Parameter-free Clipped Gradient Descent Meets Polyak

Yuki Takezawa[1,2], **Han Bao**[1,2], **Ryoma Sato**[3], **Kenta Niwa**[4], **Makoto Yamada**[2]
[1]Kyoto University, [2]OIST, [3]NII, [4]NTT Communication Science Laboratories

Abstract

Gradient descent and its variants are de facto standard algorithms for training machine learning models. As gradient descent is sensitive to its hyperparameters, we need to tune the hyperparameters carefully using a grid search. However, the method is time-consuming, particularly when multiple hyperparameters exist. Therefore, recent studies have analyzed parameter-free methods that adjust the hyperparameters on the fly. However, the existing work is limited to investigations of parameter-free methods for the stepsize, and parameter-free methods for other hyperparameters have not been explored. For instance, although the gradient clipping threshold is a crucial hyperparameter in addition to the stepsize for preventing gradient explosion issues, none of the existing studies have investigated parameter-free methods for clipped gradient descent. Therefore, in this study, we investigate the parameter-free methods for clipped gradient descent. Specifically, we propose Inexact Polyak Stepsize, which converges to the optimal solution without any hyperparameters tuning, and its convergence rate is asymptotically independent of L under L-smooth and (L_0, L_1)-smooth assumptions of the loss function, similar to that of clipped gradient descent with well-tuned hyperparameters. We numerically validated our convergence results using a synthetic function and demonstrated the effectiveness of our proposed methods using LSTM, Nano-GPT, and T5.

1 Introduction

We consider the convex optimization problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}), \tag{1}$$

where the loss function f is convex and lower bounded. In this setting, gradient descent and its variants (Duchi et al., 2011; Kingma and Ba, 2015) are the de facto standard algorithms to minimize the loss function. The performance of the algorithm is highly sensitive to the hyperparameter settings, necessitating the careful tuning of the hyperparameters to achieve best performance. More specifically, when the loss function is L-smooth, gradient descent can achieve the optimal convergence rate $\mathcal{O}(\frac{L\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2}{T})$ when we set the stepsize to $\frac{1}{L}$ where $\boldsymbol{x}_0$ is the initial parameter and $\boldsymbol{x}^\star$ is the optimal solution (Nesterov, 2018). Unfortunately, parameter L is problem-specific and unavailable in practice. Thus, gradient descent must be executed in many times with different hyperparameters to identify the good hyperparameter settings, which is a very time-consuming process. Notably, when multiple hyperparameters are under consideration, this hyperparameter search becomes computationally more demanding.

Several recent studies have examined *parameter-free methods* for tuning hyperparameters on the fly (Berrada et al., 2020; Defazio and Mishchenko, 2023; Orvieto et al., 2022; Jiang and Stich, 2023; Ivgi et al., 2023; Khaled et al., 2023; Orabona and Tommasi, 2017; Carmon and Hinder, 2022).[1]

[1]We use *parameter-free methods* to describe algorithms that provably converge to the optimal solution without any problem-specific parameters for setting their stepsizes and other hyperparameters.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-1416

These methods automatically adjust the stepsize during the training and are guaranteed to converge to the optimal solution without tuning the stepsize. In other words, the stepsize did not require tuning using the grid search. However, the existing parameter-free methods only focus on the stepsize, and parameter-free methods for other hyperparameters have not been explored. For example, in addition to the stepsize, the gradient clipping threshold is an important hyperparameter for training language models (Pascanu et al., 2013; Zhang et al., 2020a,b,c).

Clipped gradient descent can achieve the convergence rate $\mathcal{O}(\frac{L_0\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2}{T})$ under the assumption that the loss function is (L_0, L_1)-smooth when we use the optimal stepsize and gradient clipping threshold (Koloskova et al., 2023). In many cases, L_0 is significantly smaller than L (Zhang et al., 2020b). Thus, by comparing with the convergence rate of gradient descent $\mathcal{O}(\frac{L\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2}{T})$, gradient clipping often allows gradient descent to converge faster. However, we must carefully tune two hyperparameters, stepsize and gradient clipping threshold, to achieve this convergence rate. If the gradient clipping threshold is too large, the gradient clipping fails to accelerate the convergence. Moreover, if the gradient clipping threshold is too small, gradient clipping deteriorates rather than accelerating the convergence rate. *Can we develop a parameter-free method whose convergence rate is asymptotically independent of L under (L_0, L_1)-smoothness?*

In this study, we investigate a parameter-free method for clipped gradient descent. First, we provide the better convergence rate of Polyak stepsize (Polyak, 1987) under (L_0, L_1)-smoothness. We discover that the convergence rate of Polyak stepsize matches that of clipped gradient descent with well-tuned stepsize and gradient clipping threshold. Although the convergence rate of Polyak stepsize is asymptotically independent of L under (L_0, L_1)-smooth assumption as clipped gradient descent, it still requires the minimum loss value, which is a problem-specific value. Thus, we make Polyak stepsize parameter-free without losing this property under (L_0, L_1)-smoothness by proposing **Inexact Polyak Stepsize**, which converges to the optimal solution without any problem-specific parameters. We numerically evaluated Inexact Polyak Stepsize using a synthetic function and neural networks, validating our theory and demonstrating the effectiveness of Inexact Polyak Stepsize.

2 Preliminary

2.1 Gradient descent & L-smoothness

One of the most fundamental algorithms for solving Eq. (1) represents the gradient descent:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \nabla f(\boldsymbol{x}_t),$$

where $\boldsymbol{x}_0 \in \mathbb{R}^d$ is the initial parameter and $\eta_t > 0$ is the stepsize at t-th iteration. To ensure that gradient descent converges to the optimal solution quickly, we must carefully tune the stepsize η_t. When the stepsize is too large, the training collapses. By contrast, when the stepsize is too small, the convergence rate becomes too slow. Thus, we must search for a proper stepsize as the following theorem indicates.

Assumption 1 (L-smoothness)**.** *There exists a constant $L > 0$ that satisfies the following for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$:*

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|. \tag{2}$$

Theorem 1 (Nesterov (2018, Corollary 2.1.2))**.** *Assume that f is convex and L-smooth, and there exists an optimal solution $\boldsymbol{x}^\star \coloneqq \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$. Then, gradient descent with stepsize $\eta_t = \frac{1}{L}$ satisfies*

$$f(\bar{\boldsymbol{x}}) - f(\boldsymbol{x}^\star) \leq \mathcal{O}\left(\frac{L\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2}{T}\right), \tag{3}$$

where $\bar{\boldsymbol{x}} \coloneqq \frac{1}{T}\sum_{t=0}^{T-1} \boldsymbol{x}_t$ and T is the number of iterations.

2.2 Clipped gradient descent & (L_0, L_1)-smoothness

Gradient clipping is widely used to stabilize and accelerate the training of gradient descent (Pascanu et al., 2013; Devlin et al., 2019). Let $c > 0$ be the threshold for gradient clipping. Clipped gradient descent is given by:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \min\left\{1, \frac{c}{\|\nabla f(\boldsymbol{x}_t)\|}\right\} \nabla f(\boldsymbol{x}_t). \tag{4}$$

Many prior studies investigated the theoretical benefits of gradient clipping (Koloskova et al., 2023; Zhang et al., 2020a,b,c; Li and Liu, 2022; Sadiev et al., 2023). Zhang et al. (2020b) experimentally found that the gradient Lipschitz constant decreases during the training of various neural networks and is highly correlated with gradient norm $\|\nabla f(\boldsymbol{x})\|$. To describe this phenomenon, Zhang et al. (2020a) introduced a novel smoothness assumption called (L_0, L_1)-smoothness. Then, it has been experimentally demonstrated that the local gradient Lipschitz constant L_0 is thousands of times smaller than the global gradient Lipschitz constant L.

Assumption 2 ((L_0, L_1)-smoothness)**.** *There exists constants $L_0 > 0$ and $L_1 > 0$ that satisfy the following for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ with $\|\boldsymbol{x} - \boldsymbol{y}\| \leq \frac{1}{L_1}$:*

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq (L_0 + L_1 \|\nabla f(\boldsymbol{x})\|)\|\boldsymbol{x} - \boldsymbol{y}\|. \tag{5}$$

Note that (L_0, L_1)-smoothness is strictly weaker than L-smoothness because (L_0, L_1)-smoothness covers L-smoothness by taking $L_1 = 0$. Using the (L_0, L_1)-smoothness assumption, the convergence rate of clipped gradient descent was established as follows.

Theorem 2 (Koloskova et al. (2023, Theorem 2.3))**.** *Assume that f is convex, L-smooth, and (L_0, L_1)-smooth, and there exists an optimal solution $\boldsymbol{x}^\star \coloneqq \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x})$. Then, clipped gradient descent with $\eta_t = \frac{1}{L_0}$ and $c = \frac{L_0}{L_1}$ satisfies:*

$$f(\bar{\boldsymbol{x}}) - f(\boldsymbol{x}^\star) \leq \mathcal{O}\left(\frac{L_0 \|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2}{T} + \frac{L L_1^2 \|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^4}{T^2}\right), \tag{6}$$

where $\bar{\boldsymbol{x}} \coloneqq \frac{1}{T}\sum_{t=0}^{T-1} \boldsymbol{x}_t$ and T is the number of iterations.

When the number of iterations T is large, the first term $\mathcal{O}(\frac{L_0 \|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2}{T})$ becomes dominant, and the convergence rate of clipped gradient descent is asymptotically independent of L. Gradient clipping allows for the use of a larger stepsize, and thus, gradient descent converges faster because of $L_0 \ll L$. We can interpret $L \simeq L_0 + L_1 \sup_{\boldsymbol{x}} \|\nabla f(\boldsymbol{x})\|$. The stepsize of gradient descent in Theorem 1 is $\frac{1}{L_0 + L_1 \sup_{\boldsymbol{x}} \|\nabla f(\boldsymbol{x})\|}$, which is typically very small. By comparing with gradient descent, the coefficient multiplied by the gradient of clipped gradient descent in Theorem 2 is $\min\{\frac{1}{L_0}, \frac{1}{L_1 \|\nabla f(\boldsymbol{x}_t)\|}\}$, which is larger than $\frac{1}{L_0 + L_1 \sup_{\boldsymbol{x}} \|\nabla f(\boldsymbol{x})\|}$. Specifically, when parameter $\boldsymbol{x}$ is close to the optimal solution $\boldsymbol{x}^\star$ (i.e., $\|\nabla f(\boldsymbol{x})\|$ is small), clipped gradient descent can use a larger stepsize and then reach the optimal solution faster than gradient descent.

2.3 Polyak stepsize

When f is convex, $\boldsymbol{x}_{t+1}$ and $\boldsymbol{x}_t$ generated by gradient descent satisfy $\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 \leq \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - 2\eta_t(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)) + \eta_t^2 \|\nabla f(\boldsymbol{x}_t)\|^2$. By minimizing the right-hand side, we can derive well-known Polyak stepsize (Polyak, 1987):

$$\eta_t = \frac{f(\boldsymbol{x}_t) - f^\star}{\|\nabla f(\boldsymbol{x}_t)\|^2}, \tag{7}$$

where $f^\star \coloneqq f(\boldsymbol{x}^\star)$. When f is L-smooth, gradient descent with Polyak stepsize converges to the optimal solution as quickly as gradient descent with $\eta_t = \frac{1}{L}$.

Theorem 3 (Hazan and Kakade (2019, Theorem 1))**.** *Assume that f is convex and L-smooth, and there exists an optimal solution $\boldsymbol{x}^\star \coloneqq \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x})$. Then, gradient descent with Polyak stepsize Eq. (7) satisfies:*

$$f(\bar{\boldsymbol{x}}) - f(\boldsymbol{x}^\star) \leq \mathcal{O}\left(\frac{L \|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2}{T}\right), \tag{8}$$

where $\bar{\boldsymbol{x}} \coloneqq \frac{1}{T}\sum_{t=0}^{T-1} \boldsymbol{x}_t$ and T is the number of iterations.

In addition to the L-smooth setting, Polyak stepsize is known to cause gradient descent to converge to the optimal solution with the optimal rate among various settings, e.g., non-smooth convex, smooth convex, and strongly convex settings (Hazan and Kakade, 2019).

https://doi.org/10.52202/079017-1416

3 Improved convergence result of Polyak stepsize

Before proposing a parameter-free method for clipped gradient descent, in this section, we present a new convergence analysis of Polyak stepsize under (L_0, L_1)-smoothness. Surprisingly, our new analysis reveals that Polyak stepsize achieves exactly the same convergence rate as *clipped* gradient descent with appropriate hyperparameters. A bunch of prior studies established the convergence rates of Polyak stepsize, and it is well-known that Polyak stepsize allows gradient descent to converge as fast as the optimal stepsize. However, our theorem finds that Polyak stepsize achieves a faster convergence rate than gradient descent with the optimal stepsize as clipped gradient descent, and none of the existing studies have found this favorable property of Polyak stepsize.

3.1 Connection between Polyak stepsize and clipped gradient descent

Under (L_0, L_1)-smoothness, we can obtain the following results.

Proposition 1. *Assume that f is convex and (L_0, L_1)-smooth. Then, Polyak stepsize Eq. (7) satisfies:*

$$\min\left\{\frac{1}{4L_0}, \frac{1}{4L_1\|\nabla f(\boldsymbol{x}_t)\|}\right\} \leq \frac{f(\boldsymbol{x}_t) - f^\star}{\|\nabla f(\boldsymbol{x}_t)\|^2}. \tag{9}$$

Proof. Assumption 2 and Lemma 2 imply

$$\frac{f(\boldsymbol{x}_t) - f^\star}{\|\nabla f(\boldsymbol{x}_t)\|^2} \geq \frac{1}{2(L_0 + L_1\|\nabla f(\boldsymbol{x}_t)\|)}.$$

When $\|\nabla f(\boldsymbol{x}_t)\| < \frac{L_0}{L_1}$, Polyak stepsize is bounded from below by $\frac{1}{4L_0}$. When $\|\nabla f(\boldsymbol{x}_t)\| \geq \frac{L_0}{L_1}$, we have

$$\frac{1}{2(L_0 + L_1\|\nabla f(\boldsymbol{x}_t)\|)} \geq \frac{1}{4L_1\|\nabla f(\boldsymbol{x}_t)\|}.$$

Therefore, we can conclude the statement. □

Under L-smoothness, the lower bound of Polyak stepsize was obtained as follows.

Proposition 2 (Jiang and Stich (2023, Lemma 15))**.** *Assume that f is convex and L-smooth. Then, Polyak stepsize Eq. (7) satisfies:*

$$\frac{1}{2L} \leq \frac{f(\boldsymbol{x}_t) - f^\star}{\|\nabla f(\boldsymbol{x}_t)\|^2}. \tag{10}$$

By comparing Propositions 1 and 2, Proposition 1 shows that Polyak stepsize does not become excessively small when the parameter approaches the optimal solution (i.e., $\|\nabla f(\boldsymbol{x})\|$ approaches zero), similar to clipped gradient descent. If we choose the stepsize and gradient clipping threshold as in Theorem 2, clipped gradient descent can be written as follows:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \min\left\{\frac{1}{L_0}, \frac{1}{L_1\|\nabla f(\boldsymbol{x}_t)\|}\right\}\nabla f(\boldsymbol{x}_t). \tag{11}$$

Thus, Proposition 1 implies that Polyak stepsize can be regarded as internally estimating the hyperparameters for clipped gradient descent, as shown in Theorem 2.

3.2 Convergence analysis of Polyak stepsize under (L_0, L_1)-smoothness

Based on the relationship between Polyak stepsize and clipped gradient descent in Sec. 3.1, we provide a new convergence result for Polyak stepsize under (L_0, L_1)-smoothness. The proof is deferred to Sec. A.

Theorem 4. *Assume that f is convex, L-smooth, and (L_0, L_1)-smooth, and there exists an optimal solution $\boldsymbol{x}^\star := \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$. Let T be the number of iterations and define $\tau := \arg\min_{0\leq t\leq T-1} f(\boldsymbol{x}_t)$. Then, gradient descent with Polyak stepsize Eq. (7) satisfies:*

$$f(\boldsymbol{x}_\tau) - f(\boldsymbol{x}^\star) \leq \mathcal{O}\left(\frac{L_0\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2}{T} + \frac{LL_1^2\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^4}{T^2}\right). \tag{12}$$

By comparing Theorem 4 with Theorem 2, the convergence rate of Polyak stepsize is the same as that of clipped gradient descent. Thus, Polyak stepsize can converge faster than the optimal stepsize given in Theorem 1 when $L_0 \ll L$. Many prior studies analyzed the convergence rate of Polyak stepsize and discussed the relationship between Polyak stepsize and gradient descent with the optimal stepsize (Polyak, 1987; Loizou et al., 2021; Galli et al., 2023; Berrada et al., 2020). However, they only recognized Polyak stepsize as making gradient descent converge with the same convergence rate as the optimal stepsize, and none of the prior studies have found this relationship between Polyak stepsize and clipped gradient descent. Our new convergence result is the first to discover that the Polyak stepsize can achieve the same convergence rate not only as gradient descent with an appropriate stepsize but also as clipped gradient descent with an appropriate stepsize and gradient clipping threshold.

4 Making clipped gradient descent parameter-free

In the previous section, we found that the convergence rate of Polyak stepsize is asymptotically independent of L under (L_0, L_1)-smoothness as clipped gradient descent with appropriate hyperparameters. However, Polyak stepsize requires the minimum loss value $f^\star$, which is a problem-specific parameter. In this section, we propose a method that can remove the prior knowledge of $f^\star$ from Polyak stepsize without losing the property of asymptotic independence of L under (L_0, L_1)-smoothness.

4.1 Inexact Polyak Stepsize

To make Polyak stepsize parameter-free, several prior studies have proposed the use of lower bound of $f^\star$ instead of $f^\star$ (Loizou et al., 2021; Orvieto et al., 2022; Jiang and Stich, 2023). The loss functions commonly used in machine learning models are non-negative. Thus, the lower bound of $f^\star$ is trivially obtained as zero and is not a problem-specific parameter. By utilizing this lower bound, a straightforward approach to make Polyak stepsize independent of problem-specific parameters is replacing $f^\star$ in Polyak stepsize with the lower bound $l^\star$ as follows:

$$\eta_t = \frac{f(\boldsymbol{x}_t) - l^\star}{\|\nabla f(\boldsymbol{x}_t)\|^2}. \tag{13}$$

However, the stepsize in Eq. (13) becomes excessively large as the parameter approaches the optimal solution, and it does not lead to the optimal solution (Loizou et al., 2021). This is because $\|\nabla f(\boldsymbol{x}_t)\|$ approaches zero, while $f(\boldsymbol{x}_t) - l^\star$ approaches $f^\star - l^\star (> 0)$, which makes the stepsize in Eq. (13) excessively large as the parameter approaches the optimal solution. To mitigate this issue, DecSPS (Orvieto et al., 2022) and AdaSPS (Jiang and Stich, 2023), which are parameter-free methods based on Polyak stepsize that use $l^\star$ instead of $f^\star$, make the stepsize monotonically non-increasing to converge to the optimal solution.

However, making the stepsize monotonically non-increasing loses the fruitful property that the convergence rate of Polyak stepsize is asymptotically independent of L as clipping gradient descent under (L_0, L_1)-smoothness. This is because Polyak stepsize and clipped gradient descent make the convergence rate asymptotically independent of L by increasing the stepsize when the parameter approaches the optimal solution. In fact, we evaluated DecSPS and AdaSPS with a synthetic function in Sec. 6.1, demonstrating that the convergence deteriorates as L increases.

Algorithm 1 Inexact Polyak Stepsize

Input: The number of iterations T and lower bound $l^\star$.
$f^{\text{best}}, \boldsymbol{x}^{\text{best}} \leftarrow f(\boldsymbol{x}_0), \boldsymbol{x}_0$.
for $t = 0, 1, \cdots, T-1$ **do**
    $\boldsymbol{x}_{t+1} \leftarrow \boldsymbol{x}_t - \frac{f(\boldsymbol{x}_t) - l^\star}{\sqrt{T}\|\nabla f(\boldsymbol{x}_t)\|^2} \nabla f(\boldsymbol{x}_t)$.
    if $f(\boldsymbol{x}_{t+1}) \leq f^{\text{best}}$ **then**
        $f^{\text{best}}, \boldsymbol{x}^{\text{best}} \leftarrow f(\boldsymbol{x}_{t+1}), \boldsymbol{x}_{t+1}$.
return $\boldsymbol{x}^{\text{best}}$.

https://doi.org/10.52202/079017-1416

To address this issue, we propose **Inexact Polyak Stepsize**, whose details are described in Alg. 1. As discussed above, we cannot make the stepsize decrease to maintain the asymptotic independence of L under (L_0, L_1)-smoothness. Thus, we set the stepsize as follows:

$$\eta_t = \frac{f(\boldsymbol{x}_t) - l^\star}{\sqrt{T}\|\nabla f(\boldsymbol{x}_t)\|^2}, \tag{14}$$

where T denotes the number of iterations. Instead of making the stepsize decrease, we propose returning the parameter for which the lowest loss is achieved as the final parameter.

4.2 Convergence analysis of Inexact Polyak Stepsize

The following theorem provides the convergence rate of Inexact Polyak Stepsize. The proof is deferred to Sec. B.

Theorem 5. *Assume that f is convex, L-smooth, and (L_0, L_1)-smooth, and there exists an optimal solution $\boldsymbol{x}^\star \coloneqq \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$. Let T be the number of iterations and $\sigma^2 \coloneqq f^\star - l^\star$. Then, $\boldsymbol{x}$ generated by Alg. 1 satisfies:*

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^\star) \leq \mathcal{O}\left(\frac{L_0\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 + \sigma^2}{\sqrt{T}} + \frac{LL_1^2\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^4}{T} + \frac{L_1^2 L\sigma^4}{L_0^2 T}\right). \tag{15}$$

Asymptotic independence of L: When the number of iterations T is large, only the first term $\mathcal{O}(\frac{L_0\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2+\sigma^2}{\sqrt{T}})$ becomes dominant in the convergence rate, which does not depend on L. Thus, Theorem 5 shows that Inexact Polyak Stepsize successfully inherits the favorable property of Polyak stepsize under (L_0, L_1)-smoothness. In addition to Inexact Polyak Stepsieze, DecSPS (Orvieto et al., 2022) and AdaSPS (Jiang and Stich, 2023) have been proposed as parameter-free methods that use $l^\star$ instead of $f^\star$ in Polyak stepsize. However, these prior methods fail to inherit the favorable property of Polyak stepsize, and their convergence rates deteriorate when L is large because these methods decrease the stepsize during the training. In fact, we evaluated DecSPS and AdaSPS with a synthetic function in Sec. 6.1, demonstrating that convergence rates of DecSPS and AdaSPS are degraded when L becomes large, whereas the convergence rate of Inexact Polyak Stepsize does not depend on L.

Removing dependence on D_T: The convergence rates of DecSPS and AdaSPS depend on $D_T(\coloneqq \max_{0\leq t\leq T}\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|)$. Thus, strictly speaking, these convergence rates cannot show that DecSPS and AdaSPS converge to the optimal solution because D_T may increase as the number of iterations T increases. For instance, if D_T increase with $\Omega(T^{\frac{1}{4}})$, the convergence rate of AdaSPS is $\mathcal{O}(L\sigma + L^2)$, which does not show that AdaSPS converges to the optimal solution. In contrast, the convergence rate in Eq. (15) depends on only $\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|$. Theorem 5 indicates that Inexact Polyak Stepsize converges to the optimal solution.

Convergence rate with respect to T: Inexact Polyak Stepsize successfully achieves the asymptotic independence of L, while it slows down the convergence rate with respect to the number of iterations T by comparing clipped gradient descent with proper hyperparameters. The convergence rate of Inexact Polyak Stepsize $\mathcal{O}(\frac{L_0}{\sqrt{T}})$ is not optimal in terms of T, and there may be room to improve this rate. For instance, the adaptive methods proposed by Hazan and Kakade (2019) might be used to

Table 1: Summary of convergence rates of parameter-free methods based on Polyak stepsize. All convergence results are the ones under convex, L-smoothness, and (L_0, L_1)-smoothness. We define $D_T \coloneqq \max_{0\leq t\leq T}\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|$.

Algorithm	Convergence Rate	Assumption
DecSPS (Orvieto et al., 2022)[a]	$\mathcal{O}\left(\frac{\max\{L,\eta_0^{-1}\}D_T^2+\sigma^2}{\sqrt{T}}\right)$	1
AdaSPS (Jiang and Stich, 2023)[a]	$\mathcal{O}\left(\frac{LD_T^2\sigma}{\sqrt{T}} + \frac{L^2D_T^4}{T}\right)$	1
Inexact Polyak Stepsize (This work)	$\mathcal{O}\left(\frac{L_0\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2+\sigma^2}{\sqrt{T}} + \frac{LL_1^2\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^4}{T} + \frac{L_1^2L\sigma^4}{L_0^2T}\right)$	1, 2[b]

(a) We present the convergence rates of DecSPS and AdaSPS in the deterministic setting to compare DecSPS, AdaSPS, and Inexact Polyak Stepsize in the same deterministic setting, while Orvieto et al. (2022) and Jiang and Stich (2023) also analyzed the rate rates in the stochastic setting.

(b) If f is L-smooth, f is (L_0, L_1)-smooth because (L_0, L_1)-smoothness assumption is strictly weaker than L-smoothness assumption.

alleviate this issue. However, the parameter-free methods for clipped gradient descent have not been explored well in the existing studies. We believe that Inexact Polyak Stepsize is the important first step for developing parameter-free clipped gradient descent.

5 Related work

Gradient clipping: Gradient clipping was initially proposed to mitigate the gradient explosion problem for training RNN and LSTM (Mikolov et al., 2010; Merity et al., 2018) and is now widely used to accelerate and stabilize the training not only for RNN and LSTM, but also for various machine learning models, especially language models (Devlin et al., 2019; Raffel et al., 2019). Recently, many studies have investigated the theoretical benefits of gradient clipping and analyzed the convergence rate of clipped gradient descent under (1) (L_0, L_1)-smoothness assumption (Koloskova et al., 2023; Zhang et al., 2020a,b) and (2) heavy-tailed noise assumption (Zhang et al., 2020c; Li and Liu, 2022; Sadiev et al., 2023). (1) Zhang et al. (2020b) found that the local gradient Lipschitz constant is correlated with the gradient norm. To describe this phenomenon, Zhang et al. (2020b), Zhang et al. (2020a), and Koloskova et al. (2023) introduced the new assumption, (L_0, L_1)-smoothness, providing the convergence rate of clipped gradient descent under (L_0, L_1)-smoothness. Then, they showed that gradient clipping can improve the convergence rate of gradient descent, as we introduced in Sec. 2.2. (2) Besides (L_0, L_1)-smoothness, Zhang et al. (2020c) pointed out that the distribution of stochastic gradient noise is heavy-tailed for language models. Then, it has been shown that gradient clipping can make the stochastic gradient descent robust against the heavy-tailed noise of stochastic gradient (Li and Liu, 2022; Sadiev et al., 2023; Zhang et al., 2020c).

Parameter-free methods: Hyperparameter-tuning is one of the most time-consuming tasks for training machine learning models. To alleviate this issue, many parameter-free methods that adjust the stepsize on the fly have been proposed, e.g., Polyak-based stepsize (Berrada et al., 2020; Hazan and Kakade, 2019; Loizou et al., 2021; Mukherjee et al., 2023; Orvieto et al., 2022; Jiang and Stich, 2023), AdaGrad-based methods (Ivgi et al., 2023; Khaled et al., 2023), and Dual Averaging-based methods (Orabona and Tommasi, 2017; Defazio and Mishchenko, 2023). However, parameter-free methods for hyperparameters, except for stepsizes, have not been studied. In this work, we studied the parameter-free methods for two hyperparameters, the stepsize and gradient clipping threshold, and then proposed Inexact Polyak Stepsize, which converges to the optimal solution without tuning any hyperparameters and its convergence rate is asymptotically independent of L as clipped gradient descent with well-tuned hyperparameters.

6 Numerical evaluation

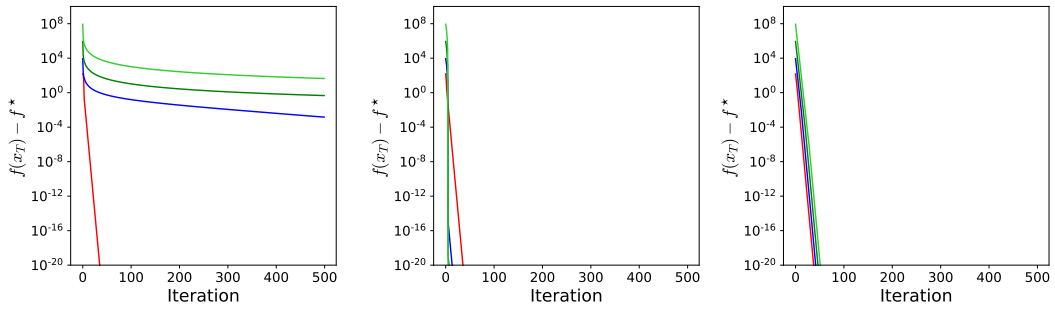
In this section, we evaluate our theory numerically. In Sec. 6.1, we evaluate Polyak stepsize and Inexact Polyak Stepsize using a synthetic function, varying that their convergence rates are asymptotically independent of L. In Sec. 6.2, we show the results obtained using neural networks.

6.1 Synthetic function

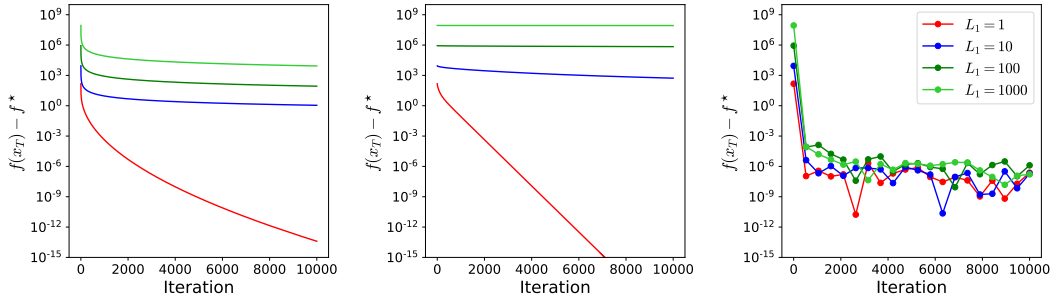
Setting: In this section, we validate our theory for Polyak stepsize and Inexact Polyak Stepsize using a synthetic function. We set the loss function as $f(x) = \frac{L_0 L_1^2}{72}x^4 + \frac{L_0}{4}x^2 + f^\star$, which is (L_0, L_1)-smooth for any $L_0 > 0$ and $L_1 > 0$ (See Proposition 3 in Appendix). We set L_0 to 1, $\boldsymbol{x}_0$ to 5, $f^\star = 1$, and $l^\star = 0$ and then evaluated various methods when varying L_1.

Results: We show the results in Fig. 1. The results indicate that gradient descent converges slowly when L_1 is large, whereas Polyak stepsize and clipped gradient descent does not depend on L_1. These observations are consistent with those discussed in Sec. 3, which shows that the convergence rate of Polyak stepsize is asymptotically independent of L as in clipped gradient descent. By comparing DecSPS, AdaSPS, and Inexact Polyak Stepsize, which are parameter-free methods, the convergence rates of DecSPS and AdaSPS degrade as L_1 increases. Thus, DecSPS and AdaSPS lose the favorable property of asymptotic independence of L under (L_0, L_1)-smoothness. In contrast, the convergence behavior of Inexact Polyak Stepsize does not depend on L_1, which is consistent with Theorem 5, and Inexact Polyak Stepsize successfully inherits the Polyak stepsize under (L_0, L_1)-smoothness.

https://doi.org/10.52202/079017-1416

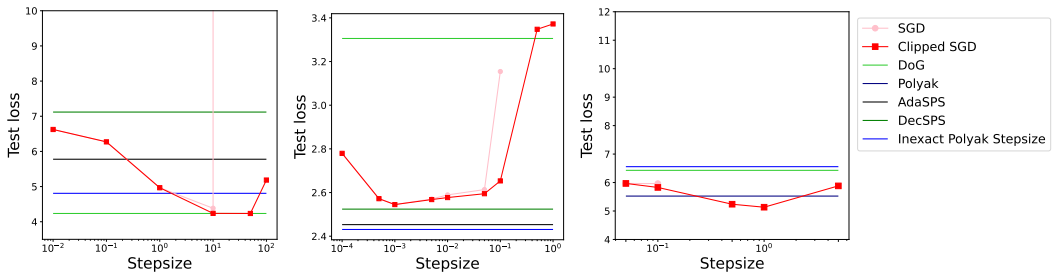


(a) Gradient Descent (b) Clipped Gradient Descent (c) Polyak Stepsize



(d) DecSPS (Orvieto et al., 2022) (e) AdaSPS (Jiang and Stich, 2023) (f) Inexact Polyak Stepsize

Figure 1: Convergence behaviors of various methods with the synthetic function.



(a) LSTM (b) Nano-GPT (c) T5

Figure 2: The final test loss with various hyperparameter settings. For T5, the results of DecSPS and AdaSPS were omitted because their final test loss was much larger than the others, as shown in Fig. 4. Furthermore, the results of SGD were also omitted when the final test loss became nan or infinity.

6.2 Neural networks

Setting: Next, we evaluated Inexact Polyak Stepsize using LSTM, Nano-GPT[2], and T5 (Nawrot, 2023). For LSTM, Nano-GPT, and T5, we used the Penn Treebank, Shakespeare, and C4 as training datasets, respectively. For SGD and Clipped SGD, we tuned the stepsize and gradient clipping threshold on validation datasets. For Polyak stepsize, we showed the results when we set $f^\star$ to zero. For Inexact Polyak Stepsize, Theorem 4 requires the selection of the best parameters. However, we do not need to choose this for neural networks because the parameters only reach the stationary point and do not reach the global minima. See Sec. D for the detailed training configuration. For all experiments, we repeated with three different seed values and reported the average.

Results: Figure 4 shows the loss curves, and Fig. 2 shows the final test losses for various hyperparameters. The results indicate that Inexact Polyak Stepsize consistently outperform DecSPS and AdaSPS for all neural network architectures. Although DoG performed the best for LSTM among the parameter-free methods, the training behavior of DoG was very unstable for Nano-GPT, and the loss values were much higher than those of the other methods. Similar to DoG, Polyak stepsize outperformed all parameter-free methods for T5, but the loss values of Polyak stepsize diverged for LSTM and Nano-GPT. Thus, Inexact Polyak Stepsize can consistently succeed in training models for all neural network architectures.

[2] `https://github.com/karpathy/nanoGPT`

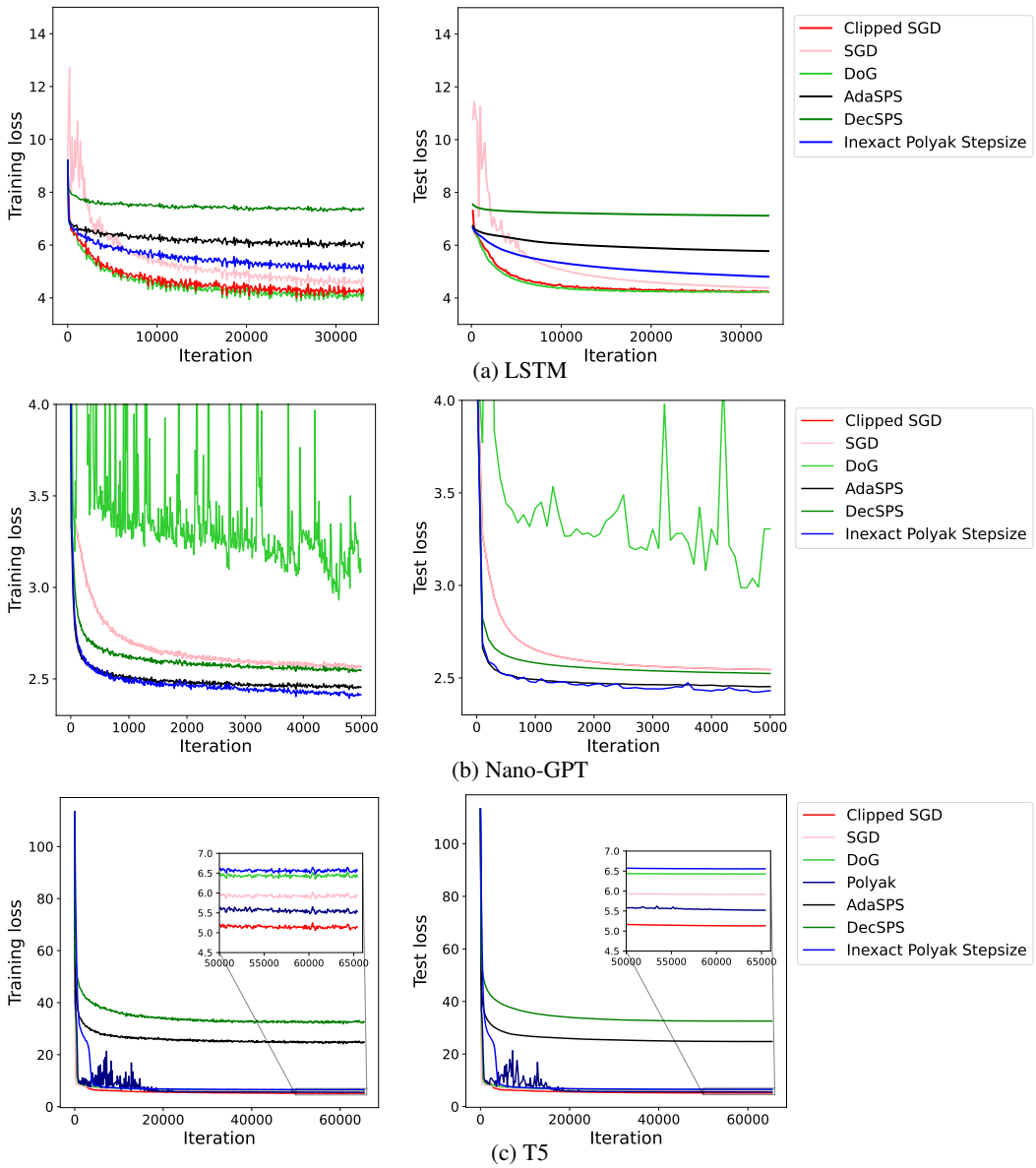


Figure 3: Loss curves for LSTM, Nano-GPT, and T5. We plotted the training loss per 100, 10, and 10 iterations for LSTM, Nano-GPT, and T5, respectively. We plotted the test loss per one epoch, 100 iterations, and 200 iterations, respectively. For LSTM and Nano-GPT, we found that Polyak stepsize does not converge, and its loss was much larger than that of other comparison methods. Thus, to make the figure easier to read, we omit the results of Polyak stepsize and provide the complete results, including Polyak stepsize in Sec. E.

7 Conclusion

In this study, we proposed Inexact Polyak Stepsize, which converges to the optimal solution without hyperparameter tuning at the convergence rate that is asymptotically independent of L under (L_0, L_1)-smoothness. Specifically, we first provided the novel convergence rate of Polyak stepsize under (L_0, L_1)-smoothness, revealing that Polyak stepsize can achieve exactly the same convergence rate as clipped gradient descent. Although Polyak stepsize can improve the convergence under (L_0, L_1)-smoothness, Polyak stepsize requires the minimum loss value, which is a problem-specific parameter. Then, we proposed Inexact Polyak Stepsize, which removes the problem-specific parameter from Polyak stepsize without losing the property of asymptotic independence of L under (L_0, L_1)-smoothness. We numerically validated our convergence results and demonstrated the effectiveness of Inexact Polyak Stepsize.

https://doi.org/10.52202/079017-1416

Acknowledgement

Y.T. was supported by KAKENHI Grant Number 23KJ1336. H.B. and M.Y. were supported by MEXT KAKENHI Grant Number 24K03004. We thank Satoki Ishikawa for his helpful comments on our experiments.

References

Berrada, L., Zisserman, A., and Kumar, M. P. (2020). Training neural networks for and by interpolation. In *International Conference on Machine Learning*.

Carmon, Y. and Hinder, O. (2022). Making SGD parameter-free. In *Conference on Learning Theory*.

Defazio, A. and Mishchenko, K. (2023). Learning-rate-free learning by D-adaptation. In *International Conference on Machine Learning*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Association for Computational Linguistics*.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. In *Journal of Machine Learning Research*.

Galli, L., Rauhut, H., and Schmidt, M. (2023). Don't be so monotone: Relaxing stochastic line search in over-parameterized models. In *Advances in Neural Information Processing Systems*.

Garrigos, G. and Gower, R. M. (2023). Handbook of convergence theorems for (stochastic) gradient methods. In *arXiv*.

Hazan, E. and Kakade, S. M. (2019). Revisiting the Polyak step size. In *arXiv*.

Ivgi, M., Hinder, O., and Carmon, Y. (2023). DoG is SGD's best friend: A parameter-free dynamic step size schedule. In *International Conference on Machine Learning*.

Jiang, X. and Stich, S. U. (2023). Adaptive SGD with Polyak stepsize and line-search: Robust convergence and variance reduction. In *Advances in Neural Information Processing Systems*.

Khaled, A., Mishchenko, K., and Jin, C. (2023). DoWG unleashed: An efficient universal parameter-free gradient descent method. In *Advances in Neural Information Processing Systems*.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Koloskova, A., Hendrikx, H., and Stich, S. U. (2023). Revisiting gradient clipping: Stochastic bias and tight convergence guarantees. In *International Conference on Machine Learning*.

Li, S. and Liu, Y. (2022). High probability guarantees for nonconvex stochastic gradient descent with heavy tails. In *International Conference on Machine Learning*.

Loizou, N., Vaswani, S., Hadj Laradji, I., and Lacoste-Julien, S. (2021). Stochastic Polyak step-size for SGD: An adaptive learning rate for fast convergence. In *International Conference on Artificial Intelligence and Statistics*.

Merity, S., Keskar, N. S., and Socher, R. (2018). Regularizing and optimizing LSTM language models. In *International Conference on Learning Representations*.

Mikolov, T., Karafiat, M., Burget, L., Cernocky, J. H., and Khudanpur, S. (2010). Recurrent neural network based language model. In *Interspeech*.

Mukherjee, S., Loizou, N., and Stich, S. U. (2023). Locally adaptive federated learning via stochastic polyak stepsizes. In *arXiv*.

Nawrot, P. (2023). NanoT5: A pytorch framework for pre-training and fine-tuning t5-style models with limited resources. In *arXiv*.

Nesterov, Y. (2018). *Lectures on Convex Optimization*. Springer.

Orabona, F. and Tommasi, T. (2017). Training deep networks without learning rates through coin betting. In *Advances in Neural Information Processing Systems*.

Orvieto, A., Lacoste-Julien, S., and Loizou, N. (2022). Dynamics of SGD with stochastic Polyak stepsizes: Truly adaptive variants and convergence to exact solution. In *Advances in Neural Information Processing Systems*.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*.

Polyak, B. (1987). *Introduction to Optimization*. Optimization Software.

Raffel, C., Shazeer, N. M., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. In *Journal of Machine Learning Research*.

Sadiev, A., Danilova, M., Gorbunov, E., Horváth, S., Gidel, G., Dvurechensky, P., Gasnikov, A., and Richtárik, P. (2023). High-probability bounds for stochastic optimization and variational inequalities: the case of unbounded variance. In *International Conference on Machine Learning*.

Zhang, B., Jin, J., Fang, C., and Wang, L. (2020a). Improved analysis of clipping algorithms for non-convex optimization. In *Advances in Neural Information Processing Systems*.

Zhang, J., He, T., Sra, S., and Jadbabaie, A. (2020b). Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*.

Zhang, J., Karimireddy, S. P., Veit, A., Kim, S., Reddi, S., Kumar, S., and Sra, S. (2020c). Why are adaptive methods good for attention models? In *Advances in Neural Information Processing Systems*.

 https://doi.org/10.52202/079017-1416

A Proof of Theorem 4

Lemma 1. *If Assumption 1 holds, the following holds for any $\boldsymbol{x} \in \mathbb{R}^d$:*

$$\frac{1}{2L}\|\nabla f(\boldsymbol{x})\|^2 \leq f(\boldsymbol{x}) - f(\boldsymbol{x}^\star). \tag{16}$$

Proof. See Lemma 2.28 in (Garrigos and Gower, 2023). □

Lemma 2. *If Assumption 2 holds, the following holds for any $\boldsymbol{x} \in \mathbb{R}^d$:*

$$\frac{1}{2(L_0 + L_1\|\nabla f(\boldsymbol{x})\|)}\|\nabla f(\boldsymbol{x})\|^2 \leq f(\boldsymbol{x}) - f(\boldsymbol{x}^\star). \tag{17}$$

Proof. See Lemma A.2 in (Koloskova et al., 2023). □

Lemma 3. *Assume that f is convex and Assumption 1 and 2 hold. Let T be the number of iterations and define $\tau \coloneqq \arg\min_{0\leq t\leq T-1} f(\boldsymbol{x}_t)$. Then, gradient descent with Polyak stepsize Eq. (7) satisfies:*

$$f(\boldsymbol{x}_\tau) - f(\boldsymbol{x}^\star) \leq \frac{8L_0\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2}{T} + \frac{64LL_1^2\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^4}{T^2}.$$

Proof. We have

$$\begin{aligned}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 &= \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - 2\eta_t\langle\nabla f(\boldsymbol{x}_t), \boldsymbol{x}_t - \boldsymbol{x}^\star\rangle + \eta_t^2\|\nabla f(\boldsymbol{x})\|^2 \\ &\leq \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - 2\eta_t(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)) + \eta_t^2\|\nabla f(\boldsymbol{x})\|^2,\end{aligned}$$

where we use the convexity of f in the inequality.

Case when $\|\nabla f(\boldsymbol{x}_t)\| \leq \frac{L_0}{L_1}$: Substituting the stepsize, we get

$$\begin{aligned}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 &\leq \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \frac{f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)}{\|\nabla f(\boldsymbol{x}_t)\|^2}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)) \\ &\leq \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \frac{1}{2(L_0 + L_1\|\nabla f(\boldsymbol{x}_t)\|)}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)) \\ &\leq \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \frac{1}{4L_0}(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)),\end{aligned}$$

where we use Lemma 2 in the second inequality. Unrolling the above inequality, we obtain

$$f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star) \leq 4L_0\left(\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2\right).$$

Case when $\|\nabla f(\boldsymbol{x}_t)\| > \frac{L_0}{L_1}$: Substituting the stepsize, we get

$$\begin{aligned}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 &\leq \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \frac{(f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star))^2}{\|\nabla f(\boldsymbol{x}_t)\|^2} \\ &\leq \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \sqrt{\frac{f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)}{2L}}\frac{f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)}{\|\nabla f(\boldsymbol{x}_t)\|},\end{aligned}$$

where we use Lemmas 1 in the last inequality. Then, $\|\nabla f(\boldsymbol{x}_t)\| > \frac{L_0}{L_1}$ implies

$$\frac{L_0 + L_1\|\nabla f(\boldsymbol{x}_t)\|}{2L_1\|\nabla f(\boldsymbol{x}_t)\|} < 1.$$

Thus, we get

$$\begin{aligned}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 &\leq \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \sqrt{\frac{f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)}{2L}}\frac{f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)}{\|\nabla f(\boldsymbol{x}_t)\|^2}\frac{L_0 + L_1\|\nabla f(\boldsymbol{x}_t)\|}{2L_1} \\ &\leq \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \frac{1}{4L_1}\sqrt{\frac{f(\boldsymbol{x}_t) - f(\boldsymbol{x}^\star)}{2L}},\end{aligned}$$

where we use Lemma 2 in the last inequality. Unrolling the above inequality and multiplying L_0 on both sides, we get

$$\frac{L_0}{L_1}\sqrt{\frac{f(\boldsymbol{x})-f(\boldsymbol{x}^\star)}{2L}} \leq 4L_0\left(\|\boldsymbol{x}_t-\boldsymbol{x}^\star\|^2-\|\boldsymbol{x}_{t+1}-\boldsymbol{x}^\star\|^2\right).$$

Summing the two cases: Define $\mathcal{T}_1$ and $\mathcal{T}_2$ as follows:

$$\mathcal{T}_1 \coloneqq \left\{t\middle|\|\nabla f(\boldsymbol{x}_t)\| \leq \frac{L_0}{L_1}\right\},\ \ \mathcal{T}_2 \coloneqq \left\{t\middle|\|\nabla f(\boldsymbol{x}_t)\| > \frac{L_0}{L_1}\right\}.$$

We obtain

$$\sum_{t\in\mathcal{T}_1}\left(f(\boldsymbol{x}_t)-f(\boldsymbol{x}^\star)\right)+\frac{L_0}{L_1}\sum_{t\in\mathcal{T}_2}\sqrt{\frac{f(\boldsymbol{x}_t)-f(\boldsymbol{x}^\star)}{2L}} \leq 4L_0\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2.$$

Then, the above inequality implies

$$\frac{1}{T}\sum_{t\in\mathcal{T}_1} f(\boldsymbol{x}_t)-f(\boldsymbol{x}^\star) \leq \frac{4L_0\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2}{T},$$
$$\frac{1}{T}\sum_{t\in\mathcal{T}_2}\sqrt{f(\boldsymbol{x}_t)-f(\boldsymbol{x}^\star)} \leq \frac{4L_1\sqrt{2L}\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2}{T}.$$

Using $a^2 \geq 2ab-b^2$, we obtain for any $b\in\mathbb{R}$

$$\frac{1}{T}\sum_{t\in\mathcal{T}_1}\left(2b\sqrt{f(\boldsymbol{x}_t)-f(\boldsymbol{x}^\star)}-b^2\right) \leq \frac{4L_0\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2}{T}.$$

Thus, when $b>0$, we obtain

$$\frac{1}{T}\sum_{t\in\mathcal{T}_1}\sqrt{f(\boldsymbol{x}_t)-f(\boldsymbol{x}^\star)} \leq \frac{4L_0\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2}{2bT}+\frac{b}{2}.$$

Choosing $b=\sqrt{\frac{4L_0\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2}{T}}$, we get

$$\frac{1}{T}\sum_{t\in\mathcal{T}_1}\sqrt{f(\boldsymbol{x}_t)-f(\boldsymbol{x}^\star)} \leq \sqrt{\frac{4L_0\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2}{T}}.$$

Thus, we get

$$\frac{1}{T}\sum_{t=0}^{T-1}\sqrt{f(\boldsymbol{x}_t)-f(\boldsymbol{x}^\star)} \leq \sqrt{\frac{4L_0\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2}{T}}+\frac{4L_1\sqrt{2L}\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2}{T}.$$

Defining $\tau \coloneqq \arg\min_t f(\boldsymbol{x}_t)$, we get

$$\sqrt{f(\boldsymbol{x}_\tau)-f(\boldsymbol{x}^\star)} \leq \sqrt{\frac{4L_0\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2}{T}}+\frac{4L_1\sqrt{2L}\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2}{T}.$$

Squaring the both sides, and using $(a+b)^2 \leq 2a^2+2b^2$ for all $a,b\in\mathbb{R}$, we obtain

$$f(\boldsymbol{x}_\tau)-f(\boldsymbol{x}^\star) \leq \frac{8L_0\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2}{T}+\frac{64LL_1^2\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^4}{T^2}.$$

This concludes the statement. □

https://doi.org/10.52202/079017-1416

B Proof of Theorem 5

Lemma 4. *Assume that f is convex and Assumptions 1 and 2 hold. Let T be the number of iterations and define $\tau \coloneqq \arg\min_{0\le t\le T-1} f(\boldsymbol{x}_t)$. If $f(\boldsymbol{x}_t) - f^\star \ge \frac{\sigma^2}{\sqrt{T}}$ for all t, then gradient descent with stepsize Eq. (14) satisfies:*

$$f(\boldsymbol{x}_\tau) - f^\star \le \frac{8L_0\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 + 2\sigma^2}{\sqrt{T}} + \frac{128L_1^2L\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^4}{T} + \frac{8L_1^2\sigma^4 L}{L_0^2 T}.$$

where $\boldsymbol{x}^\star \coloneqq \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$ and $\sigma^2 \coloneqq f^\star - l^\star$.

Proof. By the convexity of f, we have

$$\begin{aligned}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 &= \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - 2\eta_t\langle\nabla f(\boldsymbol{x}_t), \boldsymbol{x}_t - \boldsymbol{x}^\star\rangle + \eta_t^2\|\nabla f(\boldsymbol{x}_t)\|^2\\ &\le \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - 2\eta_t(f(\boldsymbol{x}_t) - f^\star) + \eta_t^2\|\nabla f(\boldsymbol{x}_t)\|^2.\end{aligned}$$

Substituting the stepsize Eq. (14), we get

$$\begin{aligned}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 &\le \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - 2\eta_t(f(\boldsymbol{x}_t) - f^\star) + \frac{\eta_t}{\sqrt{T}}(f(\boldsymbol{x}_t) - l^\star)\\ &\le \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \eta_t(2 - \frac{1}{\sqrt{T}})(f(\boldsymbol{x}_t) - f^\star) + \frac{\eta_t\sigma^2}{\sqrt{T}}\\ &\le \|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \eta_t(f(\boldsymbol{x}_t) - f^\star) + \frac{\eta_t\sigma^2}{\sqrt{T}},\end{aligned} \tag{18}$$

where we use $T \ge 1$ in the last inequality. Unrolling the above inequality and dividing by η_t, we obtain

$$f(\boldsymbol{x}_t) - f^\star \le \frac{\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2}{\eta_t} + \frac{\sigma^2}{\sqrt{T}}. \tag{19}$$

Case when $\|\nabla f(\boldsymbol{x}_t)\| \le \frac{L_0}{L_1}$: From $f(\boldsymbol{x}_t) - f^\star \ge \frac{\sigma^2}{\sqrt{T}}$ and Eq. (18), we obtain

$$\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2 \ge 0. \tag{20}$$

Thus, we get

$$\begin{aligned}f(\boldsymbol{x}_t) - f^\star &\le 2(L_0 + L_1\|\nabla f(\boldsymbol{x}_t)\|)\sqrt{T}(\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2) + \frac{\sigma^2}{\sqrt{T}}\\ &\le 4L_0\sqrt{T}(\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2) + \frac{\sigma^2}{\sqrt{T}},\end{aligned}$$

where we use $f^\star \ge l^\star$ and Lemma 2 for the first inequality and use $\|\nabla f(\boldsymbol{x}_t)\| \le \frac{L_0}{L_1}$ for the last inequality.

Case when $\|\nabla f(\boldsymbol{x}_t)\| > \frac{L_0}{L_1}$: From Lemma 2, we have

$$\eta_t \ge \frac{f(\boldsymbol{x}_t) - f^\star}{\sqrt{T}\|\nabla f(\boldsymbol{x}_t)\|^2} \ge \frac{1}{2(L_0 + L_1\|\nabla f(\boldsymbol{x}_t)\|)\sqrt{T}}.$$

Then, we obtain

$$\eta_t \ge \frac{1}{4L_1\|\nabla f(\boldsymbol{x}_t)\|\sqrt{T}} \ge \frac{1}{4L_1\sqrt{2LT(f(\boldsymbol{x}_t) - f^\star)}},$$

where we use $\|\nabla f(\boldsymbol{x}_t)\| > \frac{L_0}{L_1}$ for the first inequality, and Lemma 1 for the last inequality. Combining Eqs. (19) and (20), we obtain

$$f(\boldsymbol{x}_t) - f^\star \le 4L_1\sqrt{2LT(f(\boldsymbol{x}_t) - f^\star)}(\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2) + \frac{\sigma^2}{\sqrt{T}}.$$

Furthermore, from $\|\nabla f(\boldsymbol{x}_t)\| > \frac{L_0}{L_1}$ and Lemma 1, we obtain

$$\sqrt{f(\boldsymbol{x}_t) - f^\star} \geq \frac{L_0}{L_1}\sqrt{\frac{f(\boldsymbol{x}_t) - f^\star}{\|\nabla f(\boldsymbol{x}_t)\|^2}} \geq \frac{L_0}{L_1}\sqrt{\frac{1}{2L}}.$$

Thus, we get

$$\begin{aligned}
&f(\boldsymbol{x}_t) - f^\star \\
&\leq 4L_1\sqrt{2LT(f(\boldsymbol{x}_t) - f^\star)}(\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2) + \frac{L_1\sigma^2}{L_0\sqrt{T}}\sqrt{2L(f(\boldsymbol{x}_t) - f^\star)}.
\end{aligned}$$

Dividing by $\frac{L_1\sqrt{2L(f(\boldsymbol{x}_t)-f^\star)}}{L_0}$, we get

$$\frac{L_0}{L_1}\sqrt{\frac{f(\boldsymbol{x}_t) - f^\star}{2L}} \leq 4L_0\sqrt{T}(\|\boldsymbol{x}_t - \boldsymbol{x}^\star\|^2 - \|\boldsymbol{x}_{t+1} - \boldsymbol{x}^\star\|^2) + \frac{\sigma^2}{\sqrt{T}}.$$

Summing the two cases: Define $\mathcal{T}_1$ and $\mathcal{T}_2$ as follows:

$$\mathcal{T}_1 \coloneqq \left\{t \middle| \|\nabla f(\boldsymbol{x}_t)\| \leq \frac{L_0}{L_1}\right\}, \quad \mathcal{T}_2 \coloneqq \left\{t \middle| \|\nabla f(\boldsymbol{x}_t)\| > \frac{L_0}{L_1}\right\}.$$

We obtain

$$\frac{1}{T}\left(\sum_{t\in\mathcal{T}_1}(f(\boldsymbol{x}_t) - f^\star) + \frac{L_0}{L_1}\sum_{t\in\mathcal{T}_2}\sqrt{\frac{f(\boldsymbol{x}) - f^\star}{2L}}\right) \leq \frac{4L_0\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 + \sigma^2}{\sqrt{T}}.$$

The above inequality implies that

$$\begin{aligned}
\frac{1}{T}\sum_{t\in\mathcal{T}_1}(f(\boldsymbol{x}_t) - f^\star) &\leq \frac{4L_0\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 + \sigma^2}{\sqrt{T}}, \\
\frac{1}{T}\sum_{t\in\mathcal{T}_2}\sqrt{f(\boldsymbol{x}) - f^\star} &\leq \frac{4L_1\sqrt{2L}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2}{\sqrt{T}} + \frac{L_1\sigma^2\sqrt{2L}}{L_0\sqrt{T}}.
\end{aligned}$$

Using $a^2 \geq 2ab - b^2$, we obtain for any $b \in \mathbb{R}$

$$\frac{1}{T}\sum_{t\in\mathcal{T}_1}\left(2b\sqrt{f(\boldsymbol{x}_t) - f^\star} - b^2\right) \leq \frac{4L_0\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 + \sigma^2}{\sqrt{T}}.$$

Thus, when $b > 0$, we obtain

$$\frac{1}{T}\sum_{t\in\mathcal{T}_1}\sqrt{f(\boldsymbol{x}_t) - f^\star} \leq \frac{4L_0\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 + \sigma^2}{2b\sqrt{T}} + \frac{b}{2}.$$

Choosing $b = \sqrt{\frac{4L_0\|\boldsymbol{x}_0-\boldsymbol{x}^\star\|^2+\sigma^2}{\sqrt{T}}}$, we get

$$\frac{1}{T}\sum_{t\in\mathcal{T}_1}\sqrt{f(\boldsymbol{x}_t) - f^\star} \leq \sqrt{\frac{4L_0\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 + \sigma^2}{\sqrt{T}}}.$$

Thus, we get

$$\frac{1}{T}\sum_{t=0}^{T-1}\sqrt{f(\boldsymbol{x}_t) - f^\star} \leq \sqrt{\frac{4L_0\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 + \sigma^2}{\sqrt{T}}} + \frac{4L_1\sqrt{2L}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2}{\sqrt{T}} + \frac{L_1\sigma^2\sqrt{2L}}{L_0\sqrt{T}}.$$

Defining $\tau \coloneqq \arg\min_t f(\boldsymbol{x}_t)$, we get

$$\sqrt{f(\boldsymbol{x}_\tau) - f^\star} \leq \sqrt{\frac{4L_0\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 + \sigma^2}{\sqrt{T}}} + \frac{4L_1\sqrt{2L}\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2}{\sqrt{T}} + \frac{L_1\sigma^2\sqrt{2L}}{L_0\sqrt{T}}.$$

Squaring the both sides, and using $(a+b)^2 \leq 2a^2 + 2b^2$ for all $a, b \in \mathbb{R}$, we obtain

$$f(\boldsymbol{x}_\tau) - f^\star \leq \frac{8L_0\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 + 2\sigma^2}{\sqrt{T}} + \frac{128L_1^2L\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^4}{T} + \frac{8L_1^2\sigma^4L}{L_0^2T}.$$

This concludes the statement. □

https://doi.org/10.52202/079017-1416

Lemma 5. *Assume that f is convex and Assumptions 1 and 2 hold. Let T be the number of iterations and define $\tau \coloneqq \arg\min_{0\leq t\leq T-1} f(\boldsymbol{x}_t)$. Then, gradient descent with stepsize Eq. (14) satisfies:*

$$f(\boldsymbol{x}_\tau) - f(\boldsymbol{x}^\star) \leq \mathcal{O}\left(\frac{L_0\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 + \sigma^2}{\sqrt{T}} + \frac{LL_1^2\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^4}{T} + \frac{L_1^2 L\sigma^4}{L_0^2 T}\right), \tag{21}$$

where $\boldsymbol{x}^\star \coloneqq \arg\min_{\boldsymbol{x}} f(\boldsymbol{x})$ and $\sigma^2 \coloneqq f^\star - l^\star$.

Proof. If there exists t such that $f(\boldsymbol{x}_t) - f^\star < \frac{\sigma^2}{\sqrt{T}}$, we have

$$f(\boldsymbol{x}_\tau) - f^\star \leq f(\boldsymbol{x}_t) - f^\star < \frac{\sigma^2}{\sqrt{T}}.$$

Then, if $f(\boldsymbol{x}_t) - f^\star \geq \frac{\sigma^2}{\sqrt{T}}$ for all t, Lemma 4 shows that

$$f(\boldsymbol{x}_\tau) - f^\star \leq \frac{8L_0\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^2 + 2\sigma^2}{\sqrt{T}} + \frac{128L_1^2 L\|\boldsymbol{x}_0 - \boldsymbol{x}^\star\|^4}{T} + \frac{8L_1^2\sigma^4 L}{L_0^2 T}.$$

By combining the above two cases, we have the desired statement. □

C Additional theoretical result

Lemma 6. *Let f be a function such that $\|\nabla^2 f(\boldsymbol{x})\| \leq L_0 + L_1\|\nabla f(\boldsymbol{x})\|$ holds for any $\boldsymbol{x}$. For any $\boldsymbol{x}, \boldsymbol{y}$ such that $\|\boldsymbol{x} - \boldsymbol{y}\| \leq \frac{1}{L_0}$, we have*

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq 2(L_0 + L_1\|\nabla f(\boldsymbol{x})\|)\|\boldsymbol{x} - \boldsymbol{y}\|.$$

Proof. See Lemma A.2 in (Zhang et al., 2020a). □

Proposition 3. *For any $L_0 \geq 0$ and $L_1 \geq 0$, $f(x) \coloneqq \frac{L_0 L_1^2}{72}x^4 + \frac{L_0}{4}x^2$ is (L_0, L_1)-smooth.*

Proof. Since $f(x)$ is twice differentiable, we have

$$|\nabla^2 f(x)| = \frac{L_0 L_1^2}{6}x^2 + \frac{L_0}{2}.$$

Using $\frac{L_1}{6}x^2 + \frac{3}{2L_1} \geq |x|$, we obtain

$$\begin{aligned}
|\nabla^2 f(x)| &\leq \frac{L_0 L_1^2}{6}\left(\frac{L_1}{6}x^2 + \frac{3}{2L_1}\right)|x| + L_0 \\
&= \frac{L_1}{2}\left|\frac{L_0 L_1^2}{18}x^3 + \frac{L_0}{2}x\right| + \frac{L_0}{2} \\
&= \frac{L_1}{2}|\nabla f(x)| + \frac{L_0}{2}.
\end{aligned}$$

From Lemma 6, we have the desired statement. □

D Hyperparameter settings

D.1 Synthetic function

In our experiments, we ran the clipped gradient descent with the following hyperparameters and tuned the hyperparameters by grid search.

Table 2: Hyperparameter settings for clipped gradient descent.

Learning Rate	$\{1, 1.0 \times 10^{-1}, \cdots, 1.0 \times 10^{-8}\}$
Gradient Clipping Threshold	$\{0.01, 0.1, 1, 5, 10, 15, 20, \infty\}$

Table 3: Hyperparameters selected by grid search.

	Gradient Descent	**Clipped Gradient Descent**	
	Learning Rate	Learning Rate	Gradient Clipping Threshold
$L_1 = 1$	1.0×10^{-1}	0.1	20
$L_1 = 10$	1.0×10^{-3}	0.1	10
$L_1 = 100$	1.0×10^{-5}	0.1	10
$L_1 = 1000$	1.0×10^{-7}	0.1	10

D.2 Neural networks

In our experiments, we used the following training configuration:

- **LSTM:** `https://github.com/salesforce/awd-lstm-lm`
- **Nano-GPT:** `https://github.com/karpathy/nanoGPT`
- **T5:** `https://github.com/PiotrNawrot/nanoT5`

We ran all experiments on an A100 GPU. For Clipped SGD and SGD, we tuned the stepsize and gradient clipping threshold using the grid search. See Tables 4, 5, and 6 for detailed hyperparameter settings, and see Table 7 for the selected hyperparameters.

Table 4: Hyperparameter settings for LSTM.

Learning Rate	$\{100, 50, 10, 1, 0.1, 0.01\}$
Gradient Clipping Threshold	$\{0.5, 1, \cdots, 4.5, 5, \infty\}$
Batch Size	80

Table 5: Hyperparameter settings for Nano-GPT.

Learning Rate	$\{1, 0.5, 0.1, \cdots, 0.0005, 0.0001\}$
Gradient Clipping Threshold	$\{1, 2, \cdots, 9, 10, \infty\}$
Batch Size	64

Table 6: Hyperparameter settings for T5.

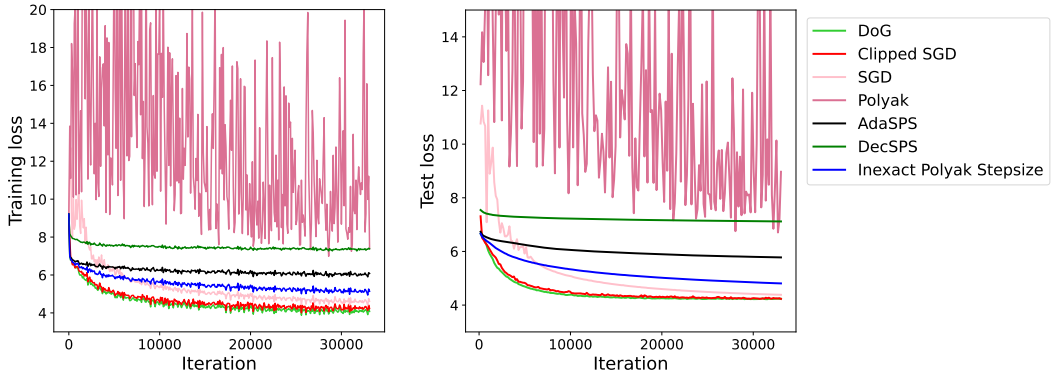
Learning Rate	$\{5.0, 1.0, 0.5, 0.1, 0.05\}$
Gradient Clipping Threshold	$\{1, 2, 3, \infty\}$
Batch Size	128

Table 7: Hyperparameters selected by grid search. Three values correspond to the selected hyperparameters for different seed values.

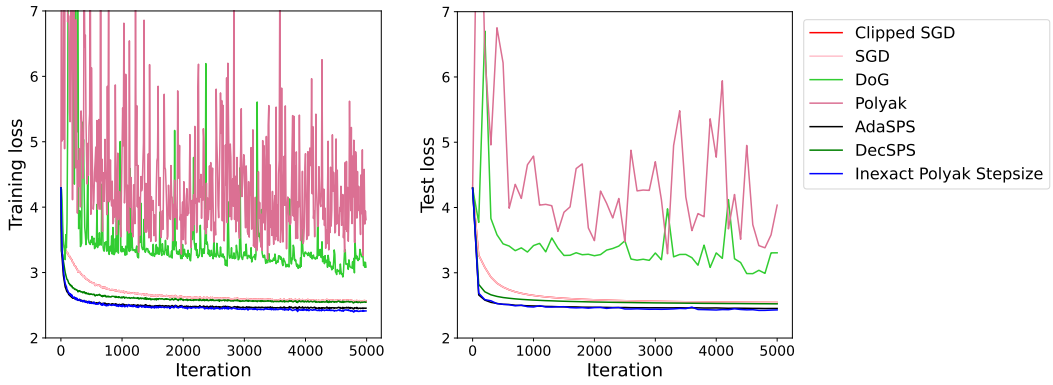
	Gradient Descent	**Clipped Gradient Descent**	
	Learning Rate	Learning Rate	Gradient Clipping Threshold
LSTM	10 / 10 / 10	10 / 50 / 50	0.5 / 1 / 0.5
Nano-GPT	0.001 / 0.001 / 0.001	0.001 / 0.001 / 0.001	∞ / 10 / 10
T5	0.1 / 0.1 / 0.05	1 / 1 / 1	2 / 2 / 2

https://doi.org/10.52202/079017-1416

E Additional numerical evaluation



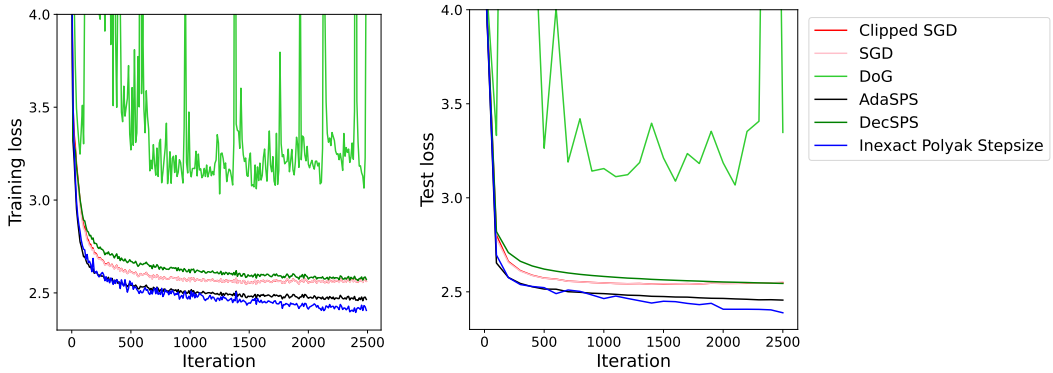
(a) LSTM



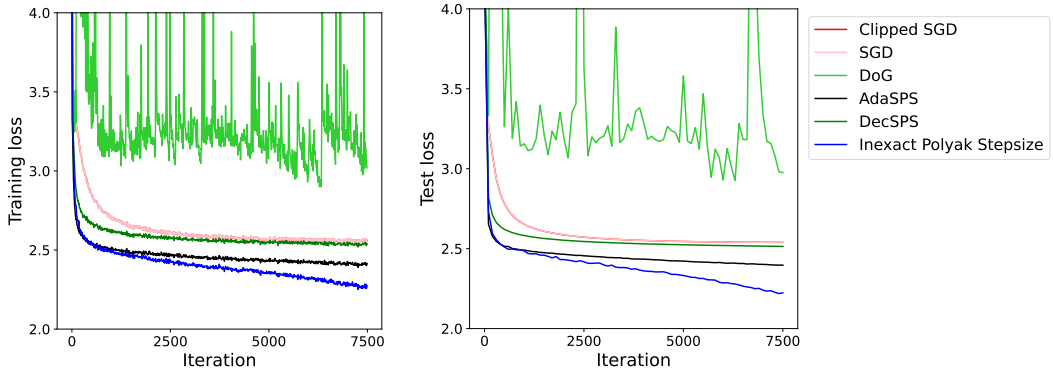
(b) Nano-GPT

Figure 4: Loss curves for LSTM and Nano-GPT. We plotted the training loss per 100, 10, and 10 iterations for LSTM, Nano-GPT, and T5, respectively. We plotted the test loss per one epoch, 100 iterations, and 200 iterations, respectively.

https://doi.org/10.52202/079017-1416



(a) $T = 2500$



(b) $T = 7500$

Figure 5: Loss curves for Nano-GPT with different T.

https://doi.org/10.52202/079017-1416

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main claims are clearly discussed in Sec. 1.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Sec. 6.2.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All proofs are provided in Sec. A and B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All training configuration and hyperparameter setting are provided in Sec. D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

https://doi.org/10.52202/079017-1416

Answer: [Yes]

Justification: Our code is contained in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Our training configuration is provided in Sec. D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We repeated the experiments in Sec. 6.2 with three different seed values and reported the average, while we did not report error bars to make the figure easy to read.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Sec. D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors read and complied with the code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The motivation and its impact of our study are clearly discussed in Sec. 1.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

https://doi.org/10.52202/079017-1416

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our study does not provide any new dataset or pre-trained models.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: See Sec. D.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our code is provided in the supplementary material with MIT license.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our experiments do not use any crowdsourcing service.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our experiments do not use any crowdsourcing service.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-1416