb{E}\|\nabla f(y^t)\|)^2 \geq \epsilon^2$. This follows from a straightforward application of Jensen's inequality. From this inequality, we have the following:

$$\epsilon^2 \leq \mathbb{E}[\|\nabla f(y^t)\|^2] \leq \frac{1}{g(n,\epsilon)}\mathbb{E}[f(x^{t-1}) - f(z^t)]. \tag{5}$$

This follows from the fact that y^t is the output of Gradient-Focused-Optimizer subroutine, which satisfies the condition that for $(y, z) =$ Gradient-Focused-Optimizer(x, n, ϵ), we have

$$\mathbb{E}[\|\nabla f(y)\|^2] \leq \frac{1}{g(n,\epsilon)}\mathbb{E}[f(x) - f(z)].$$

From Equation (5), we have

$$\mathbb{E}[f(z^t)] \leq \mathbb{E}[f(x^{t-1})] - \epsilon^2 g(n, \epsilon).$$

Furthermore, due to the property of non-increasing nature of Gradient-Focused-Optimizer, we also have $\mathbb{E}[y^t] \leq \mathbb{E}[f(x^{t-1})]$.

We now focus on the Hessian-Focused-Optimizer subroutine. From the property of Hessian-Focused-Optimizer that the objective function value is non-increasing, we have $\mathbb{E}[f(x^t)] \leq \mathbb{E}[f(u^t)]$. Therefore, combining with the above inequality, we have

$$\begin{aligned}
\mathbb{E}[f(x^t)] &\leq \mathbb{E}[f(u^t)] \\
&= p\mathbb{E}[f(y^t)] + (1-p)\mathbb{E}[f(z^t)] \\
&\leq p\mathbb{E}[f(x^{t-1})] + (1-p)(\mathbb{E}[f(x^{t-1})] - \epsilon^2 g(n,\epsilon)) \\
&= \mathbb{E}[f(x^{t-1})] - (1-p)\epsilon^2 g(n,\epsilon).
\end{aligned} \tag{6}$$

The first equality is due to the definition of u^t in Algorithm 1. Therefore, when the gradient condition is violated, irrespective of whether $\lambda_{\min}(\nabla^2 f(x)) \leq -\gamma$ or $\nabla^2 f(y^t) \succeq -\gamma\mathbb{I}$, the objective function value always decreases by at least $\epsilon^2 g(n, \epsilon)$.

Case II: $\mathbb{E}[\|\nabla f(y^t)\|] < \epsilon$ and $\lambda_{\min}(\nabla^2 f(x)) \leq -\gamma$ for some $t > 0$

In this case, we first note that for $y =$ Hessian-Focused-Optimizer(x, n, ϵ, γ) and $\lambda_{\min}(\nabla^2 f(x)) \leq -\gamma$, we have $\mathbb{E}[f(y)] \leq f(x) - h(n, \epsilon, \gamma)$. Observe that $x^t =$ Hessian-Focused-Optimizer$(u^t, n, \epsilon, \gamma)$. Therefore, if $u^t = y^t$ and $\lambda_{\min}(\nabla^2 f(x)) \leq -\gamma$, then we have

$$\mathbb{E}[f(x^t) | u^t = y^t] \leq f(y^t) - h(n, \epsilon, \gamma) \leq f(x^{t-1}) - h(n, \epsilon, \gamma).$$

The second inequality is due to the non-increasing property of Gradient-Focused-Optimizer. On the other hand, if $u^t = z^t$, we have hand, if we have $\mathbb{E}[f(x^t) | u^t = z^t] \leq f(z^t)$. This is due to the non-increasing property of Hessian-Focused-Optimizer. Combining the above two inequalities and using the law of total expectation, we get

$$\begin{aligned}
\mathbb{E}[f(x^t)] &= p\mathbb{E}[f(x^t) | u^t = y^t] + (1-p)\mathbb{E}[f(x^t) | u^t = z^t] \\
&\leq p\left(\mathbb{E}[f(y^t)] - h(n,\epsilon,\gamma)\right) + (1-p)\mathbb{E}[f(z^t)] \\
&\leq p\left(\mathbb{E}[f(x^{t-1})] - h(n,\epsilon,\gamma)\right) + (1-p)\mathbb{E}[f(x^{t-1})] \\
&= \mathbb{E}[f(x^{t-1})] - ph(n,\epsilon,\gamma).
\end{aligned} \tag{7}$$

The second inequality is due to he non-increasing property of Gradient-Focused-Optimizer. Therefore, when the hessian condition is violated, the objective function value always decreases by at least $ph(n, \epsilon, \gamma)$.

Case III: $\mathbb{E}[\|\nabla f(y^t)\|] < \epsilon$ and $\nabla^2 f(y^t) \succeq -\gamma \mathbb{I}$ for some $t > 0$

This is the favorable case for the algorithm. The only condition to note is that the objective function value will be non-increasing in this case too. This is, again, due to the non-increasing properties of subroutines Gradient-Focused-Optimizer and Hessian-Focused-Optimizer. In general, greater the occurrence of this case during the course of the algorithm, higher will the probability that the output of our algorithm satisfies the desired property.

The key observation is that Case I & II cannot occur large number of times since each of these cases strictly decreases the objective function value. In particular, from Equation (6) and (7), it is easy to see that each occurrence of Case I & II the following holds:

$$\mathbb{E}[f(x^t)] \leq \mathbb{E}[f(x^{t-1})] - \theta,$$

where $\theta = \min((1-p)\epsilon^2 g(n, \epsilon), ph(n, \epsilon, \gamma))$. Furthermore, the function f is lower bounded by B, thus, Case I & II cannot occur more than $(f(x^0) - B)/\theta$ times. Therefore, the probability of occurrence of Case III is at least $1 - (f(x^0) - B)/(T\theta)$, which completes the first part of the proof.

The second part of the proof simply follows from first part. As seen above, the probability of Case I & II is at most $(f(x^0) - B)/T\theta$. Therefore, probability that an element of the set S falls in Case III is at least $1 - ((f(x^0) - B)/T\theta)^k$, which gives us the required result for the second part.

B Proof of Lemma 1

Proof. The proof follows from the analysis in Reddi et al. [2016a] with some additional reasoning. We need to show two properties: **G.1** and **G.2**, both of which are based on objective function value. To this end, we start with an update in the s^{th} epoch. We have the following:

$$\begin{aligned}\mathbb{E}[f(x_{t+1}^{s+1})] &\leq \mathbb{E}[f(x_t^{s+1}) + \langle \nabla f(x_t^{s+1}), x_{t+1}^{s+1} - x_t^{s+1} \rangle + \tfrac{L}{2}\|x_{t+1}^{s+1} - x_t^{s+1}\|^2] \\ &\leq \mathbb{E}[f(x_t^{s+1}) - \eta_t \|\nabla f(x_t^{s+1})\|^2 + \tfrac{L\eta_t^2}{2}\|v_t^{s+1}\|^2]. \end{aligned} \tag{8}$$

The first inequality is due to L-smoothness of the function f . The second inequality simply follows from the unbiasedness of Svrg update in Algorithm 2. For the analysis of the algorithm, we need the following Lyapunov function:

$$A_t^{s+1} := \mathbb{E}[f(x_t^{s+1}) + \mu_t \|x_t^{s+1} - \tilde{x}^s\|^2].$$

This function is a combination of objective function and the distance of the current iterate from the latest snapshot $\tilde{x}_s$. Note that the term μ_t is introduced only for the analysis and is not part of the algorithm (see Algorithm 2). Here $\{\mu_t\}_{t=0}^m$ is chosen such the following holds:

$$\mu_t = \mu_{t+1}(1 + \eta_t \beta_t + 2\eta_t^2 L^2) + \eta_t^2 L^3,$$

for all $t \in \{0, \cdots, m-1\}$ and $\mu_m = 0$. For bounding the Lypunov function A, we need the following bound on the distance of the current iterate from the latest snapshot:

$$\begin{aligned} &\mathbb{E}[\|x_{t+1}^{s+1} - \tilde{x}^s\|^2] = \mathbb{E}[\|x_{t+1}^{s+1} - x_t^{s+1} + x_t^{s+1} - \tilde{x}^s\|^2] \\ &= \mathbb{E}[\|x_{t+1}^{s+1} - x_t^{s+1}\|^2 + \|x_t^{s+1} - \tilde{x}^s\|^2 + 2\langle x_{t+1}^{s+1} - x_t^{s+1}, x_t^{s+1} - \tilde{x}^s \rangle] \\ &= \mathbb{E}[\eta_t^2 \|v_t^{s+1}\|^2 + \|x_t^{s+1} - \tilde{x}^s\|^2] - 2\eta_t \mathbb{E}[\langle \nabla f(x_t^{s+1}), x_t^{s+1} - \tilde{x}^s \rangle] \\ &\leq \mathbb{E}[\eta_t^2 \|v_t^{s+1}\|^2 + \|x_t^{s+1} - \tilde{x}^s\|^2] + 2\eta_t \mathbb{E}\left[\tfrac{1}{2\beta_t}\|\nabla f(x_t^{s+1})\|^2 + \tfrac{1}{2}\beta_t \|x_t^{s+1} - \tilde{x}^s\|^2\right]. \end{aligned} \tag{9}$$

The second equality is due to the unbiasedness of the update of Svrg. The last inequality follows from a simple application of Cauchy-Schwarz and Young's inequality. Substituting Equation (8) and Equation (9) into the

Lypunov function A_{t+1}^{s+1}, we obtain the following:

$$\begin{aligned}
A_{t+1}^{s+1} &\leq \mathbb{E}[f(x_t^{s+1}) - \eta_t\|\nabla f(x_t^{s+1})\|^2 + \tfrac{L\eta_t^2}{2}\|v_t^{s+1}\|^2] \\
&\quad + \mathbb{E}[\mu_{t+1}\eta_t^2\|v_t^{s+1}\|^2 + \mu_{t+1}\|x_t^{s+1} - \tilde{x}^s\|^2] \\
&\quad + 2\mu_{t+1}\eta_t\mathbb{E}\left[\tfrac{1}{2\beta_t}\|\nabla f(x_t^{s+1})\|^2 + \tfrac{1}{2}\beta_t\|x_t^{s+1} - \tilde{x}^s\|^2\right] \\
&\leq \mathbb{E}[f(x_t^{s+1}) - \left(\eta_t - \tfrac{\mu_{t+1}\eta_t}{\beta_t}\right)\|\nabla f(x_t^{s+1})\|^2 \\
&\quad + \left(\tfrac{L\eta_t^2}{2} + \mu_{t+1}\eta_t^2\right)\mathbb{E}[\|v_t^{s+1}\|^2] + (\mu_{t+1} + \mu_{t+1}\eta_t\beta_t)\,\mathbb{E}\left[\|x_t^{s+1} - \tilde{x}^s\|^2\right]. \qquad (10)
\end{aligned}$$

To further bound this quantity, we use Lemma 3 to bound $\mathbb{E}[\|v_t^{s+1}\|^2]$, so that upon substituting it in Equation (10), we see that

$$\begin{aligned}
A_{t+1}^{s+1} &\leq \mathbb{E}[f(x_t^{s+1})] - \left(\eta_t - \tfrac{\mu_{t+1}\eta_t}{\beta_t} - \eta_t^2 L - 2\mu_{t+1}\eta_t^2\right)\mathbb{E}[\|\nabla f(x_t^{s+1})\|^2] \\
&\quad + \left[\mu_{t+1}\big(1 + \eta_t\beta_t + 2\eta_t^2L^2\big) + \eta_t^2 L^3\right]\mathbb{E}\left[\|x_t^{s+1} - \tilde{x}^s\|^2\right] \\
&\leq A_t^{s+1} - \big(\eta_t - \tfrac{\mu_{t+1}\eta_t}{\beta_t} - \eta_t^2 L - 2\mu_{t+1}\eta_t^2\big)\mathbb{E}[\|\nabla f(x_t^{s+1})\|^2].
\end{aligned}$$

The second inequality follows from the definition of μ_t and A_t^{s+1}. Since $\eta_t = \eta = 1/(4Ln^{2/3})$ for $j > 0$ and $t \in \{0, \ldots, j-1\}$,

$$A_j^{s+1} \leq A_0^{s+1} - \upsilon_n \sum\nolimits_{t=0}^{j-1} \mathbb{E}[\|\nabla f(x_t^{s+1})\|^2], \qquad (11)$$

where

$$\upsilon_n = \big(\eta_t - \tfrac{\mu_{t+1}\eta_t}{\beta_t} - \eta_t^2 L - 2\mu_{t+1}\eta_t^2\big).$$

We will prove that for the given parameter setting $\upsilon_n > 0$ (see the proof below). With $\upsilon_n > 0$, it is easy to see that $A_j^{s+1} \leq A_0^{s+1}$. Furthermore, note that $A_0^{s+1} = \mathbb{E}[f(x_0^{s+1}) + \mu_0\|x_0^{s+1} - \tilde{x}^s\|^2] = \mathbb{E}[f(x_0^{s+1})]$ since $x_0^{s+1} = \tilde{x}^s$ (see Algorithm 2). Also, we have

$$\mathbb{E}[f(x_j^{s+1}) + \mu_j\|x_j^{s+1} - \tilde{x}^s\|^2] \leq \mathbb{E}[f(x_0^{s+1})]$$

and thus, we obtain $\mathbb{E}[f(x_j^{s+1})] \leq \mathbb{E}[f(x_0^{s+1})]$ for all $j \in \{0, ..., m\}$. Furthermore, using simple induction and the fact that $x_0^{s+1} = x_m^s$ for all epoch $s \in \{0, ..., S-1\}$, it easy to see that $\mathbb{E}[f(x_j^{s+1})] \leq f(x^0)$. Therefore, with the definition of y specified in the output of Algorithm 2, we see that the condition **G.1** of Gradient-Focused-Optimizer is satisfied for Svrg algorithm.

We now prove that $\upsilon_n > 0$ and also **G.2** of Gradient-Focused-Optimizer is satisifed for Svrg algorithm. By using telescoping the sum with $j = m$ in Equation (11), we obtain

$$\sum\nolimits_{t=0}^{m-1} \mathbb{E}[\|\nabla f(x_t^{s+1})\|^2] \leq \frac{A_0^{s+1} - A_m^{s+1}}{\upsilon_n}.$$

This inequality in turn implies that

$$\sum\nolimits_{t=0}^{m-1} \mathbb{E}[\|\nabla f(x_t^{s+1})\|^2] \leq \frac{\mathbb{E}[f(\tilde{x}^s) - f(\tilde{x}^{s+1})]}{\upsilon_n}, \qquad (12)$$

where we used that $A_m^{s+1} = \mathbb{E}[f(x_m^{s+1})] = \mathbb{E}[f(\tilde{x}^{s+1})]$ (since $\mu_m = 0$), and that $A_0^{s+1} = \mathbb{E}[f(\tilde{x}^s)]$ (since $x_0^{s+1} = \tilde{x}^s$). Now sum over all epochs to obtain

$$\frac{1}{T_g}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1} \mathbb{E}[\|\nabla f(x_t^{s+1})\|^2] \leq \frac{\mathbb{E}[f(x^0) - f(x_m^S)]}{T_g\upsilon_n}. \qquad (13)$$

Here we used the the fact that $\tilde{x}^0 = x^0$. To obtain a handle on υ_n and complete our analysis, we will require an upper bound on μ_0. We observe that $\mu_0 = \frac{L}{16n^{4/3}}\frac{(1+\theta)^m - 1}{\theta}$ where $\theta = 2\eta^2L^2 + \eta\beta$. This is obtained using the relation $\mu_t = \mu_{t+1}(1 + \eta\beta + 2\eta^2 L^2) + \eta^2 L^3$ and the fact that $\mu_m = 0$. Using the specified values of β and η we have

$$\theta = 2\eta^2 L^2 + \eta\beta = \frac{1}{8n^{4/3}} + \frac{1}{4n} \leq \frac{3}{4n}.$$

Using the above bound on θ, we get

$$\begin{aligned}\mu_0 &= \frac{L}{16n^{4/3}} \frac{(1+\theta)^m - 1}{\theta} = \frac{L((1+\theta)^m - 1)}{2(1+2n^{1/3})} \\ &\leq \frac{L((1+\frac{3}{4n})^{\lfloor 4n/3 \rfloor} - 1)}{2(1+2n^{1/3})} \leq n^{-1/3}(L(e-1)/4), \end{aligned} \tag{14}$$

wherein the second inequality follows upon noting that $(1+\frac{1}{l})^l$ is increasing for $l > 0$ and $\lim_{l\to\infty}(1+\frac{1}{l})^l = e$ (here e is the Euler's number). Now we can lower bound υ_n, as

$$\upsilon_n = \min_t \big(\eta - \tfrac{\mu_{t+1}\eta}{\beta} - \eta^2 L - 2\mu_{t+1}\eta^2\big) \geq \big(\eta - \tfrac{\mu_0\eta}{\beta} - \eta^2 L - 2\mu_0\eta^2\big) \geq \frac{1}{40Ln^{2/3}}.$$

The first inequality holds since μ_t decreases with t. The second inequality holds since (a) μ_0/β can be upper bounded by $(e-1)/4$ (follows from Equation (14)), (b) $\eta^2 L \leq \eta/4$ and (c) $2\mu_0\eta^2 \leq (e-1)\eta/8$ (follows from Equation (14)). Substituting the above lower bound in Equation (13), we obtain the following:

$$\frac{1}{T_g}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1} \mathbb{E}[\|\nabla f(x_t^{s+1})\|^2] \leq \frac{40Ln^{2/3}\mathbb{E}[f(x^0) - f(x_m^S)]}{T_g}. \tag{15}$$

From the definition of (y, z) in output of Algorithm 2 i.e., y is Iterate x_a chosen uniformly random from $\{\{x_t^{s+1}\}_{t=0}^{m-1}\}_{s=0}^{S-1}$ and $z = x_m^S$, it is clear that Algorithm 2 satisfies the **G.2** requirement of Gradient-Focused-Optimizer with $g(n,\epsilon) = T_\epsilon/40Ln^{2/3}$. Since both **G.1** and **G.2** are satisified for Algorithm 2, we conclude that Svrg is a Gradient-Focused-Optimizer. □

C Proof of Lemma 2

Proof. The first important observation is that the function value never increases because $y = \arg\min_{z\in\{u,x\}} f(z)$ i.e., $f(y) \leq f(x)$, thus satisfying **H.1** of Hessian-Focused-Optimizer. We now analyze the scenario where $\lambda_{min}(\nabla^2 f(x)) \leq -\gamma$. Consider the event where we obtain v such that

$$\langle v, \nabla^2 f(x) v\rangle \leq \lambda_{min}(\nabla^2 f(x)) + \frac{\gamma}{2}.$$

This event (denoted by $\mathcal{E}$) happens with at least probability ρ. Note that, since $\lambda_{min}(\nabla^2 f(x)) \leq -\gamma$, we have $\langle v, \nabla^2 f(x) v\rangle \leq -\frac{\gamma}{2}$. In this case, we have the following relationship:

$$\begin{aligned} f(y) &\leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{1}{2}(y-x)^T \nabla^2 f(x)(y-x) + \frac{M}{6}\|y-x\|^3 \\ &= f(x) - \alpha|\langle \nabla f(x), v\rangle| + \frac{\alpha^2}{2} v^T \nabla^2 f(x) v + \frac{M\alpha^3}{6}\|v\|^3 \\ &\leq f(x) + \frac{\alpha^2}{2} v^T \nabla^2 f(x) v + \frac{M\alpha^3}{6} \\ &\leq f(x) - \frac{1}{2M^2}|v^T\nabla^2 f(x) v|^3 + \frac{1}{6M^2}|v^T\nabla^2 f(x) v|^3 \\ &= f(x) - \frac{1}{3M^2}|v^T\nabla^2 f(x) v|^3 \leq f(x) - \frac{1}{24M^2}\gamma^3. \end{aligned} \tag{16}$$

The first inequality follows from the M-lipschitz continuity of the Hessain $\nabla^2 f(x)$. The first equality follows from the update rule of HessianDescent. The second inequality is obtained by dropping the negative term and using the fact that $\|v\| = 1$. The second equality is obtained by substituting $\alpha = \frac{|v^T\nabla^2 f(x) v|}{M}$. The last inequality is due to the fact that$\langle v, \nabla^2 f(x) v\rangle \leq -\frac{\gamma}{2}$. In the other scenario where

$$\langle v, \nabla^2 f(x) v\rangle \leq \lambda_{min}(\nabla^2 f(x)) + \frac{\gamma}{2},$$

we can at least ensure that $f(y) \leq f(x)$ since $y = \arg\min_{z \in \{u,x\}} f(z)$. Therefore, we have

$$\begin{aligned} \mathbb{E}[f(y)] &= \rho \mathbb{E}[f(y)|\mathcal{E}] + (1-\rho)\mathbb{E}[f(y)|\bar{\mathcal{E}}] \\ &\leq \rho \mathbb{E}[f(y)|\mathcal{E}] + (1-\rho) f(x) \\ &\leq \rho \left[f(x) - \tfrac{\rho}{24M^2}\gamma^3 \right] + (1-\rho) f(x) \\ &= f(x) - \tfrac{\rho}{24M^2}\gamma^3. \end{aligned} \tag{17}$$

The last inequality is due to Equation (16). Hence, Hessian-Focused-Optimizer satisfies **H.2** of Hessian-Focused-Optimizer with $h(n, \epsilon, \gamma) = \frac{\rho}{24M^2}\gamma^3$, thus concluding the proof. □

D Proof of Theorem 3

First note that cubic method is a descent method (refer to Theorem 1 of Nesterov and Polyak [2006]); thus, **H.1** is trivially satisfied. Furthermore, cubic descent is a Hessian-Focused-Optimizer with $h(n, \epsilon, \gamma) = \frac{2\gamma^3}{81M^3}\gamma^3$. This, again, follows from Theorem 1 of Nesterov and Polyak [2006]. The result easily follows from the aforementioned observations.

E Other Lemmas

The following bound on the variance of Svrg is useful for our proof Reddi et al. [2016a].

Lemma 3. *Reddi et al. [2016a] Let v_t^{s+1} be computed by Algorithm 2. Then,*

$$\mathbb{E}[\|v_t^{s+1}\|^2] \leq 2\mathbb{E}[\|\nabla f(x_t^{s+1})\|^2] + 2L^2 \mathbb{E}[\|x_t^{s+1} - \tilde{x}^s\|^2].$$

Proof. We use the definition of v_t^{s+1} to get

$$\begin{aligned} &\mathbb{E}[\|v_t^{s+1}\|^2] = \mathbb{E}[\| \left(\nabla f_{i_t}(x_t^{s+1}) - \nabla f_{i_t}(\tilde{x}^s)\right) + \nabla f(\tilde{x}^s)\|^2] \\ &= \mathbb{E}[\| \left(\nabla f_{i_t}(x_t^{s+1}) - \nabla f_{i_t}(\tilde{x}^s)\right) + \nabla f(\tilde{x}^s) - \nabla f(x_t^{s+1}) + \nabla f(x_t^{s+1})\|^2] \\ &\leq 2\mathbb{E}[\|\nabla f(x_t^{s+1})\|^2] + 2\mathbb{E}\left[\left\|\nabla f_{i_t}(x_t^{s+1}) - \nabla f_{i_t}(\tilde{x}^s) - \mathbb{E}[\nabla f_{i_t}(x_t^{s+1}) - \nabla f_{i_t}(\tilde{x}^s)]\right\|^2\right] \end{aligned}$$

The inequality follows from the simple fact that $(a+b)^2 \leq a^2 + b^2$. From the above inequality, we get the following:

$$\begin{aligned} \mathbb{E}[\|v_t^{s+1}\|^2] &\leq 2\mathbb{E}[\|\nabla f(x_t^{s+1})\|^2] + 2\mathbb{E}\|\nabla f_{i_t}(x_t^{s+1}) - \nabla f_{i_t}(\tilde{x}^s)\|^2 \\ &\leq 2\mathbb{E}[\|\nabla f(x_t^{s+1})\|^2] + 2L^2 \mathbb{E}[\|x_t^{s+1} - \tilde{x}^s\|^2] \end{aligned}$$

The first inequality follows by noting that for a random variable ζ, $\mathbb{E}[\|\zeta - \mathbb{E}[\zeta]\|^2] \leq \mathbb{E}[\|\zeta\|^2]$. The last inequality follows from L-smoothness of f_{i_t}. □

F Approximate Cubic Regularization

Cubic regularization method of [19] is designed to operate on full batch, i.e., it does not exploit the finite-sum structure of the problem and requires the computation of the gradient and the Hessian on the entire dataset to make an update. However, such full-batch methods do not scale gracefully with the size of data and become prohibitively expensive on large datasets. To overcome this challenge, we devised an approximate cubic regularization method described below:

1. Pick a mini-batch $\mathcal{B}$ and obtain the gradient and the hessian based on $\mathcal{B}$, i.e.,

$$g = \frac{1}{|\mathcal{B}|}\sum_{i \in \mathcal{B}} \nabla f_i(x) \qquad H = \frac{1}{|\mathcal{B}|}\sum_{i \in \mathcal{B}} \nabla^2 f_i(x) \tag{18}$$

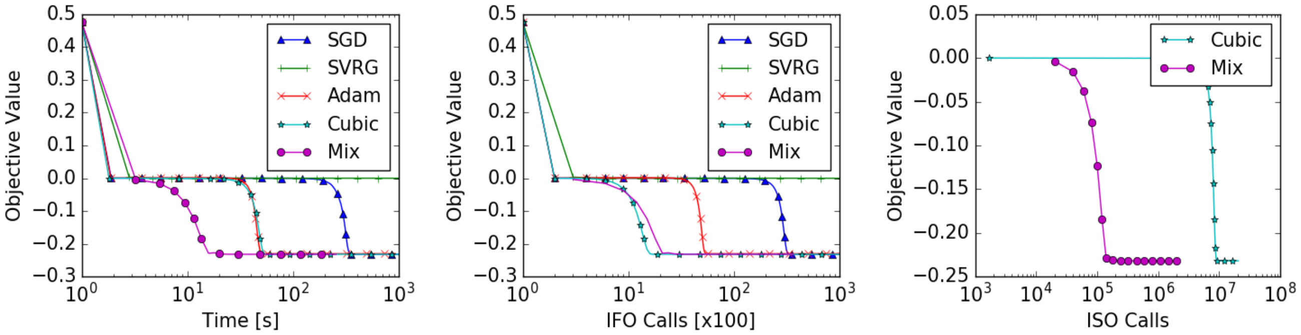

Figure 4: Comparison of various methods on a synthetic problem. Our mix framework successfully escapes saddle point.

2. Solve the sub-problem

$$v^* = \arg\min_v \langle g, v\rangle + \frac{1}{2}\langle v, Hv\rangle + \frac{M}{6}\|v\|^3 \tag{19}$$

3. Update: $x \leftarrow x + v^*$

We found that this mini-batch training strategy, which requires the computation of the gradient and the Hessian on a small subset of the dataset, to work well on a few datasets (CURVES, MNIST, CIFAR10). A similar method has been analysed in Cartis and Scheinberg [2017].

Furthermore, in many deep-networks, adaptive per-parameter learning rate helps immensely Kingma and Ba [2014]. One possible explanation for this is that the scale of the gradients in each layer of the network often differ by several orders of magnitude. A well-suited optimization method should take this into account. This is the reason for popularity of methods like Adam or RMSprop in the deep learning community. On similar lines, to account for different per-parameter behaviour in cubic regularization, we modify the sub-problem by adding a diagonal matrix M_d in addition to the scalar regularization coefficient M, i.e.,

$$\min_v \langle g, v\rangle + \frac{1}{2}\langle v, Hv\rangle + \frac{1}{6}M\|M_d v\|^3. \tag{20}$$

Also we devised an adaptive rule to obtain the diagonal matrix as $M_d = \mathsf{diag}((s + 10^{-12})^{1/9})$, where s is maintained as a moving average of third order polynomial of the mini-batch gradient g, in a fashion similar to RMSprop and Adam:

$$s \leftarrow \beta s + (1-\beta)(|g|^3 + 2g^2), \tag{21}$$

where $|g|^3$ and g^2 are vectors such that $[|g|^3]_i = |g_i|^3$ and $[g^2]_i = g_i^2$ respectively for all $i \in [n]$. The experiments reported on CURVES and MNIST in this paper utilizes both the above modifications to the cubic regularization, with β set to 0.9. We refer to this modified procedure as ACubic in our results.

G Experiment Details

In this section we provide further experimental details and results to aid reproducibility.

G.1 Synthetic Problem

The parameter selection for all the methods were carried as follows:

1. Sgd: The scalar step-size was determined by a grid search.
2. Adam: We performed a grid search over α and ε parameters of Adam tied together, i.e., $\alpha = \varepsilon$.
3. Svrg: The scalar step-size was determined by a grid search.
4. CubicDescent: The regularization parameter M was chosen by grid search. The sub-problem was solved with gradient descent Carmon et al. [2016] with the step-size of solver to be 10^{-2} and run till the gradient norm of the sub-problem is reduced below 10^{-3}.

Further Observations The results are presented in Figure 4. The other first order methods like Adam with higher noise could escape relatively faster whereas Svrg with reduced noise stayed stuck at the saddle point.

G.2 Deep Networks

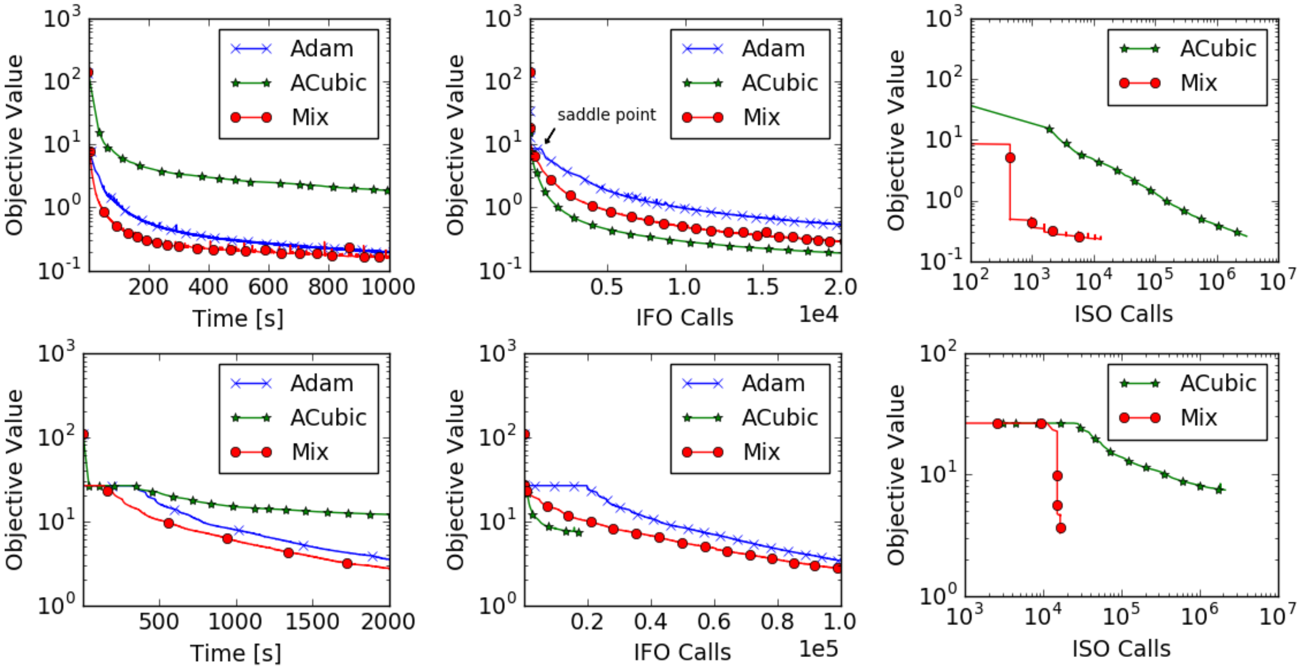

Figure 5: Comparison of various methods on a Deep Autoencoder on CURVES (top) and MNIST (bottom). Our mix approach converges faster than the baseline methods and uses relatively few ISO calls in comparison to ApproxCubicDescent

Methods The parameter selection for all the methods were carried as follows::

1. Adam: We performed a grid search over α and ε parameters of Adam so as to produce the best generalization on a held out test set. We found it to be $\alpha = 10^{-3}, \varepsilon = 10^{-3}$ for CURVES and $\alpha = 10^{-2}, \varepsilon = 10^{-1}$ for MNIST.
2. ApproxCubicDescent: The regularization parameter M was chosen as the largest value such function value does not jump in first 10 epochs. We found it to be $M = 10^3$ for both CURVES and MNIST. The sub-problem was solved with gradient descent Carmon et al. [2016] with the step-size of solver to be 10^{-3} and run till the gradient norm of the sub-problem is reduced below 0.1.

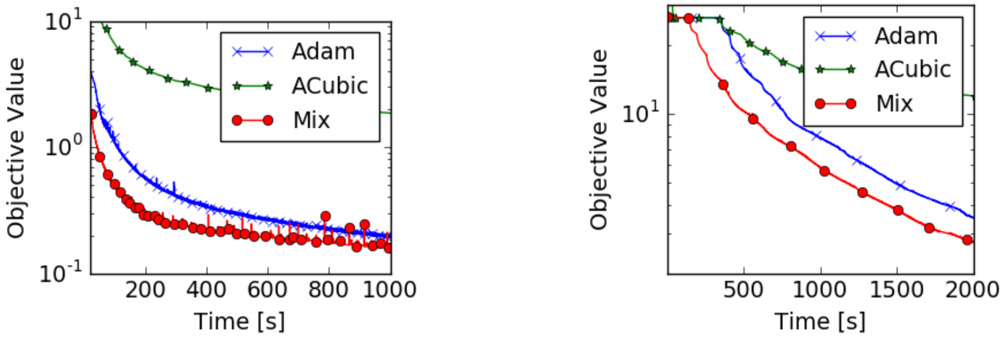

Figure 6: Zoomed in version of plots with respect to time. Here we show progress from time=10s onwards. This better exhibits the relative differences between the methods, and is illustrative of the advantage of our method.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `http://tensorflow.org/`. Software available from tensorflow.org.

Alekh Agarwal and Leon Bottou. A lower bound for the optimization of finite sums. *arXiv:1410.0723*, 2014.

Naman Agarwal, Zeyuan Allen Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima for nonconvex optimization in linear time. *CoRR*, abs/1611.01146, 2016a.

Naman Agarwal, Brian Bullins, and Elad Hazan. Second order stochastic optimization in linear time. *CoRR*, abs/1602.03943, 2016b.

Léon Bottou. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nımes*, 91(8), 1991.

Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for non-convex optimization. *CoRR*, abs/1611.00756, 2016.

Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. "convex until proven guilty": Dimension-free acceleration of gradient descent on non-convex functions. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 654–663, 2017.

C. Cartis and K. Scheinberg. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Mathematical Programming*, pages 1–39, 2017. ISSN 1436-4646. doi: 10.1007/s10107-017-1137-4.

Anna Choromanska, Mikael Henaff, Michaël Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surface of multilayer networks. *CoRR*, abs/1412.0233, 2014.

Yann Dauphin, Harm de Vries, and Yoshua Bengio. Equilibrated adaptive learning rates for non-convex optimization. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1504–1512. Curran Associates, Inc., 2015.

Yann N. Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, pages 2933–2941, 2014.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS 27*, pages 1646–1654, 2014a.

Aaron J Defazio, Tibério S Caetano, and Justin Domke. Finito: A faster, permutable incremental gradient method for big data problems. *arXiv:1407.2710*, 2014b.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points - online stochastic gradient for tensor decomposition. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015*, pages 797–842, 2015.

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23 (4):2341–2368, 2013. doi: 10.1137/120880811.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. *CoRR*, abs/1703.00887, 2017.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS 26*, pages 315–323, 2013.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Jakub Konečný, Jie Liu, Peter Richtárik, and Martin Takáč. Mini-Batch Semi-Stochastic Gradient Descent in the Proximal Setting. *arXiv:1504.04407*, 2015.

Harold Joseph Kushner and Dean S Clark. *Stochastic approximation methods for constrained and unconstrained systems*, volume 26. Springer Science & Business Media, 2012.

Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. *arXiv:1507.02000*, 2015.

Kfir Y. Levy. The power of normalization: Faster evasion of saddle points. *CoRR*, abs/1611.04831, 2016.

Xiangru Lian, Yijun Huang, Yuncheng Li, and Ji Liu. Asynchronous Parallel Stochastic Gradient for Nonconvex Optimization. In *NIPS*, 2015.

Lennart Ljung. Analysis of recursive stochastic algorithms. *Automatic Control, IEEE Transactions on*, 22(4):551–575, 1977.

James Martens. Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 735–742, 2010.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning*, pages 2408–2417, 2015.

Yurii Nesterov. *Introductory Lectures On Convex Optimization: A Basic Course*. Springer, 2003.

Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

Barak A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, 6(1):147–160, January 1994. ISSN 0899-7667.

BT Poljak and Ya Z Tsypkin. Pseudogradient adaptation and training algorithms. *Automation and Remote Control*, 34:45–67, 1973.

Sashank Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex J Smola. On variance reduction in stochastic gradient descent and its asynchronous variants. In *NIPS 28*, pages 2629–2637, 2015.

Sashank J. Reddi, Ahmed Hefny, Suvrit Sra, Barnabás Póczos, and Alexander J. Smola. Stochastic variance reduction for nonconvex optimization. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 314–323, 2016a.

Sashank J. Reddi, Suvrit Sra, Barnabás Póczos, and Alexander J. Smola. Fast incremental method for nonconvex optimization. *CoRR*, abs/1603.06159, 2016b.

Sashank J. Reddi, Suvrit Sra, Barnabás Póczos, and Alexander J. Smola. Fast stochastic methods for nonsmooth nonconvex optimization. *CoRR*, 2016c.

H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

Mark W. Schmidt, Nicolas Le Roux, and Francis R. Bach. Minimizing Finite Sums with the Stochastic Average Gradient. *arXiv:1309.2388*, 2013.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1):567–599, 2013.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

Oriol Vinyals and Daniel Povey. Krylov subspace descent for deep learning. In *AISTATS*, pages 1261–1268, 2012.