Proceedings of Machine Learning Research vol 195:1–21, 2023 36th Annual Conference on Learning Theory

On the Lower Bound of Minimizing Polyak-Łojasiewicz Functions

Pengyun Yue YUEPY@PKU.EDU.CN
National Key Lab of General AI, School of Intelligence Science and Technology, Peking University

Cong Fang✉ FANGCONG@PKU.EDU.CN
National Key Lab of General AI, School of Intelligence Science and Technology, Peking University
Institute for Artificial Intelligence, Peking University

Zhouchen Lin✉ ZLIN@PKU.EDU.CN
National Key Lab of General AI, School of Intelligence Science and Technology, Peking University
Institute for Artificial Intelligence, Peking University
Peng Cheng Laboratory

Editors: Gergely Neu and Lorenzo Rosasco

Abstract

Polyak-Łojasiewicz (PL) (Polyak, 1963) condition is a weaker condition than the strong convexity but suffices to ensure a global convergence for the Gradient Descent algorithm. In this paper, we study the lower bound of algorithms using first-order oracles to find an approximate optimal solution. We show that any first-order algorithm requires at least $\Omega\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$ gradient costs to find an ε-approximate optimal solution for a general L-smooth function that has an μ-PL constant. This result demonstrates the *optimality* of the Gradient Descent algorithm to minimize smooth PL functions in the sense that there exists a "hard" PL function such that no first-order algorithm can be faster than Gradient Descent when ignoring a numerical constant. In contrast, it is well-known that the momentum technique, e.g. Nesterov (2003, chap. 2), can provably accelerate Gradient Descent to $O\left(\sqrt{\frac{L}{\hat{\mu}}}\log\frac{1}{\varepsilon}\right)$ gradient costs for functions that are L-smooth and $\hat{\mu}$-strongly convex. Therefore, our result distinguishes the hardness of minimizing a smooth PL function and a smooth strongly convex function as the complexity of the former cannot be improved by any polynomial order in general.

Keywords: Polyak-Łojasiewicz condition, first-order optimization, complexity

1. Introduction

We consider the problem

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

where the function f is L-smooth and satisfies the Polyak-Łojasiewicz condition. A function f is said to satisfy the Polyak-Łojasiewicz condition if (2) holds for some $\mu > 0$:

$$\|\nabla f(\mathbf{x})\|^2 \geq 2\mu\left(f(\mathbf{x}) - \inf_{\mathbf{y}\in\mathbb{R}^d} f(\mathbf{y})\right), \quad \forall \mathbf{x}\in\mathbb{R}^d. \tag{2}$$

We refer to (2) as the μ-PL condition and simply denote $\inf_{\mathbf{y}\in\mathbb{R}^d} f(\mathbf{y})$ by f^*. The PL condition may be originally introduced by Polyak (Polyak, 1963) and Łojasiewicz (Lojasiewicz, 1963) independently. The PL condition is strictly weaker than strong convexity as one can show that any $\hat{\mu}$-strongly convex function which by definition satisfies:

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle + \frac{\hat{\mu}}{2} \|\mathbf{x} - \mathbf{y}\|^2$$

is also $\hat{\mu}$-PL by minimizing both sides with respect to $\mathbf{x}$ (Karimi et al., 2016). However, the PL condition does not even imply convexity. From a geometric view, the PL condition suggests that the sum of the squares of the gradient dominates the optimal function value gap, which implies that any local stationary point is a global minimizer. Because it is relatively easy to obtain an approximate local stationary point by first-order algorithms, the PL condition serves as an ideal and weaker alternative to strong convexity.

In machine learning, the PL condition has received wide attention recently. Lots of models are found to satisfy this condition under different regimes. Examples include, but are not limited to, matrix decomposition and linear neural networks under a specific initialization (Hardt and Ma, 2016; Li et al., 2018), nonlinear neural networks in the so-called neural tangent kernel regime (Liu et al., 2022), reinforcement learning with linear quadratic regulator (Fazel et al., 2018). Compared with strong convexity, the PL condition is much easier to hold since the reference point in the latter only is a minimum point such that $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{y}} f(\mathbf{y})$, instead of any $\mathbf{y}$ in the domain.

Turning to the theoretic side, it is known (Karimi et al., 2016) that the standard Gradient Descent algorithm admits a linear converge to minimize a L-smooth and μ-PL function. To be specific, in order to find an ε-approximate optimal solution $\hat{\mathbf{x}}$ such that $f(\hat{\mathbf{x}}) - f^* \leq \varepsilon$, Gradient Decent needs $O(\frac{L}{\mu} \log \frac{1}{\varepsilon})$ gradient computations. However, it is still not clear whether there exist algorithms that can achieve a provably faster convergence rate. In the optimization community, it is perhaps well-known that the momentum technique, e.g. Nesterov (2003, chap. 2), can provably accelerate Gradient Descent from $O(\frac{L}{\hat{\mu}} \log \frac{1}{\varepsilon})$ to $O\left(\sqrt{\frac{L}{\hat{\mu}}} \log \frac{1}{\varepsilon}\right)$ for functions that are L-smooth and $\hat{\mu}$-strongly convex. Even though some works (J Reddi et al., 2016; Lei et al., 2017) have considered accelerations under different settings, probably faster convergence of first-order algorithms for PL functions is still not obtained up to now.

In this paper, we study the first-order complexities to minimize a generic smooth PL function and ask the question:

“Is the Gradient Decent algorithm (nearly) optimal or can we design a much faster algorithm?”

We answer the question in the language of min-max lower bound complexity for minimizing the L-smooth and μ-PL function class. We analyze the worst complexity of minimizing any function that belongs to the class using first-order algorithms. Excitingly, we construct a hard instance function showing that any first-order algorithm requires at least $\Omega\left(\frac{L}{\mu} \log \frac{1}{\varepsilon}\right)$ gradient costs to find an ε-approximate optimal solution. This answers the aforementioned question in an explicit way: the Gradient Descent algorithm is already *optimal* in the sense that no first-order algorithm can achieve a provably faster convergence rate in general ignoring a numerical constant. For the first time, we distinguish the hardness of minimizing a PL function and a strongly convex function in terms of first-order complexities, as the momentum technique for smooth and strongly convex functions provably accelerates Gradient Descent by a certain polynomial order.

It is worth mentioning that the optimization problem under our consideration is high-dimensional and the goal is to obtain the complexity bounds that do not have an explicit dependency on the dimension.

Our technique to establish the lower bound follows from the previous lower bounds in convex (Nesterov, 2003) and non-convex optimization (Carmon et al., 2021). The main idea is to construct a so-called "zero-chain" function ensuring that any first-order algorithm per-iteratively can only solve one coordinate of the optimization variable. Then for a "zero-chain" function that has a sufficiently high dimension, some number of entries will never reach their optimal values after the execution of any first-order algorithm in certain iterations. To obtain the desired $\Omega\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$ lower bound, we propose a "zero-chain" function similar to Carmon et al. (2020), which is composed of the worst convex function designed by Nesterov (2003) and a separable function in the form as $\sum_{i=1}^{T} v_{\mathbf{y}_i}(\mathbf{x}_i)$ to destroy the convexity. Different from their separable function, the one that we introduce has a large Lipshictz constant. This property helps us to estimate the PL constant in a convenient way. This new idea gives new insights into the constructions and analyses of instance functions, which might be potentially generalized to establish the lower bounds for other non-convex problems.

Notation

We use bold letters, such as $\mathbf{x}$, to denote vectors in the Euclidean space $\mathbb{R}^d$, and bold capital letters, such as $\mathbf{A}$, to denote matrices. $\mathbf{I}_d$ denotes the identity matrix of size $d \times d$. We omit the subscript and simply denote $\mathbf{I}$ as the identity matrix when the dimension is clear from context. For $\mathbf{x} \in \mathbb{R}^d$, we use $\mathbf{x}_i$ to denote its ith coordinate. We use $\mathrm{supp}(\mathbf{x})$ to denote the subscripts of non-zero entries of $\mathbf{x}$, i.e. $\mathrm{supp}(\mathbf{x}) = \{i : \mathbf{x}_i \neq 0\}$. We use $\mathrm{span}\left\{\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(n)}\right\}$ to denote the linear subspace spanned by $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(n)}$, i.e. $\left\{\mathbf{y} : \mathbf{y} = \sum_{i=1}^{n} a_i \mathbf{x}^{(i)}, a_i \in \mathbb{R}\right\}$. We call a function f L-smooth if ∇f is L-Lipschitz continuous, i.e. $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|$. We denote $f^* = \inf_{\mathbf{x}} f(\mathbf{x})$. We let $\mathbf{x}^*$ be any minimizer of f, i.e., $\mathbf{x}^* = \mathrm{argmin}\, f$. We always assume the existence of $\mathbf{x}^*$. We say that $\mathbf{x}$ is an ε-approximate optimal point of f when $f(\mathbf{x}) - f^* \leq \varepsilon$.

2. Related Work

Lower Bounds There has been a line of research concerning the lower bounds of algorithms on certain function classes. To the best of our knowledge, (Nemirovskij and Yudin, 1983) defines the oracle model to measure the complexity of algorithms, and most existing research on lower bounds follow this formulation of complexity. For convex functions and first-order oracles, the lower bound is studied in Nesterov (2003), where well-known optimal lower bound $\Omega(\varepsilon^{-\frac{1}{2}})$ and $\Omega(\kappa \log \frac{1}{\varepsilon})$ are obtained. For convex functions and nth-order oracles, lower bounds $\Omega\left(\varepsilon^{-\frac{2}{3n+1}}\right)$ have been proposed in Arjevani et al. (2019b). When the function is non-convex, it is generally NP-hard to find its global minima, or to test whether a point is a local minimum or a saddle point (Murty and Kabadi, 1985). Instead of finding ε-approximate optimal points, an alternative measure is finding ε-stationary points where $\|\nabla f(\mathbf{x})\| \leq \varepsilon$. Sometimes, additional constraints on the Hessian matrices of second-order stationary points are needed. Results of this kind include Carmon et al. (2020, 2021); Fang et al. (2018); Zhou and Gu (2019); Arjevani et al. (2019a, 2020). Though a PL function may be non-convex, it is tractable to find an ε-approximate optimal point, as local minima of a PL function must be global minima. In this paper, we give the lower complexity bound for finding ε-approximate optimal points.

PL Condition The PL condition was introduced by Polyak (Polyak, 1963) and Łojasiewicz (Lojasiewicz, 1963) independently. Besides the PL condition, there are other relaxations of the strong convexity, including error bounds (Luo and Tseng, 1993), essential strong convexity (Liu et al., 2014), weak strong convexity (Necoara et al., 2019), restricted secant inequality (Zhang and Yin, 2013), and quadratic growth (Anitescu, 2000). Karimi et al. (2016) discussed the relationships between these conditions. All these relaxations implies the PL condition except for the quadratic growth, which implies that the PL condition is quite general. Danilova et al. (2020) studied the convergence rate of Heavy-ball method on PL functions. Wang et al. (2022) proved an accelerated convergence rate for Heavy-ball algorithm when the non-convexity is "averaged-out". There are many other papers that study designing practical algorithms to optimize a PL objective function under different scenarios, for example, Bassily et al. (2018); Nouiehed et al. (2019); Hardt and Ma (2016); Fazel et al. (2018); J Reddi et al. (2016); Lei et al. (2017).

3. Preliminaries

3.1. Upper bound on PL functions

Although the PL condition is a weaker condition than strong convexity, it guarantees linear convergence for Gradient Descent. The result can be found in Polyak (1963) and Karimi et al. (2016). We present it here for completeness.

Theorem 1 *If f is L-smooth and satisfies μ-PL condition, then the Gradient Descent algorithm with a constant step-size $\frac{1}{L}$:*

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \frac{1}{L}\nabla f(\mathbf{x}^{(k)}), \tag{3}$$

has a linear convergence rate. We have:

$$f(\mathbf{x}^{(k)}) - f^* \leq \left(1 - \frac{\mu}{L}\right)^k (f(\mathbf{x}^{(0)}) - f^*). \tag{4}$$

Theorem 1 shows that the Gradient Descent algorithm finds the ε-approximate optimal point of f in $O\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$ gradient computations. This gives an upper complexity bound for first-order algorithms. However, it remains open to us whether there are faster algorithms for smooth PL functions. We will establish a lower complexity bound on first-order algorithms, which nearly matches the upper bound.

3.2. Definitions of algorithm classes and function classes

An algorithm is a mapping from real-valued functions to sequences. For algorithm A and $f : \mathbb{R}^d \to \mathbb{R}$, we define $\mathsf{A}[f] = \{\mathbf{x}^{(i)}\}_{i\in\mathbb{N}}$ to be the sequence of algorithm A acting on f, where $\mathbf{x}^{(i)} \in \mathbb{R}^d$.

Note here, the algorithm under our consideration works on function defined on any Euclidean space. We call it the dimension-free property of the algorithm.

The definition of algorithms abstracts away from the the optimization process of a function. We consider algorithms which only make use of the first-order information of the iteration sequence. We call them first-order algorithms. If an algorithm is a first-order algorithm, then

$$\mathbf{x}^{(i)} = \mathsf{A}^{(i)}\left(\mathbf{x}^{(0)}, \nabla f(\mathbf{x}^{(0)}), \cdots, \mathbf{x}^{(i-1)}, \nabla f(\mathbf{x}^{(i-1)})\right), \tag{5}$$

where $\mathsf{A}^{(i)}$ is a function depending on A. Perhaps the simplest example of first-order algorithms is Gradient Descent.

We are interested in finding an ε-approximate point of a function f. Given a function $f : \mathbb{R}^d \to \mathbb{R}$ and an algorithm A, the complexity of A on f is the number of queries to the first-order oracle needed to find an ε-approximate point. We denote $T_\varepsilon(\mathsf{A}, f)$ to be the gradient complexity of A on f, then

$$T_\varepsilon(\mathsf{A}, f) = \min_t \left\{ t : f(\mathbf{x}^{(t)}) - f^* \leq \varepsilon \right\}. \tag{6}$$

In practice, we do not have the full information of the function f. We only know that f is in a particular function class $\mathcal{F}$, such as L-smooth functions. Given an algorithm A. We denote $\mathsf{T}_\varepsilon(\mathsf{A}, \mathcal{F})$ to be the complexity of A on $\mathcal{F}$, and define $\mathsf{T}_\varepsilon(\mathsf{A}, \mathcal{F})$ as follows:

$$\mathsf{T}_\varepsilon(\mathsf{A}, \mathcal{F}) = \sup_{f \in \mathcal{F}} T_\varepsilon(\mathsf{A}, f). \tag{7}$$

Thus, $\mathsf{T}_\varepsilon(\mathsf{A}, \mathcal{F})$ is the worst-case complexity of functions $f \in \mathcal{F}$.

For searching an ε-approximate optimal point of a function in $\mathcal{F}$, we need to find an algorithm which have a low complexity on $\mathcal{F}$. Denote an algorithm class by $\mathcal{A}$. The lower bound of an algorithm class on $\mathcal{F}$ describes the efficiency of algorithm class $\mathcal{A}$ on function class $\mathcal{F}$, which is defined to be

$$\mathcal{T}_\varepsilon(\mathcal{A}, \mathcal{F}) = \inf_{\mathsf{A} \in \mathcal{A}} \mathsf{T}_\varepsilon(\mathsf{A}, \mathcal{F}) = \inf_{\mathsf{A} \in \mathcal{A}} \sup_{f \in \mathcal{F}} T_\varepsilon(\mathsf{A}, f). \tag{8}$$

3.3. Zero-respecting Algorithm

Among all the algorithms, a special algorithm class is called zero-respecting algorithms. If A is a zero-respecting algorithm and $\mathsf{A}[f] = \left\{\mathbf{x}^{(t)}\right\}_{t \in \mathbb{N}}$, then the following condition holds for all $f : \mathbb{R}^d \to \mathbb{R}$:

$$\operatorname{supp}\{\mathbf{x}^{(n)} - \mathbf{x}^{(0)}\} \in \bigcup_{i=1}^{n-1} \operatorname{supp}\{\nabla f(\mathbf{x}^{(i)})\}. \tag{9}$$

Note that if $\mathbf{x}^{(n)} - \mathbf{x}^{(0)}$ lies in the linear subspace spanned by $\nabla f(\mathbf{x}^{(0)}), \cdots, \nabla f(\mathbf{x}^{(n-1)})$, then A is a zero-respecting algotithm. We denote the collection of first-order zero-respecting algorithms with $\mathbf{x}^{(0)} = \mathbf{0}$ by $\mathcal{A}_{\text{zr}}$. It is shown by Nemirovskij and Yudin (1983) that a lower complexity bound on first-order zero-respecting algorithms are also a lower complexity bound on all the first-order algorithm when the function class satisfies the orthogonal invariance property.

3.4. Zero-chain

A zero-chain f is a function that safisfies the following condition:

$$\operatorname{supp}(\mathbf{x}) \subseteq \{1, 2, \cdots, k\} \implies \operatorname{supp}(\nabla f(\mathbf{x})) \subseteq \{1, 2, \cdots, k+1\}, \quad \forall \mathbf{x}. \tag{10}$$

In other words, the support of $\nabla f(\mathbf{x})$ lies in a restricted linear subspace depending on the support of $\mathbf{x}$.

The "worst function in the (convex) world" in Nesterov (2003) defined as

$$f_d(\mathbf{x}) = \frac{1}{2}(\mathbf{x}_1 - 1)^2 + \sum_{i=1}^{d-1} (\mathbf{x}_{i+1} - \mathbf{x}_i)^2 \tag{11}$$

is a zero-chain, because if $\mathbf{x}_i = 0$ for $i > n$, then $(\nabla f_d(\mathbf{x}))_{i+1} = 0$ for $i > n$. A zero-chain is difficult to optimize for zero-respecting algorithms, because zero-respecting algorithms only discover one coordinate by one gradient computation.

4. Main results

According to Theorem 1, we already have an upper complexity bound $O\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$ by applying Gradient Descent to all the PL functions. In this section, we establish the lower complexity bound of first-order algorithms on PL functions. Let $\mathcal{P}(\Delta, \mu, L)$ be the collection of all L-smooth and μ-PL functions f with $f(\mathbf{x}^{(0)}) - f^* \le \Delta$. We establish a lower bound of $\mathcal{T}_\varepsilon\left(\mathcal{A}_{\mathrm{zr}}, \mathcal{P}(\Delta, \mu, L)\right)$ by constructing a function which is hard to optimize for zero-respecting algorithms, and extend the result to first-order algorithms. We present a hard instance that can achieve the desired $\Omega\left(\kappa \log\frac{1}{\varepsilon}\right)$ lower bound below.

We first introduce several components of the hard instance. For the non-convex part, we define

$$v_y(x) = \begin{cases} \frac{1}{2}x^2, & x \le \frac{31}{32}y, \\ \frac{1}{2}x^2 - 16\left(x - \frac{31}{32}y\right)^2, & \frac{31}{32}y < x \le y, \\ \frac{1}{2}x^2 - \frac{y^2}{32} + 16\left(x - \frac{33}{32}y\right)^2, & y < x \le \frac{33}{32}y, \\ \frac{1}{2}x^2 - \frac{y^2}{32}, & x > \frac{33}{32}y, \end{cases} \tag{12}$$

where $y > 0$ is a constant. By the definition of v_y, we have

$$v_y'(x) = \begin{cases} x, & x \le \frac{31}{32}y, \\ x - 32\left(x - \frac{31}{32}y\right), & \frac{31}{32}y < x \le y, \\ x + 32\left(x - \frac{33}{32}y\right), & y < x \le \frac{33}{32}y, \\ x, & x > \frac{33}{32}y. \end{cases} \tag{13}$$

Define

$$b_y(x) = \begin{cases} y - 32|x-y|, & \frac{31}{32}y \le x \le \frac{33}{32}y, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

Then we have $v_y'(x) = x - b_y(x)$.

For the convex part, we define $q_{T,t}(\mathbf{x})$ as follows (for the convenience of notation, we define $\mathbf{x}_0 = 0$):

$$q_{T,t}(\mathbf{x}) = \frac{1}{2}\sum_{i=0}^{t-1}\left[\left(\frac{7}{8}\mathbf{x}_{iT} - \mathbf{x}_{iT+1}\right)^2 + \sum_{j=1}^{T-1}\left(\mathbf{x}_{iT+j+1} - \mathbf{x}_{iT+j}\right)^2\right], \tag{15}$$

where $\mathbf{x} \in \mathbb{R}^{Tt}$. $q_{T,t}$ is a quadratic function of $\mathbf{x}$, thus can be written as

$$q_{T,t}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{B}\mathbf{x}, \tag{16}$$

where $\mathbf{B}$ is a positive semi-definite symmetric matrix. $\mathbf{B}$ satisfies $0 \preceq \mathbf{B} \preceq 4\mathbf{I}$, because the sum of absolute value of non-zero entries of each row of $\mathbf{B}$ is smaller or equal to 4.

The quadratic part q is very similar to "the worst function in the (convex) world" in Nesterov (2003), and the definition of v_y is inspired by the hard instance in Carmon et al. (2021). Our hard instance differs from previous ones mainly in the large Lipschitz constant of its gradient. We note that the controlled degree of nonsmoothness is crucial for our estimate of PL constant.

Let $\mathbf{y} \in \mathbb{R}^{Tt}$ be a vector satisfying $\mathbf{y}_{qT+b} = \left(\frac{7}{8}\right)^q$, where $q \in \mathbb{N}$, $b \in \{1, 2, \cdots, T\}$. We define the hard instance $g_{T,t} : \mathbb{R}^{Tt} \to \mathbb{R}$ as follows:

$$g_{T,t}(\mathbf{x}) = q_{T,t}(\mathbf{x}) + \sum_{i=1}^{Tt} v_{\mathbf{y}_i}(\mathbf{x}_i). \tag{17}$$

Now we list some properties of $g_{T,t}$ in Lemma 2, which we prove in Appendix B.

Lemma 2 *$g_{T,t}$ satisfies the following.*

1. *$g_{T,t}(\mathbf{y} - \mathbf{x})$ is a zero-chain.*
2. *$\mathbf{x}^* = \mathbf{0}$, $g^*_{T,t} = 0$, $g_{T,t}(\mathbf{x}) \le \frac{1}{2}\mathbf{x}^T(\mathbf{B} + \mathbf{I})\mathbf{x}$.*
3. *$g_{T,t}$ is 37-smooth.*
4. *$g_{T,t}$ satisfies the $\frac{1}{C_3 T}$-PL condition, where C_3 is a universal constant.*

Define $\tilde{g}$ to be the following function, which is hard for first-order algorithms:

$$\tilde{g}(\mathbf{x}) = \frac{LT^{-1}D^2}{37} g_{T,t}\left(\mathbf{y} - T^{1/2}D^{-1}\mathbf{x}\right), \tag{18}$$

where $D = c\|\mathbf{x}^{(0)} - \mathbf{x}^*\|$, and c is a constant. In smooth optimization, D is often treated as a constant.

In Lemma 3 below, we show that $\tilde{g}$ is hard for first-order zero-respecting algorithms:

Lemma 3 *Assume that $\varepsilon < 0.01$ and let $t = 2\left\lfloor \log_{\frac{8}{7}} \frac{3}{2\varepsilon} \right\rfloor$. A first-order zero-respecting algorithm with $\mathbf{x}^{(0)} = \mathbf{0}$ needs at least $\frac{1}{2}Tt$ gradient computations to find a point $\mathbf{x}$ satisfying $\tilde{g}(\mathbf{x}) - \tilde{g}^* \le \varepsilon(\tilde{g}(\mathbf{x}^{(0)}) - \tilde{g}^*)$.*

Proof By induction, we have $\operatorname{supp}(\mathbf{x}^{(k)}) \subseteq \{1, \cdots, k\}$. By the definition of $\tilde{g}$ and v_y, we have

$$\begin{aligned} g_{T,t}(\mathbf{0}) &= \frac{1}{2} + \sum_{i=0}^{t-1} T v_{\left(\frac{7}{8}\right)^i}\left(\left(\frac{7}{8}\right)^i\right) \\ &= \frac{1}{2} + T\sum_{i=0}^{t-1} \frac{31}{64}\left(\frac{7}{8}\right)^{2i} \\ &= \frac{1}{2} + T \cdot \frac{31}{15} \cdot \left(1 - \left(\frac{7}{8}\right)^{2t}\right) \\ &\le 3T. \end{aligned} \tag{19}$$

For $k \leq \frac{1}{2}Tt$,

$$\begin{aligned} g_{T,t}(\mathbf{x}^{(k)}) &\geq \sum_{i=\frac{t}{2}}^{t} T v_{\left(\frac{7}{8}\right)^i}\left(\left(\frac{7}{8}\right)^i\right) \\ &= T\sum_{i=\frac{t}{2}}^{t} \frac{31}{64}\left(\frac{7}{8}\right)^{2i} \\ &= T\cdot\left(\frac{7}{8}\right)^{\frac{t}{2}}\cdot\frac{31}{15}\left(1-\left(\frac{7}{8}\right)^{t}\right) \\ &\geq 2T\cdot\frac{3\varepsilon}{2} \\ &= 3T\varepsilon. \end{aligned} \tag{20}$$

Therefore, for $k \leq \frac{1}{2}Tt$, $\tilde{g}(\mathbf{x}^{(k)}) - \tilde{g}^* \geq \varepsilon(\tilde{g}(\mathbf{x}^{(0)}) - \tilde{g}^*)$. ■

With Lemma 2 and 3, we obtain a lower bound for zero-respeting algorithms:

Theorem 4 *Given $L \geq \mu > 0$. When $\kappa = \frac{L}{\mu} > C_4$ where C_4 is a universal constant, there exists T and t such that $\tilde{g}$ is L-smooth and μ-PL. Moreover, any first-order zero-respecting algorithm with $\mathbf{x}^{(0)} = \mathbf{0}$ needs at least $\Omega\left(\kappa\log\frac{1}{\varepsilon}\right)$ gradient computations to find a point $\mathbf{x}$ satisfying $\tilde{g}(\mathbf{x}) - \tilde{g}^* \leq \varepsilon(\tilde{g}(\mathbf{x}^{(0)}) - \tilde{g}^*)$.*

Proof We let $C_4 = 370C_3$. Given $\kappa > C_4$, we let

$$T = \left\lfloor \frac{\kappa}{37C_3} \right\rfloor, \tag{21}$$

and

$$t = 2\left\lfloor \log_{\frac{8}{7}} \frac{3}{2\varepsilon} \right\rfloor. \tag{22}$$

We use Lemma 2 to calculate the smoothness constant and PL constant of $\tilde{g}$. The smoothness constant of $\tilde{g}$ is:

$$\frac{LT^{-1}D^2}{37}\cdot TD^{-2}\cdot 37 = L, \tag{23}$$

and the PL constant of $\tilde{g}$ is:

$$\begin{aligned} \frac{LT^{-1}D^2}{37}\cdot TD^{-2}\cdot\frac{1}{C_3T} &= \frac{L}{37C_3T} \\ &\overset{a}{=} \frac{L}{37C_3}\cdot\frac{1}{\left\lfloor\frac{\kappa}{37C_3}\right\rfloor} \\ &\geq \frac{L}{\kappa} \\ &= \mu, \end{aligned} \tag{24}$$

where $\overset{a}{=}$ uses (21).

Finally, by Lemma 3, any first-order zero-respecting algorithm needs at least $\frac{Tt}{2}$ accesses to gradient to find $\mathbf{x}$ such that $\tilde{g}(x) - \tilde{g}^* \leq \varepsilon \left(\tilde{g}(\mathbf{x}^{(0)}) - \tilde{g}^*\right)$. By (21) and (22),

$$\frac{Tt}{2} \geq \left(\frac{\kappa}{37C_1} - 1\right) \cdot \left(\log_{\frac{8}{7}} \frac{3}{2\varepsilon}\right) = \Omega\left(\kappa \log \frac{1}{\varepsilon}\right), \tag{25}$$

which completes the proof. ■

Using the technique of Nemirovskij and Yudin (1983), for specific function classes such as PL functions, a lower complexity bound on first-order zero-respecting algorithms is also a lower complexity bound on all the first-order algorithms. Denoting the set of all first-order algorithms by $\mathcal{A}^{(1)}$, we have the following lemma:

Lemma 5

$$\mathcal{T}_\varepsilon\left(\mathcal{A}^{(1)}, \mathcal{P}(\Delta, \mu, L)\right) \geq \mathcal{T}_\varepsilon\left(\mathcal{A}_{\mathrm{zr}}, \mathcal{P}(\Delta, \mu, L)\right). \tag{26}$$

Proof The set $\mathcal{P}(\Delta, \mu, L)$ satisfies orthogonal invariance property (Bassily et al., 2018). Therefore, the results follows from Proposition 1 of Carmon et al. (2020). ■

Finally, we arrive at a lower bound for first-order algorithms:

Theorem 6 *For any* $0 < a < 1$, *when* $\varepsilon \leq \frac{1}{16}\Delta$,

$$\mathcal{T}_\varepsilon\left(\mathcal{A}^{(1)}, \mathcal{P}(\Delta, \mu, L)\right) \geq \Omega\left(\kappa \log \frac{1}{\varepsilon}\right). \tag{27}$$

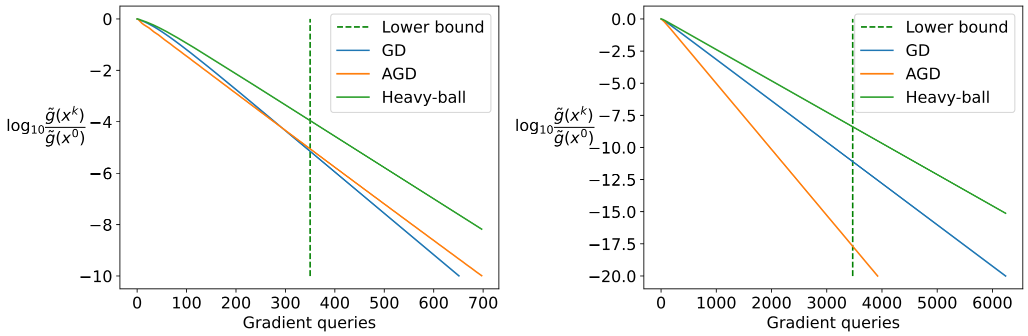
Proof The result is a direct corollary of Theorem 4 and Lemma 5. ■

This bound matches the convergence rate of Gradient Descent up to a constant.

5. Numerical experiments

We conduct numerical experiments on our hard instance. We consider the κ relatively large, which can reduce the factors from the numerical constants. We first choose κ and ε, and then decide T and t using (21) and (22). We use Gradient Descent, Nesterov's Accelerated Gradient Descent (AGD) and Polyak's Heavy-ball Method to optimize the hard instance. As AGD and the Heavy-ball Method are designed for convex functions, we need to choose appropriate parameter $\hat{\mu}$ in both algorithms, because our hard instance is non-convex. For AGD, We let $\hat{\mu} = \mu$, the PL constant of our hard instance. For Heavy-ball method, we adopt the parameter setting in (Danilova et al., 2020).

GD, AGD and Heavy-ball Method are all zero-respecting algorithms, so Lemma 3 and Theorem 4 applies to their convergence rates. From Figure 1, we observe that all three algorithms converge almost linearly, but the number of greadient queries is more than the complexity lower bound. The result is consistent with Lemma 3 and Theorem 4.



(a) $\kappa = 1.9709 \times 10^6, \varepsilon = 10^{-10}$ (b) $\kappa = 7.3119 \times 10^6, \varepsilon = 10^{-20}$

Figure 1: Convergence rate under Gradient Descent, Nesterov's Accelerated Gradient Descent and Polyak's Heavy-ball Method

6. Conclusion

We construct a lower complexity bound on optimizing smooth PL functions with first-order methods. A first-order algorithm needs at least $\Omega\left(\frac{L}{\mu}\log\frac{1}{\varepsilon}\right)$ gradient access to find an ε-approximate optimal point of an L-smooth μ-PL function. Our lower bound matches the convergence rate of Gradient Descent up to constants.

We only focus on deterministic algorithms in this paper. We conjecture that our results can be extended to randomized algorithms, using the same technique in Nemirovskij and Yudin (1983) and explicit construction in Woodworth and Srebro (2016) and Woodworth and Srebro (2017). We leave its formal derivation to the future work.

Acknowledgments

C. Fang and Z. Lin were supported by National Key R&D Program of China (2022ZD0160302). Z. Lin was also supported by the NSF China (No. 62276004), the major key project of PCL, China (No. PCL2021A12) and Qualcomm.

References

Mihai Anitescu. Degenerate nonlinear programming with a quadratic growth condition. *SIAM Journal on Optimization*, 10(4):1116–1135, 2000.

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019a.

Yossi Arjevani, Ohad Shamir, and Ron Shiff. Oracle complexity of second-order methods for smooth convex optimization. *Mathematical Programming*, 178(1-2):327–360, November 2019b. ISSN 0025-5610, 1436-4646. doi: 10.1007/s10107-018-1293-1. URL `http://link.springer.com/10.1007/s10107-018-1293-1`.

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Ayush Sekhari, and Karthik Sridharan. Second-order information in non-convex stochastic optimization: Power and limitations. In *Conference on Learning Theory*, pages 242–299. PMLR, 2020.

Raef Bassily, Mikhail Belkin, and Siyuan Ma. On exponential convergence of SGD in non-convex over-parametrized learning. *arXiv preprint arXiv:1811.02564*, 2018.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points I. *Mathematical Programming*, 184(1):71–120, 2020.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points II: first-order methods. *Mathematical Programming*, 185(1):315–355, 2021.

Marina Danilova, Anastasiia Kulakova, and Boris Polyak. Non-monotone behavior of the heavy ball method. In *Difference Equations and Discrete Dynamical Systems with Applications: 24th ICDEA, Dresden, Germany, May 21–25, 2018 24*, pages 213–230. Springer, 2020.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31, 2018.

Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*, pages 1467–1476. PMLR, 2018.

Moritz Hardt and Tengyu Ma. Identity matters in deep learning. *arXiv preprint arXiv:1611.04231*, 2016.

Sashank J Reddi, Suvrit Sra, Barnabas Poczos, and Alexander J Smola. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. *Advances in neural information processing systems*, 29, 2016.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via scsg methods. *Advances in Neural Information Processing Systems*, 30, 2017.

Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pages 2–47. PMLR, 2018.

Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 2022.

Ji Liu, Steve Wright, Christopher Ré, Victor Bittorf, and Srikrishna Sridhar. An asynchronous parallel stochastic coordinate descent algorithm. In *International Conference on Machine Learning*, pages 469–477. PMLR, 2014.

Stanislaw Lojasiewicz. A topological property of real analytic subsets. *Coll. du CNRS, Les équations aux dérivées partielles*, 117(87-89):2, 1963.

Zhi-Quan Luo and Paul Tseng. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46(1):157–178, 1993.

Katta G Murty and Santosh N Kabadi. Some NP-complete problems in quadratic and nonlinear programming. Technical report, 1985.

Ion Necoara, Yu Nesterov, and Francois Glineur. Linear convergence of first order methods for non-strongly convex optimization. *Mathematical Programming*, 175(1):69–107, 2019.

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D Lee, and Meisam Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. *Advances in Neural Information Processing Systems*, 32, 2019.

B.T. Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963.

Jun-Kun Wang, Chi-Heng Lin, Andre Wibisono, and Bin Hu. Provable acceleration of heavy ball beyond quadratics for a class of polyak-lojasiewicz functions when the non-convexity is averaged-out. In *International Conference on Machine Learning*, pages 22839–22864. PMLR, 2022.

Blake Woodworth and Nathan Srebro. Lower bound for randomized first order convex optimization. *arXiv preprint arXiv:1709.03594*, 2017.

Blake E Woodworth and Nati Srebro. Tight complexity bounds for optimizing composite objectives. *Advances in neural information processing systems*, 29, 2016.

Hui Zhang and Wotao Yin. Gradient methods for convex minimization: better rates under weaker conditions. *arXiv preprint arXiv:1303.4645*, 2013.

Dongruo Zhou and Quanquan Gu. Lower bounds for smooth nonconvex finite-sum optimization. In *International Conference on Machine Learning*, pages 7574–7583. PMLR, 2019.

Appendix A. Proof of Theorem 1

Theorem 1 *If f is L-smooth and satisfies μ-PL condition, then the Gradient Descent algorithm with a constant step-size $\frac{1}{L}$:*

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \frac{1}{L}\nabla f(\mathbf{x}^{(k)}), \tag{28}$$

has a linear convergence rate. We have:

$$f(\mathbf{x}^{(k)}) - f^* \leq \left(1 - \frac{\mu}{L}\right)^k (f(\mathbf{x}^{(0)}) - f^*). \tag{29}$$

Proof The proof is taken from Karimi et al. (2016). By the L-smoothness of f,

$$f(\mathbf{x}^{(k+1)}) \le f(\mathbf{x}^{(k)}) + \left\langle \nabla f(\mathbf{x}^{(k)}), \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\rangle + \frac{L}{2} \left\| \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right\|^2 . \tag{30}$$

Applying (3) and (2) in (30), we have

$$f(\mathbf{x}^{(k+1)}) - f(\mathbf{x}^{(k)}) \le \frac{1}{2L} - \left\| \nabla f(\mathbf{x}^{(k)}) \right\|^2 \le -\frac{\mu}{L} \left(f(\mathbf{x}^{(k)}) - f^* \right) . \tag{31}$$

Rearranging terms in (31), we have

$$f(\mathbf{x}^{(k+1)}) - f^* \le \left(1 - \frac{\mu}{L} \right) \left(f(\mathbf{x}^{(k)}) - f^* \right) . \tag{32}$$

Applying (32) recursively gives the result. ■

Appendix B. Omitted proof in Section 4

B.1. Proof of Lemma 2

Lemma 2 *$g_{T,t}$ satisfies the following.*

1. *$g_{T,t}(\mathbf{y} - \mathbf{x})$ is a zero-chain.*
2. *$\mathbf{x}^* = \mathbf{0}$, $g^*_{T,t} = 0$, $g_{T,t}(\mathbf{x}) \le \frac{1}{2}\mathbf{x}^T(\mathbf{B} + \mathbf{I})\mathbf{x}$.*
3. *$g_{T,t}$ is 37-smooth.*
4. *$g_{T,t}$ satisfies the $\frac{1}{C_3 T}$-PL condition, where C_3 is a universal constant.*

Proof In the proof of Lemma 2, we define

$$\mathbf{b}(\mathbf{x}) = \begin{bmatrix} b_{\mathbf{y}_1}(\mathbf{x}_1) \\ \vdots \\ b_{\mathbf{y}_{Tt}}(\mathbf{x}_{Tt}) \end{bmatrix} . \tag{33}$$

1. We have

   $$\nabla g_{T,t}(\mathbf{y} - \mathbf{x}) = -(\mathbf{B}(\mathbf{y} - \mathbf{x}) + (\mathbf{y} - \mathbf{x}) - \mathbf{b}(\mathbf{y} - \mathbf{x})). \tag{34}$$

   When $\mathbf{x}_i = \cdots = \mathbf{x}_{Tt} = 0$, $b_{\mathbf{y}_j}(\mathbf{y}_j - \mathbf{x}_j) = \mathbf{y}_j$ for $j \ge i$. When $k \ge i + 1$, $(\nabla g_{T,t}(\mathbf{x}))_k = (\mathbf{B}(\mathbf{y} - \mathbf{x}))_k = 0$. Therefore, $\operatorname{supp}(\nabla g_{T,t}(\mathbf{y} - \mathbf{x})) \in \{1, 2, \cdots, i + 1\}$, which implies that $g_{T,t}(\mathbf{y} - \mathbf{x})$ is a zero-chain with respect to $\mathbf{x}$.

2. $q_{T,t}(\mathbf{x})$ attains its minimum at $\mathbf{x} = \mathbf{0}$, and $v_y(x)$ attains its minimum at $x = 0$. Therefore, $g_{T,t}(\mathbf{x}) = q_{T,t}(\mathbf{x}) + \sum_{t=1}^{Tt} v_{\mathbf{y}_i}(\mathbf{x}_i)$ attains its minimum at $\mathbf{x} = \mathbf{0}$, and $g^*_{T,t} = 0$. From the definition of $v_{T,c}$ in (12), we have $v_y(x) \le \frac{1}{2}x^2$, which implies $g_{T,t}(\mathbf{x}) \le \frac{1}{2}\mathbf{x}^T(\mathbf{B} + \mathbf{I})\mathbf{x}$.

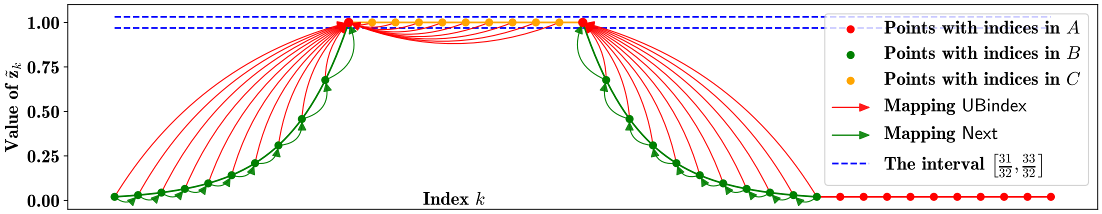


Figure 2: An example showing the intuition of our notations and how to estimate the PL constant.

3. Let

$$v(\mathbf{x}) = \sum_{i=1}^{Tt} v_{\mathbf{y}_i}(\mathbf{x}_i). \tag{35}$$

For $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{Tt}$,

$$\begin{aligned} \|\nabla v(\mathbf{x}_1) - \nabla v(\mathbf{x}_2)\| &= \|\mathbf{x}_1 - \mathbf{b}_{T,c}(\mathbf{x}_1) - \mathbf{x}_2 + \mathbf{b}_{T,c}(\mathbf{x}_2)\| \\ &\leq \|\mathbf{x}_1 - \mathbf{x}_2\| + \|\mathbf{b}_{T,c}(\mathbf{x}_1) - \mathbf{b}_{T,c}(\mathbf{x}_2)\| \\ &\leq 33\|\mathbf{x}_1 - \mathbf{x}_2\|. \end{aligned} \tag{36}$$

The last inequality of (36) is due to the definition of $b_{T,c}$, which implies that $b_{T,c}$ is 32-Lipschitz. Consequently, $g_{T,t}$ is 37-smooth because $\mathbf{B} \preceq 4\mathbf{I}$.

4. The PL constant of $g_{T,t}$ can be written as

$$\mu = \inf_{\mathbf{x} \in \mathbb{R}^{Tt}} \frac{\|\nabla g_{T,t}(\mathbf{x})\|^2}{2(g_{T,t}(x) - g^*_{T,t})} \leq \inf_{\mathbf{x} \in \mathbb{R}^{Tt}} \frac{\|\nabla g_{T,t}(\mathbf{x})\|^2}{5\|\mathbf{x}\|^2}.$$

For any $\mathbf{x}$, we estimate μ by dividing the indices of $\mathbf{x}$ into three sets A, B, C using Lemmas 7 and 8: A contains k such that $|(\nabla g_{T,t}(\mathbf{x}))_k| \geq 0.19|\mathbf{x}_k|$; B contains "exponential growing chains"; C contains "flat areas in $\left[\frac{31}{32}, \frac{33}{32}\right]$". Intuitively, if $k \in A$, then $|(\nabla g_{T,t}(\mathbf{x}))_k|^2$ is large enough, and it can be used to upper bound $|\mathbf{x}_k|^2$ and the norm of "exponential growing chains" and "flat areas in $\left[\frac{31}{32}, \frac{33}{32}\right]$" next to k (with Lemma 10). In the set B, Next defines the increasing direction of "exponential growing chains". We use $|(\nabla g_{T,t}(\mathbf{x}))_{\mathsf{UBIndex}(k)}|^2$ to upper bound $|\mathbf{x}_k|^2$. The intuitions are shown in Figure 2.

We first introduce some notations to simplify the proofs. We introduce $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{z}}$ in (37), (38) and (39) to show that the "exponential growing chains" will terminate at the beginning and end of $\{\mathbf{z}_k\}_{k=1}^{Tt}$. It also simplifies the Lemmas we introduce later. For $\mathbf{x} \in \mathbb{R}^{Tt}$, let

$$\tilde{\mathbf{x}} = \begin{bmatrix} 0 \\ \mathbf{x} \\ \mathbf{x}_{Tt} \end{bmatrix} \in \mathbb{R}^{Tt+2}, \tag{37}$$

$$\tilde{\mathbf{B}} = \begin{bmatrix} -\mathbf{e}^{(1)} & \mathbf{B} + \mathbf{E}_{Tt,Tt} & -\mathbf{e}^{(Tt)} \end{bmatrix} \in \mathbb{R}^{Tt \times (Tt+2)}, \tag{38}$$

and

$$\tilde{\mathbf{I}} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{Tt} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{Tt \times (Tt+2)}, \tag{39}$$

where $\mathbf{e}^{(i)}$ is the ith column of $\mathbf{I}$, and $\mathbf{E}_{Tt,Tt}$ is an $Tt \times Tt$ whose (Tt, Tt) entry is 1 and other entries are 0. We have $\mathbf{B}\mathbf{x} = \tilde{\mathbf{B}}\tilde{\mathbf{x}}$, and $\mathbf{x} = \tilde{\mathbf{I}}\tilde{\mathbf{x}}$. We use $\tilde{\mathbf{x}}_0, \cdots, \tilde{\mathbf{x}}_{Tt+1}$ to denote the coordinates of $\tilde{\mathbf{x}}$, i.e. $\tilde{\mathbf{x}}_0 = 0$, $\tilde{\mathbf{x}}_i = \mathbf{x}_i$ for $i = 1, \cdots, Tt$, and $\tilde{\mathbf{x}}_{Tt+1} = \mathbf{x}_{Tt}$. Similarly, we define $\tilde{\mathbf{y}}$ by $\tilde{\mathbf{y}}_0 = 1$, $\tilde{\mathbf{y}}_i = \mathbf{y}_i$ for $i = 1, \cdots, Tt$, and $\tilde{\mathbf{y}}_{Tt+1} = \mathbf{y}_{Tt}$. With these newly defined notations, we can check that

$$\left(\nabla g_{T,t}(\mathbf{x})\right)_k = (\tilde{\mathbf{B}}\tilde{\mathbf{x}})_k + \tilde{\mathbf{x}}_k - b_{\tilde{\mathbf{y}}_k}(\tilde{\mathbf{x}}_k) \tag{40}$$

for $k = 1, 2, \cdots, Tt$.

We define $C_3 = \frac{21344400}{1083}$. Let μ be the PL constant of $g_{T,t}$.

$$\begin{aligned} \mu &= \inf_{\mathbf{x} \in \mathbb{R}^{Tt}} \frac{\|\nabla g_{T,t}(\mathbf{x})\|^2}{2(g_{T,t}(x) - g^*_{T,t})} \\ &= \inf_{\mathbf{x} \in \mathbb{R}^{Tt}} \frac{\|\nabla g_{T,t}(\mathbf{x})\|^2}{2 g_{T,t}(\mathbf{x})} \\ &\overset{(a)}{\geq} \inf_{\mathbf{x} \in \mathbb{R}^{Tt}} \frac{\|(\mathbf{B}+\mathbf{I})\mathbf{x} - \mathbf{b}_{T,c}(\mathbf{x})\|^2}{\mathbf{x}^T(\mathbf{B}+\mathbf{I})\mathbf{x}} \\ &\overset{(b)}{\geq} \inf_{\mathbf{x} \in \mathbb{R}^{Tt}} \frac{\|(\mathbf{B}+\mathbf{I})\mathbf{x} - \mathbf{b}_{T,c}(\mathbf{x})\|^2}{5\|\mathbf{x}\|^2} \end{aligned} \tag{41}$$

where (a) follows from property 2 of Lemma 2, and (b) follows from $\mathbf{B} + \mathbf{I} \preceq 5\mathbf{I}$. Define

$$\tilde{\mu}(\mathbf{x}) = \frac{\|\nabla g_{T,t}(\mathbf{x})\|^2}{5\|\mathbf{x}\|^2} = \frac{\|(\mathbf{B}+\mathbf{I})\mathbf{x} - \mathbf{b}_{T,c}(\mathbf{x})\|^2}{5\|\mathbf{x}\|^2}. \tag{42}$$

We only need to give a lower bound on $\tilde{\mu}$. We estimate the proportion $\frac{\|\nabla g_{T,t}(\mathbf{x})\|}{\|\tilde{\mathbf{x}}_k\|} = \frac{\|(\mathbf{B}+\mathbf{I})\mathbf{x} - \mathbf{b}_{T,c}(\mathbf{x})\|}{\|\mathbf{x}\|}$ by computing each $\left|\frac{(\nabla g_{T,t}(\mathbf{x}))_k}{\tilde{\mathbf{x}}_k}\right| = \left|\frac{((\mathbf{B}+\mathbf{I})\mathbf{x})_k - b_{\tilde{\mathbf{y}}_k}(\tilde{\mathbf{x}}_k)}{\tilde{\mathbf{x}}_k}\right|$. If $\left|\frac{(\nabla g_{T,t}(\mathbf{x}))_k}{\tilde{\mathbf{x}}_k}\right|$ is small, we will upper bound $\tilde{\mathbf{x}}_k^2$ by one of the nearby $(\nabla g_{T,t}(\mathbf{x}))_n^2$ terms.

We define an operator Dominate : $\{1, \cdots, Tt\} \to \{\text{True}, \text{False}\}$ as follows:

$$\mathsf{Dominate}(k) = \begin{cases} \text{True}, & \text{If} \left|\frac{(\nabla g_{T,t}(\mathbf{x}))_k}{\tilde{\mathbf{x}}_k}\right| > 0.19 \text{ or } \tilde{\mathbf{x}}_k = 0, \\ \text{False}, & \text{otherwise.} \end{cases} \tag{43}$$

Define $\tilde{\mathbf{z}}_k = \frac{\tilde{\mathbf{x}}_k}{\tilde{\mathbf{y}}_k}$, we present four auxiliary lemmas below. We prove them in Section B.2.

Lemma 7 *For $k = 1, \cdots, Tt$, if $\tilde{\mathbf{z}}_k \notin [\frac{31}{32}, \frac{33}{32}]$ and $\neg\mathsf{Dominate}(k)$, then $\frac{\tilde{\mathbf{z}}_{k-1}}{\tilde{\mathbf{z}}_k} \geq \frac{4}{3}$ or $\frac{\tilde{\mathbf{z}}_{k+1}}{\tilde{\mathbf{z}}_k} \geq \frac{4}{3}$.*

Lemma 8 *If $\tilde{\mathbf{z}}_{k-1} < \frac{5}{7}$, $\tilde{\mathbf{z}}_k \in [\frac{31}{32}, \frac{33}{32}]$, $\tilde{\mathbf{z}}_{k+1} \leq \frac{33}{32}$, then $\mathsf{Dominate}(k) = \text{True}$.*

Lemma 9 *If $\tilde{\mathbf{z}}_{k-2} < \frac{31}{32}$, $\frac{5}{7} \leq \tilde{\mathbf{z}}_{k-1} < \frac{31}{32}$, $\tilde{\mathbf{z}}_k \in [\frac{31}{32}, \frac{33}{32}]$, then $\mathsf{Dominate}(k-1) = \text{True}$.*

Lemma 10 *If $\tilde{\mathbf{z}}_n \in [\frac{31}{32}, \frac{33}{32}]$, $\neg\mathsf{Dominate}(n)$ and $\tilde{\mathbf{z}}_{n-1}, \tilde{\mathbf{z}}_{n+1} \leq \frac{33}{32}$, there exist k such that $k < n$, $\mathsf{Dominate}(k)$ and $\tilde{\mathbf{z}}_k > \frac{5}{7}$.*

Algorithm 1 Algorithm to find $\mathsf{UBIndex}(n)$

```
Input: $\neg\mathsf{Dominate}(n)$
if $\mathsf{Next}(n)$ is defined then
    $m \leftarrow n$;
    while $\neg\mathsf{Dominate}(m)$ do
        $m \leftarrow \mathsf{Next}(m)$;        ▷ This process is well-defined and will terminate.
    end
    $\mathsf{UBIndex}(n) \leftarrow m$;
else if $\mathsf{Next}(n)$ is not defined then
    $\mathsf{UBIndex}(n) \leftarrow \mathrm{argmax}_{k<n}\{\tilde{\mathbf{z}}_k : \mathsf{Dominate}(k)\}$;        ▷ $\mathsf{UBIndex}(n)$ exists and $\tilde{\mathbf{z}}_{\mathsf{UBIndex}(n)} > \frac{5}{7}$ (by Lemma 10).
end
```

Note that if we alter the "$+$" and "$-$" sign in Lemma 8 or 9, the conclusion still holds.

Now we define an operator Next on indices on which Dominate operator is false. Intuitively, Next finds the direction in which $\tilde{\mathbf{z}}$ grows exponentially.

By Lemma 7, if $\tilde{\mathbf{z}}_k \notin [\frac{31}{32}, \frac{33}{32}]$ and $\neg\mathsf{Dominate}(k)$, define $\mathsf{Next}(k) \in \{k-1, k+1\}$ to be one of the coordinate satisfying $\frac{\tilde{\mathbf{z}}_{\mathsf{Next}(k)}}{\tilde{\mathbf{z}}_k} \geq \frac{4}{3}$.

Next, we define how the Next operator acts on the index n where $\tilde{\mathbf{z}}_n \in [\frac{31}{32}, \frac{33}{32}]$ and $\max\{\tilde{\mathbf{z}}_{n-1}, \tilde{\mathbf{z}}_{n+1}\} > \frac{33}{32}$. Without the loss of generality (if we alter the "$+$" and "$-$", the following conclusion still holds), if $\tilde{\mathbf{z}}_{n+1} > \frac{33}{32}$, we define $\mathsf{Next}(n) = n+1$. If $\mathsf{Next}(n-1) = n$, $\tilde{\mathbf{z}}_n \in [\frac{31}{32}, \frac{33}{32}]$ and $\neg\mathsf{Dominate}(n)$, then by Lemma 8, we have $\tilde{\mathbf{z}}_{n+1} \geq \frac{33}{32}$. Therefore, if $n = \mathsf{Next}(m)$, then $\mathsf{Next}(n)$ is defined. We can apply the Next operator recursively, and will finally reach a index n such that $\mathsf{Dominate}(n) = \mathrm{True}$. This process will terminate because $\tilde{\mathbf{z}}_0 = 0$ and $\tilde{\mathbf{z}}_{Tt+1} = \tilde{\mathbf{z}}_{Tt}$, ensuring that if the recursive Next operation reaches 0 or Tt, it terminates.

For other n such that $\neg\mathsf{Dominate}(n)$ and $\tilde{\mathbf{z}}_n \in [\frac{31}{32}, \frac{33}{32}]$ ($\tilde{\mathbf{z}}_{n-1}, \tilde{\mathbf{z}}_{n+1} \leq \frac{33}{32}$), the operator Next is undefined. We will use Lemma 10 to tackle this situation.

For n such that $\neg\mathsf{Dominate}(n)$, we define an operator UBIndex, and use a proportion of $\|g_{T,t}(\mathbf{x})\|_{\mathsf{UBIndex}(n)}$ to upper bound $\mathbf{x}_n$. The process of finding UBIndex is provided in Algorithm 1.

Define $\mathsf{Dist}(n) = |n - \mathsf{UBIndex}(n)|$, and define $\mathrm{UB}(k)$ to be a proportion of $\|g_{T,t}(\mathbf{x})\|^2_{\mathsf{UBIndex}(k)}$ as follows:

$$\mathrm{UB}(n) = \begin{cases} \frac{1}{4}|(\nabla g_{T,t}(\mathbf{x}))_n|^2, \\ \qquad\qquad \mathsf{Dominate}(n), \\ \frac{1}{4}\cdot\frac{13}{49}\cdot\left(\frac{36}{49}\right)^{\mathsf{Dist}(n)-1}|(\nabla g_{T,t}(\mathbf{x}))_{\mathsf{UBIndex}(n)}|^2, \\ \qquad\qquad \neg\mathsf{Dominate}(n), \mathsf{Next}(n) \text{ exists}, \\ \frac{1}{4T}\cdot\frac{15}{64}\cdot\frac{\tilde{\mathbf{y}}_n^2}{\tilde{\mathbf{y}}^2_{\mathsf{UBIndex}(n)}}|(\nabla g_{T,t}(\mathbf{x}))_{\mathsf{UBIndex}(n)}|^2, \\ \qquad\qquad \neg\mathsf{Dominate}(n), \mathsf{Next}(n) \text{ does not exist}. \end{cases} \tag{44}$$

Let $A = \{n : \mathsf{Dominate}(n)\}$, $B = \{n : \neg\mathsf{Dominate}(n), \mathsf{Next}(n) \text{ exists}\}$, and $C = \{n : \neg\mathsf{Dominate}(n),$
$\mathsf{Next}(n) \text{ does not exist}\}$. Now we calculate $\sum_{i=1}^{Tt} \mathrm{UB}(n)$, and show that it is smaller than $\|g_{T,t}(\mathbf{x})\|^2$.

$$\begin{aligned}\sum_{i=1}^{Tt} \mathrm{UB}(n) &= \sum_{n\in A} \mathrm{UB}(n) + \sum_{n\in B} \mathrm{UB}(n) + \sum_{n\in C} \mathrm{UB}(n)\\ &= \frac{1}{4}\sum_{n\in A} |(\nabla g_{T,t}(\mathbf{x}))_n|^2\\ &\quad + \sum_{n\in B} \frac{13}{196}\left(\frac{36}{49}\right)^{\mathsf{Dist}(n)-1} |(\nabla g_{T,t}(\mathbf{x}))_{\mathsf{UBIndex}}|^2\\ &\quad + \sum_{n\in C} \frac{15}{256T}\left(\frac{\tilde{\mathbf{y}}_n}{\tilde{\mathbf{y}}_{\mathsf{UBIndex}(n)}}\right)^2 |(\nabla g_{T,t}(\mathbf{x}))_{\mathsf{UBIndex}}|^2.\end{aligned} \tag{45}$$

By changing the order of summation, we have

$$\begin{aligned}&\sum_{n\in B} \frac{13}{196}\left(\frac{36}{49}\right)^{\mathsf{Dist}(n)-1} |(\nabla g_{T,t}(\mathbf{x}))_{\mathsf{UBIndex}}|^2\\ \le{}& \frac{13}{196}\sum_{n\in A} 2\cdot\sum_{i=1}^{\infty}\left(\frac{36}{49}\right)^{i-1} |(\nabla g_{T,t}(\mathbf{x}))_n|^2\\ ={}& \frac{1}{2}\sum_{n\in A} |(\nabla g_{T,t}(\mathbf{x}))_n|^2,\end{aligned} \tag{46}$$

and

$$\begin{aligned}&\sum_{n\in C} \frac{15}{256T}\left(\frac{\tilde{\mathbf{y}}_n}{\tilde{\mathbf{y}}_{\mathsf{UBIndex}(n)}}\right)^2 |(\nabla g_{T,t}(\mathbf{x}))_{\mathsf{UBIndex}}|^2\\ \le{}& \frac{15}{256T}\sum_{n\in A}\sum_{m\ge n}\left(\frac{\tilde{\mathbf{y}}_m}{\tilde{\mathbf{y}}_n}\right)^2 |(\nabla g_{T,t}(\mathbf{x}))_n|^2\\ ={}& \frac{15}{256T}\sum_{n\in A} T\sum_{i=0}^{\infty}\left(\frac{7}{8}\right)^{2i} |(\nabla g_{T,t}(\mathbf{x}))_n|^2\\ ={}& \frac{1}{4}\sum_{n\in A} |(\nabla g(\mathbf{x}))_n|^2.\end{aligned} \tag{47}$$

Summing up (45), (46) and (47), we have

$$\begin{aligned}\sum_{i=1}^{Tt} \mathrm{UB}(n) &= \frac{1}{4}\sum_{n\in A} |(\nabla g_{T,t}(\mathbf{x}))_n|^2 + \frac{1}{2}\sum_{n\in A} |(\nabla g_{T,t}(\mathbf{x}))_n|^2 + \frac{1}{4}\sum_{n\in A} |(\nabla g_{T,t}(\mathbf{x}))_n|^2\\ &= \sum_{n\in A} |(\nabla g_{T,t}(\mathbf{x}))_n|^2\\ &\le \|\nabla g_{T,t}(\mathbf{x})\|^2.\end{aligned} \tag{48}$$

Finally, we calculate $\frac{\mathrm{UB}(n)}{\mathbf{x}_n^2}$ to give an universal lower bound of the PL constant. For $n \in A$, we have

$$\begin{aligned} \frac{\mathrm{UB}(n)}{\mathbf{x}_n^2} &= \frac{1}{4} \cdot \frac{|(\nabla g_{T,t}(\mathbf{x}))_n|^2}{\mathbf{x}_n^2} \\ &\overset{(a)}{\geq} \frac{361}{40000}, \end{aligned} \tag{49}$$

where (a) is due to $\mathsf{Dominate}(n) = \mathrm{True}$.

For $n \in B$, by Lemma 7 we have $\left|\frac{\tilde{\mathbf{z}}_{\mathsf{Next}(m)}}{\tilde{\mathbf{z}}_m}\right| \geq \frac{4}{3}$ for $m = n, \mathsf{Next}(n), \mathsf{Next}(\mathsf{Next}(n)) \cdots$, with only one possible exception when $\tilde{\mathbf{z}}_m \in [\frac{31}{32}, \frac{33}{32}]$, in which case there is $|\tilde{\mathbf{z}}_{\mathsf{Next}(m)}| \geq |\tilde{\mathbf{z}}_m|$. Therefore, $|\tilde{\mathbf{z}}_{\mathsf{UBIndex}(n)}| \geq \left(\frac{4}{3}\right)^{\mathsf{Dist}(n)-1} |\tilde{\mathbf{z}}_n|$, so we have

$$\begin{aligned} \frac{\mathrm{UB}(n)}{\mathbf{x}_n^2} &= \frac{1}{4} \cdot \frac{13}{49} \cdot \left(\frac{36}{49}\right)^{\mathsf{Dist}(n)-1} |(\nabla g_{T,t}(\mathbf{x}))_{\mathsf{UBIndex}(n)}|^2 \\ &\overset{(a)}{\geq} \frac{13}{196} \cdot \left(\frac{36}{49}\right)^{\mathsf{Dist}(n)-1} \frac{361}{10000} \frac{\mathbf{x}_{\mathsf{UBIndex}(n)}^2}{\mathbf{x}_n^2} \\ &\overset{(b)}{=} \frac{4693}{1960000} \left(\frac{36}{49}\right)^{\mathsf{Dist}(n)-1} \cdot \left(\frac{\tilde{\mathbf{z}}_{\mathsf{UBIndex}(n)}}{\tilde{\mathbf{z}}_n}\right)^2 \cdot \left(\frac{\tilde{\mathbf{y}}_{\mathsf{UBIndex}(n)}}{\tilde{\mathbf{y}}_n}\right)^2 \\ &\overset{(c)}{\geq} \frac{4693}{1960000} \left(\frac{36}{49}\right)^{\mathsf{Dist}(n)-1} \cdot \left(\frac{16}{9}\right)^{\mathsf{Dist}(n)-1} \cdot \left(\frac{49}{64}\right)^{\mathsf{Dist}(n)} \\ &= \frac{4693}{2560000}, \end{aligned} \tag{50}$$

where (a) is due to $\mathsf{Dominate}(\mathsf{UBIndex}(n)) = \mathrm{True}$, (b) is due to $\tilde{\mathbf{z}}_m = \frac{\tilde{\mathbf{x}}_m}{\tilde{\mathbf{y}}_m}$, (c) is due to $|\tilde{\mathbf{z}}_{\mathsf{UBIndex}(n)}| \geq \left(\frac{4}{3}\right)^{\mathsf{Dist}(n)-1} |\tilde{\mathbf{z}}_n|$ and $\frac{|\tilde{\mathbf{y}}_{m+1}|}{|\tilde{\mathbf{y}}_m|} \geq \frac{7}{8}$.

For $n \in C$, we have

$$\begin{aligned} \frac{\mathrm{UB}(n)}{\mathbf{x}_n^2} &= \frac{1}{4T} \cdot \frac{15}{64} \cdot \frac{\tilde{\mathbf{y}}_n^2}{\tilde{\mathbf{y}}_{\mathsf{UBIndex}(n)}^2} \cdot \frac{|(\nabla g_{T,t}(\mathbf{x}))_{\mathsf{UBIndex}(n)}|^2}{\mathbf{x}_n^2} \\ &\overset{(a)}{\geq} \frac{15}{256T} \cdot \frac{\tilde{\mathbf{y}}_n^2}{\tilde{\mathbf{y}}_{\mathsf{UBIndex}(n)}^2} \cdot \frac{361\tilde{\mathbf{x}}_{\mathsf{UBIndex}(n)}^2}{40000\tilde{\mathbf{x}}_n^2} \\ &\overset{(b)}{\geq} \frac{1083}{2048000T} \frac{\tilde{\mathbf{z}}_{\mathsf{UBIndex}(n)}^2}{\tilde{\mathbf{z}}_n^2} \\ &\overset{(c)}{\geq} \frac{1083}{2048000T} \left(\frac{5/7}{33/32}\right)^2 \\ &= \frac{1083}{4268880T}, \end{aligned} \tag{51}$$

where (a) is due to $\mathsf{Dominate}(\mathsf{UBIndex}(n)) = \text{True}$, (b) is due to $\tilde{\mathbf{z}}_m = \frac{\tilde{\mathbf{x}}_m}{\tilde{\mathbf{y}}_m}$, and (c) is due to Lemma 10 and $\tilde{\mathbf{z}}_n \in [\frac{31}{32}, \frac{33}{32}]$. Therefore,

$$\begin{aligned} \mu &\geq \inf_{\mathbf{x}} \frac{\|\nabla g_{T,t}(\mathbf{x})\|^2}{5\mathbf{x}^2} \\ &\geq \inf_{\mathbf{x}} \frac{\sum_{n=1}^{Tt} \mathrm{UB}(n)}{5\sum_{n=1}^{Tt} \mathbf{x}_n^2} \\ &\geq \inf_{\mathbf{x}} \min_{n} \frac{\mathrm{UB}(n)}{5\mathbf{x}_n^2} \\ &\geq \inf_{\mathbf{x}} \frac{1}{5} \cdot \min\left\{ \frac{361}{40000}, \frac{4693}{2560000}, \frac{1083}{4268880T} \right\} \\ &= \frac{1083}{21344400T}. \end{aligned} \tag{52}$$

■

B.2. Proof of Lemma 7, 8, 9 and 10

Lemma 7 *For $k = 1.\cdots.Tt$, if $\tilde{\mathbf{z}}_k \notin [\frac{31}{32}, \frac{33}{32}]$ and $\neg\mathsf{Dominate}(k)$, then $\frac{\tilde{\mathbf{z}}_{k-1}}{\tilde{\mathbf{z}}_k} \geq \frac{4}{3}$ or $\frac{\tilde{\mathbf{z}}_{k+1}}{\tilde{\mathbf{z}}_k} \geq \frac{4}{3}$.*

Proof If $\tilde{\mathbf{z}}_k \notin [\frac{31}{32}, \frac{33}{32}]$, then $b_{\tilde{\mathbf{y}}_k}(\tilde{\mathbf{x}}_k) = 0$, and $(\nabla g_{T,t}(\mathbf{x}))_k = (\tilde{\mathbf{B}}\tilde{\mathbf{x}})_k + \tilde{\mathbf{x}}_k$. By $\neg\mathsf{Dominate}(k)$, we have $\left|(\tilde{\mathbf{B}}\tilde{\mathbf{x}})_k + \tilde{\mathbf{x}}_k\right| < 0.19\,|\tilde{\mathbf{x}}_k|$. We consider three cases:

1. If $k = T+1, 2T+1, \cdots, T(t-1)+1$, $(\tilde{\mathbf{B}}\tilde{\mathbf{x}})_k = -\frac{7}{8}\tilde{\mathbf{x}}_{k-1} + 2\tilde{\mathbf{x}}_k - \tilde{\mathbf{x}}_{k+1}$, $(\tilde{\mathbf{B}}\tilde{\mathbf{x}})_k + \tilde{\mathbf{x}}_k = \frac{7}{8}\tilde{\mathbf{x}}_{k-1} + 3\tilde{\mathbf{x}}_k - \tilde{\mathbf{x}}_{k+1}$, and $\frac{7}{8}\tilde{\mathbf{y}}_{k-1} = \tilde{\mathbf{y}}_k = \tilde{\mathbf{y}}_{k+1}$. We have

$$\begin{aligned} 3|\tilde{\mathbf{x}}_k| - \frac{7}{8}|\tilde{\mathbf{x}}_{k-1}| - |\tilde{\mathbf{x}}_{k+1}| &\leq |(\tilde{\mathbf{B}}\tilde{\mathbf{x}})_k + \tilde{\mathbf{x}}_k| \\ &< 0.19|\tilde{\mathbf{x}}_k|. \end{aligned} \tag{53}$$

   Dividing both sides by $\tilde{\mathbf{y}}_k$, we have

$$3|\tilde{\mathbf{z}}_k| - |\tilde{\mathbf{z}}_{k-1}| - |\tilde{\mathbf{z}}_{k+1}| < 0.19|\tilde{\mathbf{z}}_k|. \tag{54}$$

   Thus, we have $|\tilde{\mathbf{z}}_{k-1}| + \tilde{\mathbf{z}}_{k+1} > \frac{8}{3}|\tilde{\mathbf{z}}_k|$, which indicates that $\frac{\tilde{\mathbf{z}}_{k-1}}{\tilde{\mathbf{z}}_k} \geq \frac{4}{3}$ or $\frac{\tilde{\mathbf{z}}_{k+1}}{\tilde{\mathbf{z}}_k} \geq \frac{4}{3}$.

2. If $k = T, 2T, \cdots, T(t-1)$, $(\tilde{\mathbf{B}}\tilde{\mathbf{x}})_k + \tilde{\mathbf{x}}_k = -\tilde{\mathbf{x}}_{k-1} + \frac{177}{64}\tilde{\mathbf{x}}_k - \frac{7}{8}\tilde{\mathbf{x}}_{k+1}$, and $\tilde{\mathbf{y}}_{k-1} = \tilde{\mathbf{y}}_k = \frac{8}{7}\tilde{\mathbf{y}}_{k+1}$. Thus, we have

$$\frac{177}{64}|\tilde{\mathbf{x}}_k| - |\tilde{\mathbf{x}}_{k-1}| - \frac{7}{8}|\tilde{\mathbf{x}}_{k+1}| < 0.19|\tilde{\mathbf{x}}_k|. \tag{55}$$

   Dividing both sides by $\tilde{\mathbf{y}}_k$, we have

$$\frac{177}{64}|\tilde{\mathbf{z}}_k| - |\tilde{\mathbf{z}}_{k-1}| - \frac{49}{64}|\tilde{\mathbf{z}}_{k+1}| < 0.19|\tilde{\mathbf{z}}_k|. \tag{56}$$

   Thus, we have

$$|\tilde{\mathbf{z}}_{k-1}| + \frac{49}{64}|\tilde{\mathbf{z}}_{k+1}| > \left(\frac{177}{64} - 0.19\right)|\tilde{\mathbf{z}}_k| > \frac{4}{3} \cdot \left(1 + \frac{49}{64}\right)|\tilde{\mathbf{z}}_k|. \tag{57}$$

3. For other k, $(\tilde{\mathbf{B}}\tilde{\mathbf{x}})_k + \tilde{\mathbf{x}}_k = -\tilde{\mathbf{x}}_{k-1} + 3\tilde{\mathbf{x}}_k - \tilde{\mathbf{x}}_{k+1}$, and $\tilde{\mathbf{y}}_{k-1} = \tilde{\mathbf{y}}_k = \tilde{\mathbf{y}}_{k+1}$. Therefore, we have (by dividing $\tilde{\mathbf{y}}_k$ to both sides of (54))

$$|\tilde{\mathbf{z}}_{k-1}| + |\tilde{\mathbf{z}}_{k+1}| > 2.81|\tilde{\mathbf{z}}_k| > \frac{8}{3}|\tilde{\mathbf{z}}_k|. \tag{58}$$

■

Lemma 8 *If* $\tilde{\mathbf{z}}_{k-1} < \frac{5}{7}$, $\tilde{\mathbf{z}}_k \in [\frac{31}{32}, \frac{33}{32}]$, $\tilde{\mathbf{z}}_{k+1} \leq \frac{33}{32}$, *then* $\mathsf{Dominate}(k) = \mathrm{True}$.

Proof For any $y > 0$, we have $0 \leq b_y(x) \leq y$. Therefore, $(\nabla g_{T,t}(\mathbf{x}))_k \geq (\tilde{\mathbf{B}}\tilde{\mathbf{x}})_k$.

1. If $k = T+1, 2T+1, \cdots, T(t-1)+1$, $(\tilde{\mathbf{B}}\tilde{\mathbf{x}})_k = -\frac{7}{8}\tilde{\mathbf{x}}_{k-1} + 2\tilde{\mathbf{x}}_k - \tilde{\mathbf{x}}_{k+1}$ and $\frac{7}{8}\tilde{\mathbf{y}}_{k-1} = \tilde{\mathbf{y}}_k = \tilde{\mathbf{y}}_{k+1}$.

$$\begin{aligned} \frac{(\nabla g_{T,t}(\mathbf{x}))_k}{\tilde{\mathbf{x}}_k} &\geq \frac{(\tilde{\mathbf{B}}\tilde{\mathbf{x}})_k}{\tilde{\mathbf{x}}_k} \\ &= \frac{-\frac{7}{8}\tilde{\mathbf{x}}_{k-1} + 2\tilde{\mathbf{x}}_k = \tilde{\mathbf{x}}_{k+1}}{\tilde{\mathbf{x}}_k} \\ &\overset{(a)}{=} \frac{-\tilde{\mathbf{z}}_{k-1} + \tilde{\mathbf{z}}_k - \tilde{\mathbf{z}}_{k+1}}{\tilde{\mathbf{z}}_k} \\ &\overset{(b)}{>} -\frac{5/7}{31/32} + 2 - \frac{33/32}{31/32} \\ &> 0.19, \end{aligned} \tag{59}$$

where (a) holds by dividing $\tilde{\mathbf{y}}_k$ to the numerator and denominator, (b) holds by the assumptions on $\tilde{\mathbf{z}}_{k-1}, \tilde{\mathbf{z}}_k, \tilde{\mathbf{z}}_{k+1}$.

2. If $k = T, 2T, \cdots, T(t-1)$, $(\tilde{\mathbf{B}}\tilde{\mathbf{x}})_k = -\tilde{\mathbf{x}}_{k-1} + \frac{113}{64}\tilde{\mathbf{x}}_k - \frac{7}{8}\tilde{\mathbf{x}}_{k+1}$, and $\tilde{\mathbf{y}}_{k-1} = \tilde{\mathbf{y}}_k = \frac{8}{7}\tilde{\mathbf{y}}_{k+1}$.

$$\begin{aligned} \frac{(\nabla g_{T,t}(\mathbf{x}))_k}{\tilde{\mathbf{x}}_k} &\geq \frac{(\tilde{\mathbf{B}}\tilde{\mathbf{x}})_k}{\tilde{\mathbf{x}}_k} \\ &= \frac{-\tilde{\mathbf{x}}_{k-1} + \frac{113}{64}\tilde{\mathbf{x}}_k - \frac{7}{8}\tilde{\mathbf{x}}_{k+1}}{\tilde{\mathbf{x}}_k} \\ &\overset{(a)}{=} \frac{-\tilde{\mathbf{z}}_{k-1} + \frac{113}{64}\tilde{\mathbf{z}}_k - \frac{49}{64}\tilde{\mathbf{z}}_{k+1}}{\tilde{\mathbf{z}}_k} \\ &\overset{(b)}{>} -\frac{5/7}{31/32} + \frac{113}{64} - \frac{49}{64} \cdot \frac{33/32}{31/32} \\ &> 0.19, \end{aligned} \tag{60}$$

where (a) holds by dividing $\tilde{\mathbf{y}}_k$ to the numerator and denominator, and (b) holds by the assumptions on $\tilde{\mathbf{z}}_{k-1}, \tilde{\mathbf{z}}_k$, and $\tilde{\mathbf{z}}_{k+1}$.

3. For other k, $(\tilde{\mathbf{B}}\tilde{\mathbf{x}})_k = -\tilde{\mathbf{x}}_{k-1} + 2\tilde{\mathbf{x}}_k - \tilde{\mathbf{x}}_{k+1}$, and $\tilde{\mathbf{y}}_{k-1} = \tilde{\mathbf{y}}_k = \tilde{\mathbf{y}}_{k+1}$. Therefore,

$$\begin{aligned}\frac{(\nabla g_{T,t}(\mathbf{x}))_k}{\tilde{\mathbf{x}}_k} &\geq \frac{(\tilde{\mathbf{B}}\tilde{\mathbf{x}})_k}{\tilde{\mathbf{x}}_k} \\ &= \frac{-\tilde{\mathbf{x}}_{k-1} + 2\tilde{\mathbf{x}}_k = \tilde{\mathbf{x}}_{k+1}}{\tilde{\mathbf{x}}_k} \\ &\overset{(a)}{=} \frac{-\tilde{\mathbf{z}}_{k-1} + \tilde{\mathbf{z}}_k - \tilde{\mathbf{z}}_{k+1}}{\tilde{\mathbf{z}}_k} \\ &\overset{(b)}{>} -\frac{5/7}{31/32} + 2 - \frac{33/32}{31/32} \\ &> 0.19,\end{aligned} \tag{61}$$

where (a) holds by dividing $\tilde{\mathbf{y}}_k$ to the numerator and denominator, and (b) holds by the assumptions on $\tilde{\mathbf{z}}_{k-1}, \tilde{\mathbf{z}}_k$, and $\tilde{\mathbf{z}}_{k+1}$.

■

Lemma 9 *If* $\tilde{\mathbf{z}}_{k-2} < \frac{31}{32}$, $\frac{5}{7} \leq \tilde{\mathbf{z}}_{k-1} < \frac{31}{32}$, $\tilde{\mathbf{z}}_k \in [\frac{31}{32}, \frac{33}{32}]$, *then* $\mathsf{Dominate}(k-1) = \mathrm{True}$.

Proof Like the proof of Lemma 8, we directly compute $\frac{(\nabla g_{T,t}(\mathbf{x}))_{k-1}}{\tilde{\mathbf{x}}_{k-1}}$. Because $\tilde{\mathbf{z}}_{k-1} < \frac{31}{32}$, $b_{\tilde{\mathbf{y}}_{k-1}}(\tilde{\mathbf{x}}_{k-1}) = 0$.

$$\begin{aligned}\frac{(\nabla g_{T,t}(\tilde{\mathbf{x}}))_{k-1}}{\tilde{\mathbf{x}}_{k-1}} &= \frac{(\tilde{\mathbf{B}}\tilde{\mathbf{x}})_{k-1} + \tilde{\mathbf{x}}_{k-1}}{\tilde{\mathbf{x}}_{k-1}} \\ &\overset{(a)}{=} -a\frac{\tilde{\mathbf{z}}_{k-2}}{\tilde{\mathbf{z}}_{k-1}} + b - c\frac{\tilde{\mathbf{z}}_k}{\tilde{\mathbf{z}}_{k-1}},\end{aligned} \tag{62}$$

where (a) holds by dividing $\tilde{\mathbf{y}}_k$ to the numerator and denominator. For $k-1 = T, 2T, \cdots, T(t-1)$, $a = -1$, $b - \frac{177}{64}$ and $c = \frac{49}{64}$. For other k, $a = c = 1$ and $b = 3$. By the assumptions on $\tilde{\mathbf{z}}_{k-2}, \tilde{\mathbf{z}}_{k-1}$ and $\tilde{\mathbf{z}}_k$, we have $\frac{\tilde{\mathbf{z}}_{k-2}}{\tilde{\mathbf{z}}_{k-1}} < \frac{31/32}{5/7}$, and $\frac{\tilde{\mathbf{z}}_k}{\tilde{\mathbf{z}}_{k-1}} < \frac{33/32}{5/7}$. Plugging everything into (62), we have $\frac{(\nabla g_{T,t}(\tilde{\mathbf{x}}))_{k-1}}{\tilde{\mathbf{x}}_{k-1}} > 0.2$. ■

Lemma 10 *If* $\tilde{\mathbf{z}}_n \in [\frac{31}{32}, \frac{33}{32}]$, $\neg\mathsf{Dominate}(n)$ *and* $\tilde{\mathbf{z}}_{n-1}, \tilde{\mathbf{z}}_{n+1} \leq \frac{33}{32}$, *there exist* k *such that* $k < n$, $\mathsf{Dominate}(k)$ *and* $\tilde{\mathbf{z}}_k > \frac{5}{7}$.

Proof Define $m = \mathrm{argmax}_{m'} \left\{\tilde{\mathbf{z}}_{m'} \notin [\frac{31}{32}, \frac{33}{32}] : m' < n\right\}$. If $\tilde{\mathbf{z}}_m > \frac{33}{32}$, let k be m if $\mathsf{Dominate}(m)$ and $\mathsf{UBIndex}(m)$ if $\neg\mathsf{Dominate}(m)$. If $\tilde{\mathbf{z}}_m \in [\frac{5}{7}, \frac{33}{32}]$, let k be m if $\mathsf{Dominate}(m)$ and $\mathsf{UBIndex}(m)$ if $\neg\mathsf{Dominate}(m)$. Finally, if $\tilde{\mathbf{z}}_m < \frac{5}{7}$, let $k = m + 1$. ■