DADA: Dual Averaging with Distance Adaptation

Mohammad Moshtaghifar M.MOSHTAGHI@SHARIF.EDU
Sharif University of Technology, Iran

Anton Rodomanov ANTON.RODOMANOV@CISPA.DE
CISPA, Germany

Daniil Vankov DVANKOV@ASU.EDU
Arizona State University, USA

Sebastian U. Stich STICH@CISPA.DE
CISPA, Germany

Abstract

We present a novel parameter-free universal gradient method for solving convex optimization problems. Our algorithm—Dual Averaging with Distance Adaptation (DADA)–is based on the classical scheme of dual averaging and dynamically adjusts its coefficients based on the observed gradients and the distance between its iterates to the starting point, without the needing to know any problem-specific parameters. DADA is a universal algorithm that simultaneously works for a wide range of problem classes as long as one is able to bound the local growth of the objective around its minimizer. Particular examples of such problem classes are nonsmooth Lipschitz functions, Lipschitz-smooth functions, Hölder-smooth functions, functions with high-order Lipschitz derivative, quasi-self-concordant functions, and (L_0, L_1)-smooth functions. Furthermore, in contrast to many existing methods, DADA is suitable not only for unconstrained problems, but also constrained ones, possibly with unbounded domain, and it does not require fixing neither the number of iterations nor the accuracy in advance.

Keywords: Convex Optimization, Gradient Methods, Adaptive Algorithms, Parameter-Free Methods, Dual Averaging, Distance Adaption, Universal Methods, Worst-Case Complexity Guarantees

1. Introduction

We consider the following optimization problem:

$$f^* := \min_{x \in Q} f(x), \tag{1}$$

where $Q \subseteq \mathbb{R}^d$ is a simple and nonempty closed convex set, and $f \colon \mathbb{R}^d \to \mathbb{R}$ is a convex function on Q. By simplicity, we mean that it is possible to compute the projection onto Q. We assume this problem has a solution, which we denote by x^*.

Gradient methods are among the most popular and efficient optimization algorithms for solving machine learning problems, as they are highly adaptable and scalable across various settings [3]. One of the key challenges in solving (1) using gradient methods is selecting appropriate hyperparameters, particularly stepsizes, which significantly impact performance. Hyperparameter tuning, as one of the standard approaches to address this issue, is a time-consuming and resource-intensive process, especially as models grow larger and more complex. Therefore, the cost of training these models has become a significant concern [20, 23]. To address this, there has been a growing interest in so-called parameter-free algorithms [4, 5, 10, 11, 13, 19], which aim to eliminate the need for manual tuning.

Typically, line search techniques have been used to select step sizes in optimization, and they work well for certain function classes, such as Hölder-smooth problems [16]. However, in recent years, several parameter-free approaches have been developed which do not utilize line search. Notably, one strategy involves dynamically adjusting stepsizes based on estimates of the initial distance to the optimal solution, $D_0 = \|x_0 - x^*\|$ [4, 10, 11]. Another approach leverages lower bounds for D_0 combined with the dual averaging scheme [6, 13]. However, these methods primarily focus on nonsmooth Lipschitz or, in some cases, Lipschitz-smooth functions. In contrast, our method establishes a universal result that is not restricted to these classes but applies to a broader range of convex function classes. Additionally, These methods sometimes have limitations, such as requiring bounded domain assumptions [11], and lacking applicability to constrained optimization problems [6, 13].

Contributions. In this paper, we introduce DADA—Dual Averaging with Distance Adaptation—a novel parameter-free universal optimization algorithm for solving (1). DADA is based on the classical scheme of weighted Dual Averaging (DA) [15], but uses a specially designed, dynamically adjusted estimate of $D_0 = \|x_0 - x^*\|$, based on the recent technique proposed in [4, 10] and also used in [11], without requiring prior knowledge of problem-specific parameters. Furthermore, our approach applies to both unconstrained problems and problems with simple constraints, whose domains are not required to be bounded, making it a powerful tool across a wide range of applications.

This paper is organized as follows. In Section 2, we present our method and outline its foundational structure based on the DA scheme [15]. We establish our main result in Theorem 1, showing convergence guarantees that apply to a broad range of function classes.

To demonstrate the versatility and effectiveness of DADA, in Section 3, we provide complexity estimates across several interesting function classes: Nonsmooth Lipschitz functions, Lipschitz-smooth functions, Hölder-smooth functions, Quasi-Self-Concordant (QSC) functions, functions with Lipschitz pth derivative, and (L_0, L_1)-smooth functions. This highlights the ability of DADA to deliver competitive performance without requiring knowledge of class-specific parameters.

Notation. In this text, we work in the space $\mathbb{R}^d$ equipped with the standard inner product $\langle \cdot, \cdot \rangle$ and the general Euclidean (Mahalanobis) norm:

$$\|x\| := \langle Bx, x \rangle^{1/2}, \qquad x \in \mathbb{R}^d,$$

where B is a fixed symmetric positive definite matrix. The corresponding dual norm is defined in the standard way:

$$\|s\|_* := \max_{\|x\|=1} \langle s, x \rangle = \langle s, B^{-1}s \rangle^{1/2}, \qquad s \in \mathbb{R}^d.$$

Thus, for any $s, x \in \mathbb{R}^d$, we have the Cauchy–Schwarz inequality $|\langle s, x \rangle| \leq \|s\|_* \|x\|$. For a convex function $f \colon \mathbb{R}^d \to \mathbb{R}$, we denote its subdifferential at a point $x \in \mathbb{R}^d$ by $\partial f(x)$.

```
Algorithm 1: General Scheme of DA
Input: $x_0 \in Q, T \geq 1$, coefficients $(a_k)_{k=0}^{T-1}$, $(\beta_k)_{k=1}^{T}$ with nondecreasing $\beta_k$
for $k = 1, \dots, T$ do
    Compute arbitrary $g_k \in \partial f(x_k)$;
    $x_k = \operatorname{argmin}_{x \in Q} \{\psi_k(x) = \sum_{i=0}^{k-1} a_i \langle g_i, x - x_i \rangle + \frac{\beta_k}{2} \|x - x_0\|^2\}$;
end
return $x_T^* = \operatorname{argmin}_{x \in \{x_0, \dots, x_T\}} f(x)$;
```

2. DADA Method

Measuring the quality of solution. Rather than focusing on bounding the distance to the optimal point x^*, this work focuses on bounding the distance from x^* to the hyperplane $\{y : \langle \nabla f(x), x - y \rangle = 0\}$,

$$v(x) := \frac{\langle \nabla f(x), x - x^* \rangle}{\|\nabla f(x)\|_*} \, (\geq 0) \, , \quad \text{where} \quad x \in \mathbb{R}^d.$$

This goal is meaningful because minimizing $v(x)$ also reduces the corresponding function residual, $f(x) - f^*$. Indeed, there exists the following simple relationship between $v(x)$ and the function residual [17, Section 3.2.2] (see also Appendix A for the short proof).

$$f(x) - f^* \leq \omega(v(x)), \tag{2}$$

where

$$\omega(t) := \max_x \{f(x) - f^* \colon \|x - x^*\| \leq t\} \, ,$$

measures the local growth of f around the solution x^*. By bounding the $\omega(t)$, we can derive convergence rate estimates that simultaneously apply to a broad range of problem classes (we discuss several examples in Section 3).

The method. Our proposed approach is based on the general scheme of DA [15] shown in Algorithm 1. Using a standard (sub)gradient method with time-varying coefficients is also possible but requires either short steps by fixing the number of iterations in advance, or paying an extra logarithmic factor in the convergence rate [17, Section 3.2.3].

The classical DA method has two primary variants. The first, Simple DA, uses a constant coefficient $a_i = \hat{D}_0$. The second, Weighted DA, instead of using a constant sequence, adjusts the coefficients using $a_i = \frac{\hat{D}_0}{\|g_i\|_*}$. However, both variants pay a multiplicative cost of ρ^2, where $\rho := \max\{\frac{\hat{D}_0}{D_0}, \frac{D_0}{\hat{D}_0}\}$, due to the lack of prior knowledge about the parameter D_0. This cost can be significantly high. To address this issue, we propose DADA, which reduces the cost to a logarithmic term, $\log^2 \rho$, offering a substantial improvement. Specifically, our approach introduces a dynamic sequence for $(a_i)_{i=0}^{\infty}$ and $(\beta_i)_{i=1}^{\infty}$, thereby eliminating the need for the parameter D_0. In our approach, we utilize the following time-varying coefficients:

$$\boxed{a_k = \frac{\bar{r}_k}{\|g_k\|_*}, \quad \beta_k = 2\sqrt{k+1}} \, , \quad \text{where} \quad \bar{r}_k = \max\{\max_{1 \leq t \leq k} r_t, \bar{r}\}, \; r_t = \|x_0 - x_t\|, \tag{3}$$

and $\bar{r} > 0$ is a certain user-specified parameter. In what follows, we assume w.l.o.g. that $g_k \neq 0$ for all $0 \leq k \leq T$ since otherwise the exact the solution has been found, and the method could be successfully terminated before making T iterations.

Our method estimates the parameter D_0 using $\bar{r}_t$, the distance between x_t and the initial point x_0. This idea has been recently explored in recent works [4, 10], which similarly utilize $\bar{r}_t$ in various ways. Other methods also attempt to estimate this quantity using alternative strategies, based on Dual Averaging and the similar principle of employing an increasing sequence of lower bounds for D_0 [6, 13].

As discussed earlier, our goal is to bound the function $v(\cdot)$ to establish the convergence of our method. We present the following convergence result for our method (see Appendix C for the proof).

Theorem 1 *Consider Algorithm 1 for solving problem* (1) *using the coefficients from* (3)*. Then, for any $T \geq 1$ and $v_T^* = \min_{0 \leq t \leq T} v(x_t)$, it holds that,*

$$f(x_T^*) - f^* \leq \omega(v_T^*),$$

where

$$v_T^* \leq \frac{9R}{\sqrt{T}} \left(\frac{8R}{\bar{r}}\right)^{\frac{1}{T}} \log \frac{8eR}{\bar{r}},$$

and $R = \max\{\|x_0 - x^*\|, \bar{r}\}$. Further, for a given $\delta > 0$, it holds that $v_T^* \leq \delta$ whenever $T \geq \max\{\log \frac{8R}{\bar{r}}, \frac{81e^2R^2}{\delta^2} \log^2 \frac{8eR}{\bar{r}}\}$.

The detailed proof can be found in Appendix C. Let us provide a proof sketch for Theorem 1. We begin by applying the standard result for DA (Theorem 4), which holds for any choice of coefficients a_k and β_k.

$$\sum_{i=0}^{k-1} a_i v_i \|g_i\|_* + \frac{\beta_k}{2} d_k^2 \leq \frac{\beta_k}{2} D_0^2 + \sum_{i=0}^{k-1} \frac{a_i^2}{2\beta_i} \|g_i\|_*^2, \tag{4}$$

where $D_0 = \|x_0 - x^*\|$ and $d_k = \|x_k - x^*\|$. Next, we introduce specific choices for a_k and β_k as defined in (3). This gives us,

$$\sum_{i=0}^{k-1} \bar{r}_i v_i + d_k^2 \sqrt{k+1} \leq D_0^2 \sqrt{k+1} + \frac{1}{4} \sum_{i=0}^{k-1} \frac{\bar{r}_i^2}{\sqrt{i+1}} \leq D_0^2 \sqrt{k+1} + \frac{\bar{r}_{k-1}^2}{2} \sqrt{k}. \tag{5}$$

Using the fact that $\bar{r}_i v_i \geq 0$ for all $1 \leq i \leq k-1$, we can show by induction that $\bar{r}_k$ is bounded up to a constant factor by $R = \max\{D_0, \bar{r}\}$ (see Theorem 6):

$$\bar{r}_{k-1} \leq 8R.$$

This bound is crucial to our analysis, as we need to eliminate $\bar{r}_{k-1}$ from the right-hand side of (5). Achieving this requires selecting the coefficients precisely as defined in (3), which is the primary difference compared to the standard DA method [15]. Next, using the following inequality $\|x_0 - x^*\|^2 - \|x_k - x^*\|^2 \leq 2\|x_k - x_0\| D_0$, we derive the next result:

$$\sum_{i=0}^{k-1} \bar{r}_i v_i \leq \bar{r}_k (2R + \tfrac{1}{2}\bar{r}_k)\sqrt{k+1} \leq 6\bar{r}_k R\sqrt{k+1}.$$

After establishing this, the rest of the proof follows straightforwardly by dividing both sides with $\sum_{i=0}^{k-1} \bar{r}_i$ and then using the following inequality (valid for any nondecreasing sequence $\bar{r}_t$, see Theorem 2):

$$\min_{0 \leq t \leq T} \frac{\bar{r}_t}{\sum_{i=0}^{t-1} \bar{r}_i} \leq \frac{\left(\frac{\bar{r}_T}{\bar{r}_0}\right)^{\frac{1}{T}} \log \frac{e\bar{r}_T}{\bar{r}_0}}{T}.$$

At this point, we clarify the key differences between our method and approaches like DoG [10]. One obvious difference is that we use DA instead of the classical (sub)gradient method employed by DoG. However, the most significant difference lies in how the sequence of gradients is handled. DoG normalizes the current gradient g_k by the accumulated norms of the previous gradients, an idea inspired by AdaGrad [9]. In contrast, our method simply normalize g_k by its own norm. Additionally, this modification makes our method universal, enabling it to work with the growth function ω, which is not known to be the case for DoG, even for deterministic problems.

2.1. Comparison with Recent Parameter-Free Methods

Let us briefly compare our method with several recently proposed parameter-free algorithms, namely, DoG [10], DoWG [11], D-Adaptation [6] and Prodigy [13].

Comparison with DoG/DoWG. Both DoG and DoWG employ a similar approach to estimate D_0 and achieve comparable convergence rates for Lipschitz-smooth and nonsmooth functions. However, neither of them extends to the important cases explored in this paper. Additionally, like our approach, the DoWG method considers only the deterministic case, but with an additional assumption on a bounded domain. They have a different definition of universality, considering only smooth and nonsmooth settings. However, in this work by universality we mean that our method, without any modifications, ensures a reasonable convergence rate on any problem class, as long as the corresponding growth function is reasonably bounded, which is a very mild assumption as demonstrated by the variety of examples we present later (see Section 3).

Finally, we note the main focus of DoG is providing guarantees in the stochastic setting, given that stochastic gradients are bounded by a known constant. In this work, however, we focus exclusively on the deterministic setting, but obtain stonger results which are valid for a large variety of problem classes.

Comparison with D-Adaptation/Prodigy. D-Adaptation and Prodigy are similar to our method in their use of Dual Averaging. However, their approaches cannot be extended to the constrained optimization setting and are limited to Lipschitz functions. Nonetheless, their methods yielded notable results in experiments, demonstrating strong empirical performance.

3. Universality of DADA: Examples of Applications

Let us demonstrate that our method is universal in the sense that it simultaneously works for several interesting problem classes without the need for choosing different parameters for each of these function classes. For simplicity, we assume that $\nabla f(x^*) = 0$ (this happens, in particular, when our problem (1) is unconstrained). In what follows, the ϵ-accuracy is measured in terms of the function residual and we also use $\log_+ t := 1 + \log t$ to simplify the notation. Recall that $R = \max\{\|x_0 - x^*\|, \bar{r}\}$, where $\bar{r}$ is a parameter of our method.

Nonsmooth Lipschitz functions. This function class is defined by the inequality $\|\nabla f(x)\|_* \leq L_0$ for all $x \in Q$. For this problem class, DADA requires at most

$$O\left(\frac{L_0^2 R^2}{\epsilon^2} \log_+^2 \frac{R}{r}\right)$$

oracle calls to reach ϵ-accuracy (see Appendix D.1), which coincides with the standard complexity of (sub)gradient methods [15, 17], up to an extra logarithmic factor. This logarithmic factor is common for all distance-adaptation methods [6, 10, 11, 13].

Lipschitz-smooth functions. Another important class of functions are those with Lipschitz gradient: $\|\nabla f(x) - \nabla f(y)\|_* \leq L_1 \|x - y\|$ for all $x, y \in Q$. In this case, the complexity of our method is

$$O\left(\frac{L_1 R^2}{\epsilon} \log_+^2 \frac{R}{\bar{r}}\right)$$

(see Appendix D.2), coincides with the standard complexity of the (nonaccelerated) gradient method on Lipschitz-smooth functions [17, Section 3] up to the extra logarithmic factor, which arises due to the parameter-free nature of the method.

Note that the complexity of DADA is slightly worse than that of the classical gradient method with line search [16], which achieves a complexity bound of $O\bigl(\frac{L_1 D_0^2}{\epsilon} + \log|\frac{L_1}{\hat{L}_1}|\bigr)$, where $\hat{L}_1$ is our initial guess for L_1. The difference that they have an additive logarithmic factor in their rate instead of multiplicative.

Hölder-smooth functions. The previous two functions classes are subclasses of the more general class of Hölder-smooth functions. It is defined by the following inequality: $\|\nabla f(x) - \nabla f(y)\|_* \leq H_\nu \|x - y\|^\nu$ for all $x, y \in Q$, where $\nu \in [0, 1]$ and $H_\nu \geq 0$. Therefore, for $\nu = 0$, we get functions with bounded variation of subgradients (which contains all Lipschitz functions) and for $\nu = 1$ we get L_1-smooth functions.

The complexity of DADA on this problem class is

$$O\left(\left[\frac{H_\nu}{\epsilon}\right]^{\frac{2}{1+\nu}} R^2 \log_+^2 \frac{R}{\bar{r}}\right),$$

This is similar to the $O\bigl(\bigl[\frac{H_\nu}{\epsilon}\bigr]^{\frac{2}{1+\nu}} D_0^2 + \log|\frac{H_\nu^{\frac{2}{1+\nu}}}{L_0 \epsilon^{\frac{1-\nu}{1+\nu}}}|\bigr)$ complexity of the universal (nonaccelerated) gradient method [16]. Again, the complexity of the line-search method (GM-LS) is slightly better since the logarithmic factor is additive (and not multiplicative). However, GM-LS is not guaranteed to work (well) on other problem classes such as those we consider next.

Functions with Lipschitz high-order derivative. Functions in this class have the property that their pth derivative ($p \geq 2$) is Lipschitz, i.e.,

$$\|\nabla^p f(x) - \nabla^p f(y)\| \leq L_p \|x - y\|,$$

where the $\|\cdot\|$ norm in the left-hand side is the usual operator norm of a symmetric multilinear opeator: $\|A\| = \max_{h \in \mathbb{R}^d\colon \|h\|=1} \|Ah\|$. For example, pth power of the Euclidean norm [21] is an example of functions in this class. This class generalizes the Lipschitz-smooth class. The complexity of DADA on this class is

$$O\left(\max\left\{ \max_{2 \leq i \leq p} \left[\frac{p}{i!} \frac{\|\nabla^i f(x^*)\|_*}{\epsilon} \right]^{\frac{2}{i}}, \left[\frac{p}{(p+1)!} \frac{L_p}{\epsilon} \right]^{\frac{2}{p+1}} \right\} R^2 \log_+^2 \frac{R}{\bar{r}} \right),$$

Although line-search gradient methods might be better for Hölder-smooth problems, to our knowledge, they are not known to attain comparable bounds on this function class.

Quasi-self-concordant (QSC) functions [2]. A convex function f is said to be QSC with parameter $M \geq 0$, if for any $x \in \mathbb{R}^d$ and arbitrary directions $u, v \in \mathbb{R}^d$ it holds that

$$\nabla^3 f(x)[u, u, v] \leq M \langle \nabla^2 f(x) u, u \rangle \|v\|. \tag{6}$$

For example, the exponential functions, the logistic function $f(x) = \sum_{i=1}^n \log(1 + e^{\langle a_i, x \rangle})$, and the soft-max $f(x) = \mu \log(\sum_{i=1}^n e^{[\langle a_i, x \rangle + b_i]/\mu})$ are QSC. For more details and other examples, see [7]. Our method guarantees convergence for QSC functions with the following complexity:

$$O\left(\frac{\|\nabla^2 f(x^*)\| R^2}{\epsilon} \log_+^2 \frac{R}{\bar{r}} + (MR)^2 \log_+^2 \frac{R}{\bar{r}} + \log_+ \frac{R}{\bar{r}} \right).$$

In terms of comparisons, second-order methods, such as those explored in [7], are more powerful for minimizing QSC functions, as they leverage additional curvature information. Their complexity bound is $O(MD_0 \log \frac{F_0}{\epsilon} + \log \frac{D_0 g_0}{\epsilon F_0})$, where $F_0 = f(x_0) - f^*$, in terms of queries to the second-order oracle [7, Corollary 3.4]. However, each iteration of these methods is significantly more expensive.

To our knowledge, this class has not been studied before in the context of first-order methods. The only other first-order methods for which one can prove similar bounds are nonadaptive variants of our scheme, namely the normalized gradient method from [17, Section 5] and the recent variant of this method for constrained problems [18].

(L_0, L_1)-smooth functions. As introduced in [25], a function f is said to be (L_0, L_1)-smooth if for all $x \in \mathbb{R}^n$, we have $\|\nabla^2 f(x)\| \leq L_0 + L_1 \|\nabla f(x)\|_*$. The complexity of DADA on this class

$$O\left(\frac{L_0 R^2}{\epsilon} \log_+^2 \frac{R}{\bar{r}} + (L_1 R)^2 \log_+^2 \frac{R}{\bar{r}} + \log_+ \frac{R}{\bar{r}} \right).$$

Up to the extra logarithmic factors, this is exactly the same complexity as that of NGM from [24], knowing the exact distance to the solution D_0.

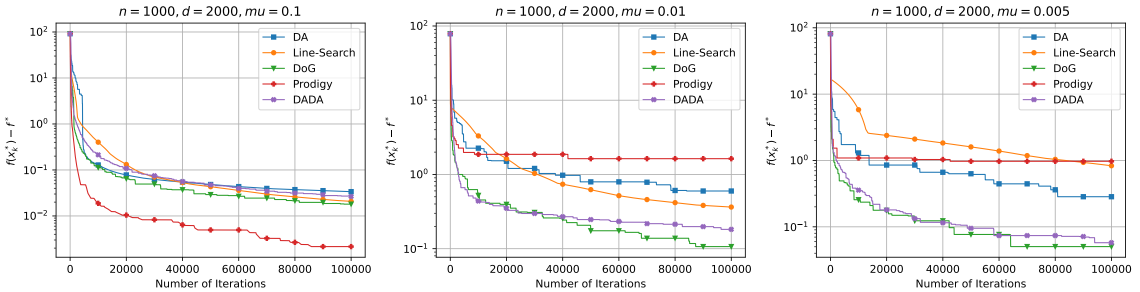

Figure 1: Comparison of different methods on the Softmax function.

4. Experiments

To evaluate the performance of our proposed method, DADA, we conduct a series of experiments on convex optimization problems. Our goal is to demonstrate the effectiveness of DADA in achieving competitive convergence rates across various function classes without relying on hyperparameter tuning. We compare DADA against several parameter-free optimization algorithms, such as DoG [10] and Prodigy [13]. We also consider gradient descent with line search [1] and classical Dual Averaging method [15]. The experiments also explore the relationship between method's dynamic distance-based step size to the true value of initial distance D_0.

For each method, we plot the best function value among all the test points generated by the algorithm after k gradient-oracle calls. Throughout these experiments, we set the initial point as $x_0 = (1, \cdots, 1)$. Additionally, we selected our initial guess for the true distance, denoted as $\bar{r}$ [1], as follows: $\bar{r} = 10^{-6}(1 + \|x_0\|)$. This choice is reasonable, primarily because, at the start, we have no prior knowledge of how far x^* might be from x_0. By choosing $\bar{r}$ in this way, which has a small coefficient, we can be reasonably confident that it does not exceed the actual value of $\|x_0 - x^*\|$. It is worth noting that this initial guess is also used by DoG.

Softmax function. We consider the softmax function:

$$\min_{x \in \mathbb{R}^d} \left[f(x) := \mu \log \left(\sum_{i=1}^{n} \exp \left[\frac{\langle a_i, x \rangle - b_i}{\mu} \right] \right) \right],$$

where $a_i \in \mathbb{R}^d$, and $b_i \in \mathbb{R}$ for all $1 \leq i \leq n$, and $\mu > 0$. This function can be seen as a smooth approximation to $\max_{1 \leq i \leq n} [\langle a_i, x \rangle - b_i]$ [14]. To generate the data for our problem we proceed as follows. First, we set $x^* = 0$. Next, we generate i.i.d. vectors a_i with components uniformly distributed in the interval $[-1, 1]$ for $i = 1, \ldots, n$, and similarly for the scalar values b_i. To ensure that x^* is a minimizer of f, we compute $\nabla f(0)$ and then, adjust a_i by setting $a_i \leftarrow a_i - \nabla f(0)$ to ensure that $\nabla f(0) = 0$.

The results are shown in Fig. 1, where we fix $n = 10^3$, $d = 10^2$, and $R = 1$ with the starting point $x_0 = 0$. We plot the total number of gradient-oracle calls against the function residual for different values of $\mu \in \{1, 0.1, 0.01\}$. As shown in Fig. 1, most methods exhibit similar performance for $\mu = 0.1$, with the exception of Prodigy, which performs slightly better in this case. However, as μ decreases, Prodigy's performance declines, whereas our method remains largely unaffected.

[1] This value corresponds to r_ϵ in [10]

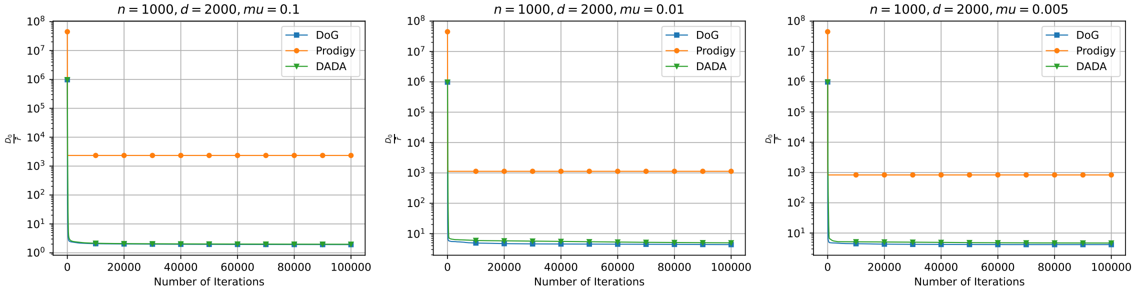

Figure 2: The ratio $\frac{D}{\bar{r}_t}$ for the Softmax function with different optimal points x^*.

This decline in performance is observed for DA, Gradient Descent with Line-Search. Notably, DoG performs similarly to DADA, which we hypothesize is primarily due to the similarity in estimating D_0. Additionally, Fig. 2 illustrates the ratio between $\bar{r}$ and D_0, highlighting the estimation error of Prodigy, DoG, and DADA at each iteration. For Prodigy, we used $\frac{D_0}{d_{\max}}$ to generate the plot. The figure demonstrates that DADA and DoG exhibit similar behavior in estimating D_0, despite employing different update methods—Dual Averaging and Gradient Descent, respectively. However, Prodigy appears to encounter challenges in estimating D_0. As shown in Fig. 2, its estimation error converges to a relatively large value. Addressing this issue could be a promising direction for improving Prodigy's estimation accuracy and, consequently, its performance in optimizing the objective function.

Hölder-Smooth Sample Function. In this section, we focus on solving the following test problem:

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{n} \sum_{i=1}^{n} [\langle a_i, x \rangle - b_i]_+^q, \tag{7}$$

where $a_i, b_i \in \mathbb{R}^d$, $q \in [1, 2]$, and $[x]_+ = \max(0, x)$.

Note that f is a Hölder-smooth function with the parameter $\nu = q - 1$. This allows us to evaluate the effectiveness of parameter-free algorithms for different values of ν in Hölder-smooth functions. By varying $q \in [1, 2]$, we demonstrate the robustness of DADA in achieving convergence over this spectrum of convex functions.

The data for our problem is generated randomly, following the procedure in [22]. First, we sample x^* uniformly from the sphere of radius $0.95R$ centered at the origin. Next, we generate i.i.d. vectors a_i with components uniformly distributed in $[-1, 1]$. To ensure that $\langle a_n, x^* \rangle < 0$, we invert the sign of a_n if necessary. We then sample positive reals s_i uniformly from $[0, -0.1c_{\min}]$, where $c_{\min} := \min_i \langle a_i, x^* \rangle < 0$, and set $b_i = \langle a_i, x^* \rangle + s_i$. By construction, x^* is a solution to the problem with $f^* = 0$. Moreover, the origin $x_0 = 0$ lies outside the polyhedron, since there exists a j (corresponding to $c_{\min}$) such that $b_j = c_{\min} + s_j \leq 0.9c_{\min} < 0$.

In this section, we fix $n = 10^4$, $d = 10^3$ and $R = 10^6$. As shown in Fig. 3, as q increases and approaches 2, the performance of DoG declines and fails to converge as effectively as it does for smaller values of q. The figure also illustrates that classical methods, DA and GD-LS, perform poorly on this class of functions for $q < 2$. However, DADA and Prodigy demonstrate similar performance regardless of the choice of q.

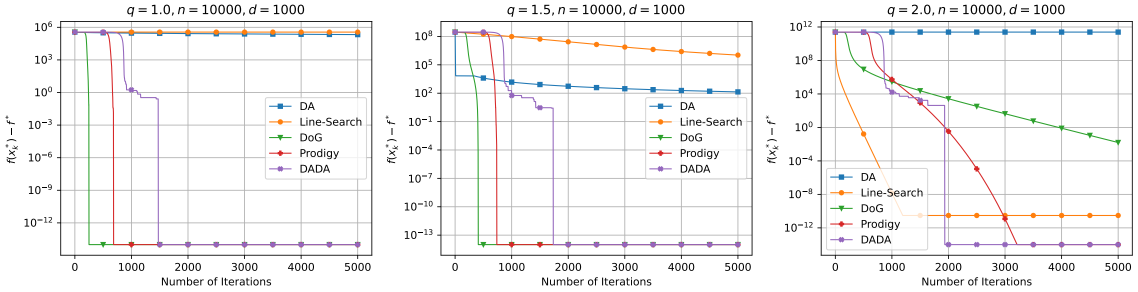

Figure 3: Comparison of different methods on the polyhedron feasibility problem.

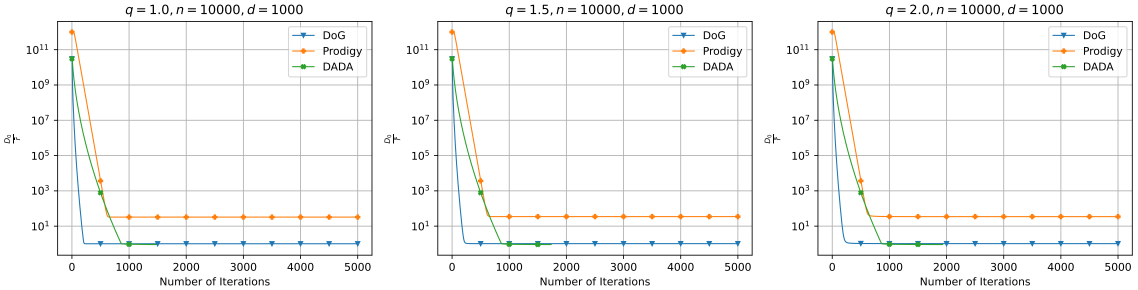

Figure 4: The ratio $\frac{D}{\bar{r}_t}$ for the polyhedron feasibility problem.

Worst-case Function. As an example of functions with Lipschitz high-order derivative, we consider worst-case function from [8]:

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{q} \sum_{i=1}^{d-1} |x^{(i)} - x^{(i+1)}|^q + \frac{1}{q} |x^{(d)}|^q, \tag{8}$$

where $q \geq 2$.

As shown in Fig. 5 and Fig. 6, both the performance and the estimation of D_0 in DoG and Prodigy deteriorate as q increases. A similar trend is observed for GD-LS, which performs comparably to DoG and slightly better than DADA when $q = 2$. However, for $q = 6$, GD-LS exhibits significantly worse performance compared to when it applied to smaller values of q.

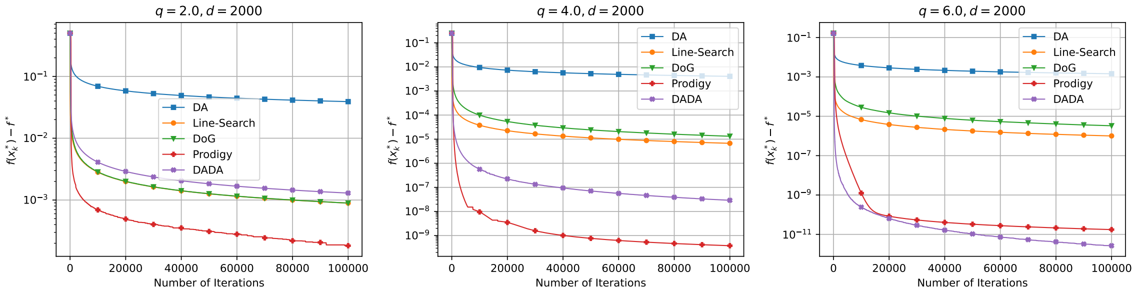

Figure 5: Comparison of different methods on the worst-case function.

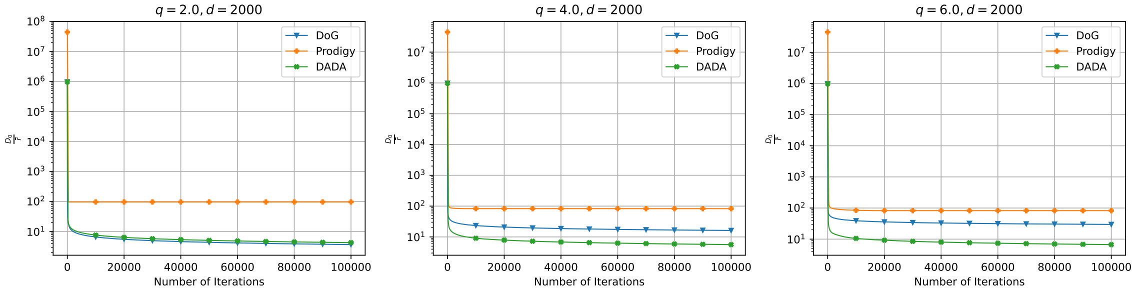

Figure 6: The ratio $\frac{D}{\bar{r}_t}$ for the worst-case function with different optimal points x^*.

References

[1] Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

[2] Francis Bach. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4(none):384 – 414, 2010. doi: 10.1214/09-EJS521. URL https://doi.org/10.1214/09-EJS521.

[3] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018. URL https://doi.org/10.1137/16M1080173.

[4] Yair Carmon and Oliver Hinder. Making sgd parameter-free. In *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178, pages 2360–2389, 2022. URL https://proceedings.mlr.press/v178/carmon22a.html.

[5] Ashok Cutkosky and Francesco Orabona. Black-box reductions for parameter-free online learning in banach spaces. In *Annual Conference Computational Learning Theory*, 2018. URL https://api.semanticscholar.org/CorpusID:3346292.

[6] Aaron Defazio and Konstantin Mishchenko. Learning-rate-free learning by d-adaptation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 7449–7479, 2023. URL https://proceedings.mlr.press/v202/defazio23a.html.

[7] Nikita Doikov. Minimizing quasi-self-concordant functions by gradient regularization of newton method, 2023. URL https://arxiv.org/abs/2308.14742.

[8] Nikita Doikov, Konstantin Mishchenko, and Yurii Nesterov. Super-universal regularized newton method. *SIAM Journal on Optimization*, 34(1):27–56, 2024. doi: 10.1137/22M1519444. URL https://doi.org/10.1137/22M1519444.

[9] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011. URL http://jmlr.org/papers/v12/duchi11a.html.

[10] Maor Ivgi, Oliver Hinder, and Yair Carmon. DoG is SGD's best friend: A parameter-free dynamic step size schedule. In *Proceedings of the 40th International Conference on Machine Learning*, pages 14465–14499, 2023. URL https://proceedings.mlr.press/v202/ivgi23a.html.

[11] Ahmed Khaled, Konstantin Mishchenko, and Chi Jin. DoWG unleashed: An efficient universal parameter-free gradient descent method. In *Advances in Neural Information Processing Systems*, volume 36, pages 6748–6769, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/15ce36d35622f126f38e90167de1a350-Paper-Conference.pdf.

[12] Zijian Liu and Zhengyuan Zhou. Stochastic nonsmooth convex optimization with heavy-tailed noises. *ArXiv*, abs/2303.12277, 2023. URL https://api.semanticscholar.org/CorpusID:257663403.

[13] Konstantin Mishchenko and Aaron Defazio. Prodigy: An expeditiously adaptive parameter-free learner. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pages 35779–35804, 2024. URL https://proceedings.mlr.press/v235/mishchenko24a.html.

[14] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103:127–152, 2005.

[15] Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120:221–259, 2005. URL https://api.semanticscholar.org/CorpusID:14935076.

[16] Yurii Nesterov. Universal gradient methods for convex optimization problems. *Mathematical Programming*, 152:381–404, 2015. URL https://api.semanticscholar.org/CorpusID:18062781.

[17] Yurii Nesterov. *Lectures on Convex Optimization*. Springer Publishing Company, Incorporated, 2nd edition, 2018. ISBN 3319915770. URL https://api.semanticscholar.org/CorpusID:14935076.

[18] Yurii Nesterov. Primal subgradient methods with predefined step sizes. *Journal of Optimization Theory and Applications*, 2024. doi: 10.1007/s10957-024-02456-9. URL https://arxiv.org/abs/2308.14742.

[19] Francesco Orabona and Tatiana Tommasi. Training deep networks without learning rates through coin betting. In *Neural Information Processing Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:6762437.

[20] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training, 2021. URL https://arxiv.org/abs/2104.10350.

[21] Anton Rodomanov and Yurii Nesterov. Smoothness parameter of power of euclidean norm. *Journal of Optimization Theory and Applications*, 185:303–326, 2019. URL https://api.semanticscholar.org/CorpusID:198968030.

[22] Anton Rodomanov, Xiaowen Jiang, and Sebastian U. Stich. Universality of adagrad stepsizes for stochastic optimization: Inexact oracle, acceleration and variance reduction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=rniiAVjHi5.

[23] Or Sharir, Barak Peleg, and Yoav Shoham. The cost of training nlp models: A concise overview, 2020. URL https://arxiv.org/abs/2004.08900.

[24] Daniil Vankov, Anton Rodomanov, Angelia Nedich, Lalitha Sankar, and Sebastian U. Stich. Optimizing (l_0, l_1)-smooth functions by gradient methods, 2024. URL https://arxiv.org/abs/2410.10800.

[25] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJgnXpVYwS.

Appendix A. Auxiliary Results

The following result has been proved in prior works such as [12, Lemma 30]. We include the proof here for the reader's convenience.

Lemma 2 *Let $(d_i)_{i=0}^{\infty}$ be a positive nondecreasing sequence. Then for any $T \geq 1$,*

$$\min_{0 \leq t \leq T} \frac{d_t}{\sum_{i=0}^{t-1} d_i} \leq \frac{\left(\frac{d_T}{d_0}\right)^{\frac{1}{T}} \log \frac{e d_T}{d_0}}{T}.$$

Proof Let $A_t := \sum_{i=0}^{t-1} \frac{d_i}{d_t}$ for all $t \in \mathbb{N}^+$ where $A_0 = 0$. Then we know

$$d_t A_t - d_{t-1} A_{t-1} = d_{t-1},$$

which implies that

$$\frac{d_{t-1}}{d_t} = A_t - \frac{d_{t-1}}{d_t} A_{t-1} = A_t - A_{t-1} + \left(1 - \frac{d_{t-1}}{d_t} A_{t-1}\right).$$

By summing up for all $1 \leq t \leq T$ we get

$$A_T + \sum_{t=0}^{T-1} \left(1 - \frac{d_t}{d_{t+1}}\right) A_t = \sum_{t=0}^{T-1} \frac{d_t}{d_{t+1}}.$$

Denote $S_T = \sum_{t=0}^{T-1} \frac{d_t}{d_{t+1}}$ and $A_T^* = \max_{0 \leq t \leq T} A_t$. Since $(d_i)_{i=0}^{\infty}$ is a non-decreasing sequence, we have

$$A_T^* \left(1 + T - S_T\right) \geq S_T.$$

Using AM-GM inequality we have $S_T \geq T\gamma_T$, where $\gamma_T = \left(\frac{d_0}{d_T}\right)^{\frac{1}{T}}$. Therefore,

$$A_T^* \geq \frac{T\gamma_T}{1 + T(1 - \gamma_T)},$$

and

$$\min_{0 \leq t \leq T} \frac{d_t}{\sum_{i=0}^{t-1} d_i} = \frac{1}{A_T^*} \leq \frac{\frac{1}{\gamma_T}\left(1 + T(1 - \gamma_T)\right)}{T}. \tag{9}$$

Using the inequality $1 - \frac{1}{x} \leq \log x$ (for any $x \leq 1$), we have $1 - \frac{1}{\gamma_T} \leq \log \gamma_T$. Substituting this into (9) completes the proof. ■

This Lemma has been established in [17, Lemma 3.2.1] and the proof included here for the reader's convenience.

Lemma 3 *For any $x \in \mathbb{R}^d$ we have $f(x) - f^* \leq \omega(v(x))$.*

Proof Note that $\langle \nabla f(x), x - x^* \rangle \geq 0$ becuase x^* is the minimizer of f. Now, consider the point $\bar{y}$ as follows

$$\bar{y} = x^* + v(x) \frac{g(x)}{\|g(x)\|_*}.$$

Thus, we have $\langle g(x), \bar{y} - x \rangle = 0$, and $\|\bar{y} - x^*\| = v(x)$. Therefore, $f(x) \leq f(y)$, and hence

$$f(x) - f^* \leq f(\bar{y}) - f^* \leq \omega(\|\bar{y} - x^*\|) = \omega(v(x)). \qquad \blacksquare$$

Appendix B. Analysis of Dual Averaging

Theorem 4 *In Algorithm Algorithm 1, for any $0 \leq k \leq T$, it holds that*

$$\sum_{i=0}^{k-1} a_i \langle g_i, x_i - x^* \rangle + \frac{\beta_k}{2} \|x_k - x^*\|^2 \leq \frac{\beta_k}{2} \|x_0 - x^*\|^2 + \sum_{i=0}^{k-1} \frac{a_i^2}{2\beta_i} \|g_i\|_*^2,$$

Proof

First, define $\psi_0(x) = \frac{\beta_0}{2}\|x - x_0\|^2$. Note that ψ_k is a β_k-strongly convex function and x_k is its minimizer. Hence, for any $x \in Q$, we have

$$\psi_k(x) \geq \psi_k(x_k) + \frac{\beta_k}{2} \|x - x_k\|^2 \tag{10}$$

Indeed,

$$\begin{aligned} \psi_{k+1}(x_{k+1}) &= \psi_k(x_{k+1}) + a_k \langle g_k, x_{k+1} - x_k \rangle + \frac{\beta_{k+1} - \beta_k}{2} \|x_{k+1} - x_0\|^2 \\ &\geq \psi_k(x_k) + \frac{\beta_k}{2} \|x_{k+1} - x_k\|^2 + a_k \langle g_k, x_{k+1} - x_k \rangle + \frac{\beta_{k+1} - \beta_k}{2} \|x_{k+1} - x_0\|^2, \end{aligned}$$

where the last inequality follows from (10). Hence,

$$\begin{aligned} \psi_{k+1}(x_{k+1}) &\geq \psi_k(x_k) + \frac{\beta_k}{2} \|x_{k+1} - x_k\|^2 + a_k \langle g_k, x_{k+1} - x_k \rangle \\ &\geq \psi_k(x_k) - \frac{a_k^2}{2\beta_k} \|g_k\|_*^2, \end{aligned}$$

Telescoping these inequalities and using the fact that $\psi_0(x_0) = 0$, we obtain

$$\psi_k(x_k) \geq -\sum_{i=0}^{k-1} \frac{a_i^2}{2\beta_i} \|g_i\|_*^2.$$

Finally, using (10), we complete our proof:

$$\begin{aligned} \sum_{i=0}^{k-1} a_i \langle g_i, x^* - x_i \rangle + \frac{\beta_k}{2} D_0^2 = \psi_k(x^*) &\geq \psi_k(x_k) + \frac{\beta_k}{2} \|x_k - x^*\|^2 \\ &\geq -\sum_{i=0}^{k-1} \frac{a_i^2}{2\beta_i} \|g_i\|_*^2 + \frac{\beta_k}{2} \|x_k - x^*\|^2, \end{aligned}$$

Rearranging, we get

$$\sum_{i=0}^{k-1} a_i \langle g_i, x_i - x^* \rangle + \frac{\beta_k}{2} \|x_k - x^*\|^2 \le \frac{\beta_k}{2} D_0^2 + \sum_{i=0}^{k-1} \frac{a_i^2}{2\beta_i} \|g_i\|_*^2.$$

Substituting $v_i = \frac{\langle g_i, x_i - x^* \rangle}{\|g_i\|_*}$, we get the claim. ■

Appendix C. Proof of Theorem 1

Lemma 5 *Consider Algorithm 1 using the coefficients defined in (3). Then we have the following inequality for all* $1 \le k \le T$,

$$r_k \le 2D_0 + \frac{1}{\sqrt{2}} \bar{r}_{k-1},$$

where $D_0 = \|x_0 - x^*\|$.

Proof Applying Theorem 4, dropping the nonnegative $\langle g_i, x_i - x^* \rangle$ from the left-hand side and rearranging, we obtain

$$d_k^2 \le D_0^2 + \frac{1}{\beta_k} \sum_{i=0}^{k-1} \frac{a_i^2}{\beta_i} \|g_i\|_*^2.$$

Substituting our choice of the coefficients given by (3), we get

$$d_k^2 \le D_0^2 + \frac{1}{4\sqrt{k+1}} \sum_{i=0}^{k-1} \frac{\bar{r}_i^2}{\sqrt{i+1}} \le D_0^2 + \frac{\bar{r}_{k-1}^2}{2}, \tag{11}$$

where we have used the fact that $\bar{r}_k$ is nondecreasing and, $\sum_{i=0}^{k-1} \frac{1}{\sqrt{i+1}} \le 2\sqrt{k}$. Extracting the square root from both sides of the above inequality we get,

$$d_k \le D_0 + \frac{1}{\sqrt{2}} \bar{r}_{k-1}.$$

Therefore,

$$r_k = \|x_k - x_0\| \le \|x_k - x^*\| + \|x_0 - x^*\| \le 2\|x_0 - x^*\| + \frac{1}{\sqrt{2}} \bar{r}_{k-1}.$$ ■

Lemma 6 *Consider Algorithm 1 using the coefficients defined in (3). Then, for all* $1 \le k \le T$,

$$\bar{r}_k \le 8R,$$

where $R = \max\{D_0, \bar{r}\}$.

Proof Hence,

$$\bar{r}_k \equiv \max\{r_k, \bar{r}\} \leq 2R + \frac{1}{\sqrt{2}}\bar{r}_{k-1}$$

As we need an upper bound for $\bar{r}_k$ and not r_k, we use induction here to prove that $\bar{r}_k \leq 8R$. Suppose that we know $\bar{r}_{k-1} \leq 8R$ and we prove this inequality holds for $\bar{r}_k$:

$$\bar{r}_k \leq 2R + \frac{1}{\sqrt{2}}\bar{r}_{k-1} \leq 2R + \frac{8}{\sqrt{2}}R \leq 8R. \qquad \blacksquare$$

Proof [proof of Theorem 1] Using Theorem 3, we get

$$f(\bar{x}_T^*) - f^* = \min_{0\leq i\leq T}(f(x_i) - f^*) \leq \min_{0\leq i\leq T}\omega(v_i) = \omega(v_T^*) \tag{12}$$

According to Theorem 4, for all $1 \leq k \leq T$,

$$\begin{aligned}\sum_{i=0}^{k-1} a_i v_i \|g_i\|_* &\leq \frac{\beta_k}{2}\|x_0 - x^*\|^2 - \frac{\beta_k}{2}\|x_k - x^*\|^2 + \sum_{i=0}^{k-1}\frac{a_i^2}{2\beta_i}\|g_i\|_*^2 \\ &\leq \beta_k r_k D_0 + \sum_{i=0}^{k-1}\frac{a_i^2}{2\beta_i}\|g_i\|_*^2.\end{aligned}$$

where we have used the fact that

$$\begin{aligned}\|x_0 - x^*\|^2 - \|x_k - x^*\|^2 &= (\|x_0 - x^*\| - \|x_k - x^*\|)(\|x_0 - x^*\| + \|x_k - x^*\|) \\ &\leq 2\|x_k - x_0\|\|x_0 - x^*\| = 2r_k D_0.\end{aligned}$$

Now let us define k as follows:

$$k = \operatorname*{argmin}_{1\leq k\leq T}\frac{\bar{r}_k}{\sum_{i=0}^{k-1}\bar{r}_i}.$$

Using (3) we get the following inequality:

$$\begin{aligned}\sum_{i=0}^{k-1}\bar{r}_i v_i &\leq 2\|x_k - x_0\|R\sqrt{k+1} + \frac{\sqrt{k}}{2}\bar{r}_k^2 \\ &\leq 2\bar{r}_k R\sqrt{k+1} + 4\bar{r}_k R\sqrt{k},\end{aligned}$$

where we have used (11) and Theorem 6. Applying Theorem 2 we obtain

$$\begin{aligned}v_T^* \leq \frac{\sum_{i=0}^{k-1}\bar{r}_i v_i}{\sum_{i=0}^{k-1}\bar{r}_i} &\leq \frac{\bar{r}_k}{\sum_{i=0}^{k-1}\bar{r}_i}\left(2R\sqrt{k+1} + 4R\sqrt{k}\right) \\ &\leq \frac{6R\sqrt{k+1}}{T}\left(\frac{\bar{r}_T}{\bar{r}}\right)^{\frac{1}{T}}\log\frac{e\bar{r}_T}{\bar{r}},\end{aligned}$$

therefore, using Theorem 6 we have

$$v_T^* \leq \frac{6R\sqrt{k+1}}{T}\left(\frac{\bar{r}_T}{\bar{r}}\right)^{\frac{1}{T}} \log\frac{e\bar{r}_T}{\bar{r}} \leq \frac{6R\sqrt{\frac{T+1}{T}}}{\sqrt{T}}\left(\frac{8R}{\bar{r}}\right)^{\frac{1}{T}} \log\frac{8eR}{\bar{r}} \tag{13}$$

$$\leq \frac{9R}{\sqrt{T}}\left(\frac{8R}{\bar{r}}\right)^{\frac{1}{T}} \log\frac{8eR}{\bar{r}}, \tag{14}$$

where we have used $\sqrt{\frac{T+1}{T}} \leq \frac{3}{2}$. To make the right-hand side $\leq \delta$, it suffices to ensure that the following two inequalities are satisfied:

$$T \geq \log\frac{8R}{\bar{r}}, \quad T \geq \frac{81e^2R^2}{\delta^2}\log^2\frac{8eR}{\bar{r}}.$$

To prove this claim, note that

$$\left(\frac{8R}{\bar{r}}\right)^{\frac{1}{T}} = \exp(\frac{\log\frac{8R}{\bar{r}}}{T}) \leq e \tag{15}$$

whenever $T \geq \log\frac{8R}{\bar{r}}$. Finally, using $T \geq \frac{81e^2R^2}{\delta^2}\log^2\frac{8eR}{\bar{r}}$ along with (15), we obtain

$$v_T^* \leq \frac{9R}{\sqrt{T}}\left(\frac{8R}{\bar{r}}\right)^{\frac{1}{T}} \log\frac{8eR}{\bar{r}} \leq \delta. \tag{16}$$

Together, (12) and (16) establish the proof. ■

Appendix D. Convergence of DADA on Various Problem Classes

D.1. Nonsmooth Lipschitz Functions

In this section, we assume that the function f in problem (1) is Lipschitz: for all $x, y \in \mathbb{R}^d$, it holds that

$$|f(x) - f(y)| \leq L_0\|x - y\|,$$

where $L_0 > 0$ is a fixed constant.

Lemma 7 *Assume that f is an L_0-Lipschitz function. Then, $\omega(t) \leq \epsilon$ for any given $\epsilon > 0$ whenever*

$$t \leq \frac{\epsilon}{L_0}.$$

Proof Indeed,

$$f(x) - f^* \leq L_0\|x - x^*\|,$$

Therefore, for any $t \geq 0$, we have

$$\omega(t) \leq L_0 t.$$

Making the right-hand side $\leq \epsilon$, we get the claim. ■

Theorem 8 *Consider problem 1 under the assumption that f is an L_0-Lipschitz function. Let Algorithm 1 with coefficients (3) be applied for solving this problem. Then, $f(x_T^*) - f^* \le \epsilon$ for any given $\epsilon > 0$ whenever*

$$T_\epsilon \ge \max\left\{\log\frac{8R}{\bar{r}}, \frac{81e^2R^2L_0^2}{\epsilon^2}\log^2\frac{8eR}{\bar{r}}\right\}.$$

Proof Using Theorem 1 for $\delta = \frac{\epsilon}{L_0}$, we obtain $v_{T_\epsilon} \le \frac{\epsilon}{L_0}$. Therefore,

$$f(x_{T_\epsilon}^*) - f^* \le \omega(v_{T_\epsilon}^*) \le \epsilon,$$

where we have used (2) and Theorem 7. ■

D.2. Lipschitz-Smooth Functions

Let us now consider the case when f is Lipschitz-smooth, meaning that for any $x, y \in \mathbb{R}^d$, the following inequality holds:

$$f(y) - f(x) \le \langle \nabla f(x), y - x\rangle + \frac{L_1}{2}\|y - x\|^2,$$

where $L_1 > 0$ is a fixed constant.

Lemma 9 *Assume that f is Lipschitz-smooth with constant L_1. Then, $\omega(t) \le \epsilon$ for any given $\epsilon > 0$ whenever*

$$t \le \min\left\{\sqrt{\frac{\epsilon}{L_1}}, \frac{\epsilon}{2\|\nabla f(x^*)\|_*}\right\}.$$

Proof Indeed,

$$\begin{aligned} f(x) - f(x^*) &\le \langle \nabla f(x^*), x - x^*\rangle + \frac{L_1}{2}\|x - x^*\|^2 \\ &\le \|\nabla f(x^*)\|_*\|x - x^*\| + \frac{L_1}{2}\|x - x^*\|^2. \end{aligned}$$

Hence, for any $t \ge 0$,

$$\omega(t) \le \frac{L_1}{2}t^2 + \|\nabla f(x^*)\|_* t.$$

To make the right-hand side $\le \epsilon$, it suffices to ensure that each of the two terms is $\le \frac{\epsilon}{2}$:

$$L_1 t^2 \le \frac{\epsilon}{2}, \qquad \|\nabla f(x^*)\|_* t \le \frac{\epsilon}{2}.$$

Solving this system of inequalities, we get the claim. ■

Theorem 10 *Consider problem 1 under the assumption that f is Lipschitz-smooth with constant L_1. Let Algorithm 1 with coefficients (3) be applied for solving this problem. Then, $f(x_T^*) - f^* \leq \epsilon$ for any given $\epsilon > 0$ whenever*

$$T_\epsilon \geq \max\left\{\log\frac{8R}{\bar{r}}, \frac{81e^2R^2}{h_\epsilon^2}\log^2\frac{8eR}{\bar{r}}\right\},$$

where $h_\epsilon = \min\{\sqrt{\frac{\epsilon}{L_1}}, \frac{\epsilon}{2\|\nabla f(x^*)\|_*}\}$.

Proof Using Theorem 1 for $\delta = h_\epsilon$, we obtain $v_{T_\epsilon} \leq h_\epsilon$. Therefore,

$$f(x_{T_\epsilon}^*) - f^* \leq \omega(v_{T_\epsilon}^*) \leq \epsilon,$$

where we have used (2) and Theorem 9. ■

D.3. Hölder-Smooth Functions

Let us now consider a more general case, when f is Hölder-smooth, meaning that for any $x, y \in \mathbb{R}^d$, it holds that

$$f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{H_\nu}{1+\nu}\|y - x\|^{1+\nu},$$

where $\nu \in [0, 1]$ and $H_\nu > 0$.

Lemma 11 *Assume that f is a Hölder-smooth function with constants ν and H_ν. Then, $\omega(t) \leq \epsilon$ for any given $\epsilon > 0$ whenever*

$$t \leq \min\left\{\left[\frac{(1+\nu)\epsilon}{2H_\nu}\right]^{\frac{1}{1+\nu}}, \frac{\epsilon}{2\|\nabla f(x^*)\|_*}\right\}.$$

Proof Indeed,

$$\begin{aligned} f(x) - f(x^*) &\leq \langle \nabla f(x^*), x - x^*\rangle + \frac{H_\nu}{1+\nu}\|x - x^*\|^{1+\nu} \\ &\leq \|\nabla f(x^*)\|_*\|x - x^*\| + \frac{H_\nu}{1+\nu}\|x - x^*\|^{1+\nu}. \end{aligned}$$

Hence, for any $t \geq 0$,

$$\omega(t) \leq \|\nabla f(x^*)\|_* t + \frac{H_\nu}{1+\nu}t^{1+\nu}$$

To make the right-hand side of the last inequality $\leq \epsilon$, it suffices to ensure that each of the two terms is $\leq \frac{\epsilon}{2}$:

$$\|\nabla f(x^*)\|_* t \leq \frac{\epsilon}{2}, \qquad \frac{H_\nu}{1+\nu}t^{1+\nu} \leq \frac{\epsilon}{2}.$$

Solving this system of inequalities, we get the claim. ■

Theorem 12 *Consider problem 1 under the assumption that f is a Hölder-smooth function with constants ν and H_ν. Let Algorithm 1 with coefficients (3) be applied for solving this problem. Then, $f(x_T^*) - f^* \leq \epsilon$ for any given $\epsilon > 0$ whenever*

$$T_\epsilon \geq \max\left\{\log \frac{8R}{\bar{r}}, \frac{81e^2R^2}{h_\epsilon^2} \log^2 \frac{8eR}{\bar{r}}\right\},$$

where $h_\epsilon = \min\{\left[\frac{(1+\nu)\epsilon}{2H_\nu}\right]^{\frac{1}{1+\nu}}, \frac{\epsilon}{2\|\nabla f(x^*)\|_*}\}$.

Proof Using Theorem 1 for $\delta = h_\epsilon$, we obtain $v_{T_\epsilon} \leq h_\epsilon$. Therefore,

$$f(x_{T_\epsilon}^*) - f^* \leq \omega(v_{T_\epsilon}^*) \leq \epsilon,$$

where we have used (2) and Theorem 11. ■

D.4. Functions with Lipschitz High-Order Derivative

In this section, we assume that function f in problem (1) has L_p-Lipschitz pth derivative. It means that for any $x, y \in \mathbb{R}^d$, the following inequality holds:

$$\|\nabla^p f(x) - \nabla^p f(y)\| \leq L_p\|x - y\|.$$

This implies the following global upper bound on the function value:

$$f(y) \leq f(x) + \sum_{i=1}^{p} \frac{1}{i!}\nabla^i f(x)\,[y-x]^i + \frac{L_p}{(p+1)!}\|y-x\|^{p+1}.$$

Lemma 13 *Assume that f has L_p-Lipschitz pth derivative. Then, $\omega(t) \leq \epsilon$ for any given $\epsilon > 0$ whenever*

$$t \leq \min\left\{\min_{1\leq i\leq p}\left[\frac{i!\,\epsilon}{(p+1)\|\nabla^i f(x^*)\|}\right]^{\frac{1}{i}}, \left[\frac{p!\,\epsilon}{L_p}\right]^{\frac{1}{p+1}}\right\}.$$

Proof Indeed,

$$\begin{aligned} f(x) - f^* &\leq \sum_{i=1}^{p} \frac{1}{i!}\nabla^i f(x^*)\,[x-x^*]^i + \frac{L_p}{(p+1)!}\|x-x^*\|^{p+1} \\ &\leq \sum_{i=1}^{p} \frac{1}{i!}\|\nabla^i f(x^*)\|\|x-x^*\|^i + \frac{L_p}{(p+1)!}\|x-x^*\|^{p+1}. \end{aligned}$$

Therefore, for any $t \geq 0$, we have

$$\begin{aligned} \omega(t) &\leq \sum_{i=1}^{p} \frac{1}{i!}\|\nabla^i f(x^*)\|t^i + \frac{L_p}{(p+1)!}t^{p+1} \\ &\leq \sum_{i=1}^{p} \frac{\epsilon}{p+1} + \frac{\epsilon}{p+1} = \epsilon \end{aligned}$$

To make the right-hand side $\leq \epsilon$, it suffices to ensure that each of the following inequalities holds:

$$\forall_{1\leq i\leq p}: \frac{1}{i!}\|\nabla^i f(x^*)\|t^i \leq \frac{\epsilon}{p+1}, \qquad \frac{L_p}{(p+1)!}t^{p+1} \leq \frac{\epsilon}{p+1}$$

Solving this system of inequalities, we get the claim. ■

Theorem 14 *Consider problem 1 under the assumption that f has L_p-Lipschitz pth derivative. Let Algorithm 1 with coefficients (3) be applied for solving this problem. Then, $f(x_T^*) - f^* \leq \epsilon$ for any given $\epsilon > 0$ whenever*

$$T_\epsilon \geq \max\left\{\log\frac{8R}{\bar{r}}, \frac{81e^2R^2}{h_\epsilon^2}\log^2\frac{8eR}{\bar{r}}\right\},$$

where

$$h_\epsilon = \min\left\{\min_{1\leq i\leq p}\left[\frac{i!\,\epsilon}{(p+1)\|\nabla^i f(x^*)\|}\right]^{\frac{1}{i}}, \left[\frac{p!\,\epsilon}{L_p}\right]^{\frac{1}{p+1}}\right\}.$$

Proof Using Theorem 1 for $\delta = h_\epsilon$, we obtain $v_{T_\epsilon} \leq h_\epsilon$. Therefore,

$$f(x_{T_\epsilon}^*) - f^* \leq \omega(v_{T_\epsilon}^*) \leq \epsilon,$$

where we have used (2) and Theorem 13. ■

D.5. Quasi-Self-Concordant Functions

In this section, we assume that the function f in problem (1) is Quasi-Self-Concordant (QSC), meaning that it is three times continuously differentiable and for any $x \in \mathbb{R}^d$ and arbitrary directions $u, v \in \mathbb{R}^d$ it holds that

$$\nabla^3 f(x)[u,u,v] \leq M\langle\nabla^2 f(x)u,u\rangle\|v\|,$$

with parameter $M \geq 0$.

The following lemma provides an important global upper bound on the function value.

Lemma 15 [7, Lemma 2.7] *Let f be QSC with the parameter M. Then for any x, y the following inequality holds*

$$f(y) \leq f(x) + \langle\nabla f(x), y-x\rangle + \langle\nabla^2 f(x)(y-x), y-x\rangle\varphi(M\|y-x\|),$$

where $\varphi(t) := \frac{e^t-t-1}{t^2}$.

Lemma 16 *Assume that f is a QSC function with constant M. Then, $\omega(t) \leq \epsilon$ for any given $\epsilon > 0$ whenever*

$$t \leq \min\left\{\frac{\epsilon}{2\|f(x^*)\|_*}, \sqrt{\frac{\epsilon}{2\|\nabla^2 f(x^*)\|}}, \frac{1}{M}\right\}.$$

Proof Since φ is an increasing function, if $M\|x - x^*\| \leq 1$, we can estimate

$$\varphi(M\|y - x\|) \leq \varphi(1) = e - 2 \leq 1.$$

Therefore, according to Theorem 15

$$\begin{aligned} f(x) - f^* &\leq \langle \nabla f(x^*), x - x^* \rangle + \langle \nabla^2 f(x^*)(x - x^*), x - x^* \rangle \varphi(M\|x - x^*\|) \\ &\leq \langle \nabla f(x^*), x - x^* \rangle + (e-2)\langle \nabla^2 f(x^*)(x - x^*), x - x^* \rangle \\ &\leq \|\nabla f(x^*)\|_* \|x - x^*\| + \|\nabla^2 f(x^*)\| \|x - x^*\|^2, \end{aligned}$$

Hence, for any $0 \leq t \leq \frac{1}{M}$,

$$\begin{aligned} \omega(t) &\leq \|\nabla f(x^*)\|_* \|x - x^*\| + \|\nabla^2 f(x^*)\| \|x - x^*\|^2 \\ &\leq \|\nabla f(x^*)\|_* t + \|\nabla^2 f(x^*)\| t^2 \end{aligned}$$

To make the right-hand side $\leq \epsilon$, it suffices to ensure that each of the two terms is $\leq \frac{\epsilon}{2}$:

$$\|\nabla f(x^*)\|_* t \leq \frac{\epsilon}{2}, \qquad \|\nabla^2 f(x^*)\| t^2 \leq \frac{\epsilon}{2}.$$

Solving this system of inequalities, we get the claim. ■

Theorem 17 *Consider problem 1 under the assumption that f is QSC with constant M. Let Algorithm 1 with coefficients (3) be applied for solving this problem. Then, $f(x_T^*) - f^* \leq \epsilon$ for any given $\epsilon > 0$ whenever*

$$T_\epsilon \geq \max\left\{ \log \frac{8R}{\bar{r}}, \frac{81e^2R^2}{h_\epsilon^2} \log^2 \frac{8eR}{\bar{r}} \right\},$$

where $h_\epsilon = \min\{\frac{\epsilon}{2\|f(x^*)\|_*}, \sqrt{\frac{\epsilon}{2\|\nabla^2 f(x^*)\|}}, \frac{1}{M}\}$.

Proof Using Theorem 1 for $\delta = h_\epsilon$, we obtain $v_{T_\epsilon} \leq h_\epsilon$. Therefore,

$$f(x_{T_\epsilon}^*) - f^* \leq \omega(v_{T_\epsilon}^*) \leq \epsilon,$$

where we have used (2) and Theorem 16. ■

D.6. (L_0, L_1)-Smooth Functions

Definition 18 *Let us now consider the case when f is (L_0, L_1)-smooth, meaning that for any $x \in \mathbb{R}^d$,*

$$\|\nabla^2 f(x)\| \leq L_0 + L_1 \|\nabla f(x)\|_*,$$

where $L_0, L_1 \geq 0$ are fixed constants.

Lemma 19 **[24, Lemma 2.2]** *Let f be a (L_0, L_1)-smooth and x, y be arbitrary points, then the following inequality holds*

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L_0 + L_1 \|\nabla f(x)\|_*}{L_1^2} \varphi(L_1 \|y - x\|),$$

where $\varphi(t) = e^t - t - 1$.

Lemma 20 *Assume that f is a (L_0, L_1)-smooth function. Then, $\omega(t) \leq \epsilon$ for any given $\epsilon > 0$ whenever*

$$t \leq \min\left\{ \frac{\epsilon}{2\|f(x^*)\|_*}, \sqrt{\frac{2\epsilon}{3(L_0 + L_1\|\nabla f(x^*)\|_*)}}, \frac{1}{L_1} \right\}.$$

Proof Since φ is an increasing function, if $L_1\|x - x^*\| \leq 1$, we can estimate

$$\varphi(L_1\|y - x\|) \leq \frac{3}{4} L_1^2 \|y - x\|^2.$$

Therefore, according to Theorem 19

$$\begin{aligned} f(x) - f^* &\leq \langle \nabla f(x^*), x - x^* \rangle + \frac{3(L_0 + L_1\|\nabla f(x^*)\|_*)}{4} \|x - x^*\|^2 \\ &\leq \|\nabla f(x^*)\|_* \|x - x^*\| + \frac{3(L_0 + L_1\|\nabla f(x^*)\|_*)}{4} \|x - x^*\|^2, \end{aligned}$$

Hence, for any $0 \leq t \leq \frac{1}{L_1}$,

$$\begin{aligned} \omega(t) &\leq \|\nabla f(x^*)\|_* \|x - x^*\| + \frac{3(L_0 + L_1\|\nabla f(x^*)\|_*)}{4} \|x - x^*\|^2 \\ &\leq \|\nabla f(x^*)\|_* t + \frac{3(L_0 + L_1\|\nabla f(x^*)\|_*)}{4} t^2 \end{aligned}$$

To make the right-hand side of the last inequality $\leq \epsilon$, it suffices to ensure that each of the two terms is $\leq \frac{\epsilon}{2}$:

$$\|\nabla f(x^*)\|_* t \leq \frac{\epsilon}{2}, \qquad \frac{3(L_0 + L_1\|\nabla f(x^*)\|_*)}{4} \leq \frac{\epsilon}{2}.$$

Solving this system of inequalities, we get the claim. ■

Theorem 21 *Consider problem 1 under the assumption that f is an (L_0, L_1)-smooth function. Let Algorithm 1 with coefficients (3) be applied for solving this problem. Then, $f(x_T^*) - f^* \leq \epsilon$ for any given $\epsilon > 0$ whenever*

$$T_\epsilon \geq \max\left\{ \log\frac{8R}{\bar{r}}, \frac{81e^2R^2}{h_\epsilon^2} \log^2\frac{8eR}{\bar{r}} \right\},$$

where $h_\epsilon = \min\{\frac{\epsilon}{2\|f(x^*)\|_*}, \sqrt{\frac{2\epsilon}{3(L_0+L_1\|\nabla f(x^*)\|_*)}}, \frac{1}{L_1}\}$.

Proof Using Theorem 1 for $\delta = h_\epsilon$, we obtain $v_{T_\epsilon} \leq h_\epsilon$. Therefore,

$$f(x_{T_\epsilon}^*) - f^* \leq \omega(v_{T_\epsilon}^*) \leq \epsilon,$$

where we have used (2) and Theorem 20. ■