COMMUNICATION-EFFICIENT DISTRIBUTED COORDINATE-DESCENT METHODS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a distributed block-coordinate method, where coordinates are partitioned across clients and communication is the main bottleneck. Each client performs multiple local updates between communication rounds to approximately solve a local problem, using Gradient Descent or an accelerated version as local solvers for efficiently minimizing the gradient norm of the local problem. We introduce cross-coupling M, which measures how far f is from a fully separable function, with $M = 0$ when f is fully separable. For nonconvex but coordinate-wise convex functions and for convex functions, we establish communication complexity bounds proportional to M. Under a Polyak–Łojasiewicz condition, our method converges linearly. We compare our method with baselines and explain when it is more communication efficient. We also develop an accelerated version; for convex objectives, it achieves a communication complexity bound proportional to $\sqrt{M}$. Alongside communication complexity, we bound local computation for each problem class. We also introduce a line search method that automatically estimates the constant M in an efficient way resulting in strong practical performance. Finally, we show that the proposed accelerated method is optimal (in terms of communication complexity) by establishing a lower bound valid for any first-order oracle distributed coordinate-descent method.

1 INTRODUCTION

Coordinate descent (CD) has a long history, from early quadratic-programming procedures to its modern variants (Hildreth, 1957; Luo & Tseng, 1992; Grippo & Sciandrone, 2000; Nesterov, 2012; Beck & Tetruashvili, 2013; Richtárik & Takáč, 2014; Wright, 2015). In its sequential form, CD updates one coordinate or block at each iteration, keeping the others fixed. Each update uses only the partial gradient for the selected coordinate. CD methods have been used to solve optimization problems with sparse solutions, high-dimensional linear and logistic regression, support-vector machines, dual optimization problems in machine learning (Wright, 2015), and, more recently, for Vertical Federated Learning (VFL) (Yang et al., 2019; Liu et al., 2022; 2024). Consecutive CD iterates differ in only one coordinate. For some classes of functions, this makes computing partial gradients much cheaper than the full gradient. As a result, CD can minimize these functions faster than gradient descent (GD) (Nesterov, 2012).

Distributed optimization has become a standard approach for large-scale problems (Verbraeken et al., 2020; Kairouz et al., 2021). A common setting is to minimize $f(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x})$, where each f_i is the objective for the data stored at client i. A central server and compute nodes, called clients, communicate over a network. Each function f_i is accessible only to client i, so the server must communicate with that client to obtain any information about it. A communication round can be much more expensive than local computation. Algorithms therefore often perform several computation steps on each client (local steps) before communicating (McMahan et al., 2017; Stich, 2019; Kairouz et al., 2021).

In this setting, however, every client maintains a copy of the full vector $\mathbf{x} \in \mathbb{R}^d$. This is impractical when storing or optimizing the full vector exceeds a client's memory or computational capacity. Many standard machine learning problems, such as least-squares regression and logistic regression, take the form $f(\mathbf{x}) = \Phi(A\mathbf{x})$, where $\mathbf{A}$ is the data matrix, and Φ is a convex loss function. Assuming Φ is separable across data points, we can write $f(x) = \sum_{i=1}^{n} \Phi_i(A_{i,:}x)$, where $A_{i,:}$ denotes the i-th row and Φ_i is the corresponding loss. This gives exactly the sum-of-functions decomposition

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

described above. However, this decomposition is not suitable when client i can only locally access only a subset of features (column $\mathbf{A}_{:,i}$) rather than data points (row $\mathbf{A}_{i,:}$), as in VFL (Liu et al., 2022). It could even be impossible when each client cannot store the full high-dimensional vector in huge-scale optimization Nesterov (2012). This motivates an alternative: divide the parameter vector $\mathbf{x} = (\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)})$ into lower-dimensional blocks corresponding to the features. The same objective then takes the form $f(\mathbf{x}) = \Phi(\sum_{i=1}^{n} \mathbf{A}_{:,i}\mathbf{x}^{(i)})$, where $\mathbf{A}_{:,i}$ contains the features held by client i. Here the sum is inside Φ, rather than over local functions Φ_i.

With this coordinate partitioning, distributed CD is defined in a server–client setting where only client i can access the partial-gradient oracle for coordinate i. The server must communicate with client i to query its oracle. Between communication rounds, however, each client may perform any number of local oracle calls and local computations before returning its result to the server. This setting includes existing distributed CD methods (Mahajan et al., 2017; Richtárik & Takáč, 2016a; Mareček et al., 2015). The latter two use exactly one local step.

Surprisingly, when each client takes exactly one gradient step per round, the combined update is exactly a GD step (in a weighted norm if different step sizes are used for different coordinates), with the full-gradient computation distributed across clients. More precisely, instead of computing the full gradient directly, one can distribute the computational load by having each client compute its partial gradient, then assemble them into the full gradient. This idea of computing coordinate updates in parallel is already well established in the literature on parallel CD methods (Bradley et al., 2011; Scherrer et al., 2012; Necoara & Clipici, 2013; Liu & Wright, 2015; Richtárik & Takáč, 2016b; Fercoq & Richtárik, 2015; Hsieh et al., 2015; Peng et al., 2016).

In Parallel CD methods, at each iteration, sample a subset or all of the coordinate blocks and assign them to clients. At communication round t, all clients compute partial gradients, $\nabla_i f(\mathbf{x}_t)$ in parallel, an then they are combined to obtain the next iterate, $\mathbf{x}_{t+1}$. The key problem is that clients communicate after just one partial gradient step, usually over a network, leaving them idle most of the time. Therefore, these methods do not exploit the potential for longer and more extensive local computations to reduce communication rounds. This design, however, is not surprising, as many parallel CD methods were developed for shared-memory systems (Liu & Wright, 2015; Peng et al., 2016), where communication is relatively inexpensive compared with local computation. There is, therefore, little incentive to perform more than one local step, and most existing parallel CD methods perform just one. In distributed settings, however, this can lead to many unnecessary and costly communications. For example, when $f(x) = \sum_{i=1}^{n} f_i(x^{(i)})$ is fully separable, these methods use multiple communication rounds even though independent local minimization of each $f_i(x^{(i)})$, followed by a single round of communication, suffices. Even distributed methods such as Hydra and DBCD (Richtárik & Takáč, 2016a; Mareček et al., 2015) use only one local step per round. A more suitable weighted norm can improve their performance over standard GD, but these two method face the same communication limitation.

Distributed CD is also used in VFL (Trofimov & Genkin, 2016; Chen et al., 2020; Liu et al., 2022; Castiglia et al., 2024). Among the methods discussed above, VFL methods are, to the best of our knowledge, the only ones that use additional local steps to reduce communication. In particular, FedBCD (Liu et al., 2022) each client to perform multiple local steps on each client, between communication rounds. Although these methods have been studied empirically and perform well in practice, their theoretical guarantees remain limited in an important respect: to the best of our knowledge, they do not establish when additional local steps reduce communication compared with baselines such as GD and parallel CD.

This raises a natural question: can doing more computing on clients and taking multiple local gradient steps reduce communication in distributed CD methods? Simply taking independent local steps and then synchronizing does not guarantee convergence, so these local steps must be designed carefully. The benefit of more local steps also depends on the function. An extreme example is a fully separable function, which can be minimized locally with one communication round at the end. This suggests the need for a measure to determine how beneficial additional local steps can be for specific classes of functions.

Contributions. We make the following contributions: **A measure of near-separability (Section 3).** We define M through cross-block gradient sensitivity, estimate it for quadratics, and derive bounds

for generalized linear models. **Proximal-Point Coordinate Descent Scheme (PiCS) (Section 4).** PiCS scales regularization with M and permits multiple inexact local steps between synchronizations. In practice, a line search adapts this regularization without knowing M. For fully block-separable objectives, PiCS needs one communication round. **Communication guarantees (Sections 4.1 to 4.3).** We prove $\mathcal{O}(M/\varepsilon^2)$ communication rounds for an ε-stationary point, $\mathcal{O}(MD_0^2/\varepsilon)$ for convex objectives, and linear convergence under a Polyak–Łojasiewicz condition. These guarantees scale with cross-coupling rather than local problem difficulty. **Acceleration, lower bound, and optimality (Sections 5 and 6).** Accelerated PiCS achieves $\mathcal{O}(R_0\sqrt{M/\varepsilon})$ communication complexity for convex functions matching our lower bound. **Comparison with baselines (Section 7).** Under one distributed setup, we compare PiCS with other baselines, lustrating how it is more communication efficient in comparison to them.

2 Setting and Preliminaries

Setting. We are interested in minimizing a differentiable function

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

in a distributed setting, where we partition $\mathbf{x} \in \mathbb{R}^d$ into n blocks, $\mathbf{x} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)})$, with $\mathbf{x}^{(i)} \in \mathbb{R}^{d_i}$ and $d = \sum_{i=1}^n d_i$. At each communication round, the server shares $\bar{x}$ with all clients, which then perform independent local steps starting from this point. Let $\mathbf{U}_i \in \mathbb{R}^{d\times d_i}$ be the matrix transforming any $\mathbf{h} \in \mathbb{R}^{d_i}$ into $\mathbf{U}_i\mathbf{h} = (0, \dots, 0, \mathbf{h}, 0, \dots, 0) \in \mathbb{R}^d$, with $\mathbf{h}$ placed at the i-th position. During local computation, client i can only evaluate $\nabla_i f(\bar{\mathbf{x}} + U_i h)$, where $h \in \mathbb{R}^{d_i}$ is arbitrary and all other coordinates remain fixed at $\bar{x}$. Access to other partial gradients requires communication. The clients perform local computations using partial gradients, and then they exchange their local solutions with the server. The server has no direct access to the objective or the local oracles. It then aggregates these solutions to obtain a new iterate $\mathbf{x}_+$ with $f(\mathbf{x}_+) \le f(\bar{\mathbf{x}})$, completing the communication round. We use the following metrics for comparing the efficiency of methods and baselines. Define the accuracy measure $\Delta(\mathbf{x}) \coloneqq \|\nabla f(\mathbf{x})\|$ for nonconvex objectives and $\Delta(\mathbf{x}) \coloneqq f(\mathbf{x}) - f^\star$ for convex objectives. For a target accuracy $\varepsilon > 0$, the goal is to find $\mathbf{x}_T$ such that $\Delta(\mathbf{x}_T) < \varepsilon$. The *communication complexity* and *parallel bottleneck complexity* are

$$R(\varepsilon) \coloneqq \inf\{T \ge 0 : \Delta(\mathbf{x}_T) < \varepsilon\}, \qquad W(\varepsilon) \coloneqq \textstyle\sum_{t=0}^{R(\varepsilon)-1} \max_{1\le i\le n} K_{t,i}. \tag{2}$$

Here, Here, $K_{t,i}$ is the number of local oracle calls made by client i in round t. Thus, $R(\varepsilon)$ counts communication rounds, and $W(\varepsilon)$ counts the local work on the slowest client. Note that, as communication is the main bottleneck in distributed settings, the primary metric is $R(\epsilon)$, and we are primarily interested in methods that achieve a smaller $R(\epsilon)$. We discuss its precise formalization and further details in Section B.

Notation. $\|\cdot\|$ denotes the standard Euclidean norm on $\mathbb{R}^d$ and, for matrices, the spectral norm. Together, these matrices decompose the identity matrix $I_d \in \mathbb{R}^{d\times d}$ by columns: $I_d = [\mathbf{U}_1, \dots, \mathbf{U}_n]$. In particular, $\mathbf{U}_i^\top \mathbf{U}_j = 0$ for $i \ne j$ and $\mathbf{U}_i^\top \mathbf{U}_i = I_{d_i}$, so $\mathbf{x} = \sum_{i=1}^n \mathbf{U}_i \mathbf{x}^{(i)}$ and $\mathbf{x}^{(i)} = \mathbf{U}_i^\top \mathbf{x}$. We we define $\mathbf{P}_{-i} \coloneqq I_d - \mathbf{U}_i\mathbf{U}_i^\top \in \mathbb{R}^{d\times d}$ as a orthogonal projector. For the gradient vector $\nabla f(\mathbf{x}) \in \mathbb{R}^d$, we use the notation $\nabla_i f(\mathbf{x}) \in \mathbb{R}^{d_i}$ to denote its i-th block. For a matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$, we define its *block off-diagonal part* by $\mathrm{off}(\mathbf{A}) \coloneqq \mathbf{A} - \sum_{i=1}^n \mathbf{U}_i\mathbf{U}_i^\top \mathbf{A}\, \mathbf{U}_i\mathbf{U}_i^\top$. We assume that a minimizer $\mathbf{x}^\star \in \arg\min_{\mathbf{x}} f(\mathbf{x})$ exists.

3 Cross-coupling

As the coordinate blocks become more independent, clients can make more local progress and synchronize less often. We then need to measure "how far" an objective $f(\mathbf{x})$ is from being fully block-separable under the fixed partition $\mathbf{x} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)})$. Thus, we introduce *Cross-coupling* (M), a measure of block separability, even when it is implicit. In the following section, we show how M naturally appears across all communication complexities of our proposed method.

Assumption 3.1. The function f is *cross-coupled* with constant $M \geq 0$, i.e., for all $\mathbf{x}, \mathbf{h} \in \mathbb{R}^d$, we have

$$\sum_{i=1}^{n} \|\nabla_i f(\mathbf{x} + \mathbf{P}_{-i}\mathbf{h}) - \nabla_i f(\mathbf{x})\|^2 \;\leq\; M^2 \, \|\mathbf{h}\|^2 . \tag{3}$$

Note that, for a separable function $f(x) = \sum_i f_i(x)$, we have $M = 0$.

The cross-coupling assumption (Assumption 3.1) gives a global separable upper bound for f. For any $\mathbf{x}, \mathbf{h} \in \mathbb{R}^d$,

$$f(\mathbf{x} + \mathbf{h}) \leq f(\mathbf{x}) + \sum_{i=1}^{n} \big[f(\mathbf{x} + \mathbf{U}_i \mathbf{h}^{(i)}) - f(\mathbf{x})\big] + \frac{M}{2} \|\mathbf{h}\|^2 =: \Omega(\mathbf{x}, \mathbf{x} + \mathbf{h}). \tag{4}$$

Thus, $f(\mathbf{x} + \mathbf{h}) \leq \Omega(\mathbf{x}, \mathbf{x} + \mathbf{h})$; see Section C.1 for the derivation. The second term, which uses function-value differences, is quite non-standard compared with smoothness-based upper bounds, as used in ESO-PCDM (Richtárik & Takáč, 2016b), Safaryan et al. (2021), and CoCoA (Smith et al., 2018). While these bounds use the linear approximation $f(x) + \langle \nabla f(x), h \rangle$, we use the function variations along each block. Such an approximation is therefore more precise for functions that are almost separable; consider a separable function $f(x) = \sum_{i=1}^{n} f_i(x^{(i)})$, we have $M = 0$ and $\Omega(x, y) = f(y)$, therefore minimizing the bound solves the original problem exactly. Smoothness-based bounds, however, are still upper bounds. This key property of Ω allows our method to communicate efficiently when M is small, with communication bounds that depend on M (see Sections 4.1 to 4.3 and 5).

We next show how M relates to the Hessian and smoothness constants. First, for convex L-smooth f with $L > 0$, we have $\boxed{M \leq L}$; see Lemma C.1 in the appendix. We next establish the following bounds on M. For twice differentiable f, M bounds the block off-diagonal Hessian norm everywhere.

Theorem 3.2. *Let $f \colon \mathbb{R}^d \to \mathbb{R}$ be twice differentiable. If f satisfies Assumption 3.1 with constant M, then $\left\|\mathrm{off}\left(\nabla^2 f(\mathbf{x})\right)\right\| \leq M$ for every $\mathbf{x} \in \mathbb{R}^d$.*

For positive-definite quadratics, this lower bound on M is exact.

Theorem 3.3. *For quadratic function $f(\mathbf{x}) = \frac{1}{2}\langle \mathbf{A}\mathbf{x}, \mathbf{x}\rangle + \langle \mathbf{b}, \mathbf{x}\rangle + c$ with $\mathbf{A} \succ 0$, we have exact equality $M = \|\mathrm{off}(\mathbf{A})\|_2$.*

For general functions, we obtain the following upper bound.

Theorem 3.4. *Let $f \colon \mathbb{R}^d \to \mathbb{R}$ be twice differentiable and suppose $\left\|\mathrm{off}(\nabla^2 f(\mathbf{x}))\right\| \leq \widetilde{M}$ for every $\mathbf{x} \in \mathbb{R}^d$, for some constant $\widetilde{M} \geq 0$. Then f satisfies Assumption 3.1, and its smallest cross-coupling constant satisfies $M \leq \sqrt{n}\,\widetilde{M}$.*

Unfortunately, this upper bound includes an extra factor $\sqrt{n}$, where n is the number of the blocks, and it is unclear whether this factor can be avoided in general. With additional structure on the Hessian, the next lemma gives a bound without this factor.

Lemma 3.5. *Let f be twice differentiable. Suppose there exists $\mathbf{Q} \in \mathbb{R}^{d\times d}_{+}$ such that $|\nabla^2 f(\mathbf{x})| \leq \mathbf{Q}$ entry-wise for all $\mathbf{x} \in \mathbb{R}^d$. Then, $M \leq \|\mathrm{off}(\mathbf{Q})\|_2$.*

Consider a generalized linear model (GLM) $f(\mathbf{x}) = \Phi(\mathbf{A}\mathbf{x}) = \sum_{j=1}^{m} \phi(\langle \mathbf{a}_j, \mathbf{x}\rangle - b_j)$, where $\mathbf{A} \in \mathbb{R}^{m\times d}$ has rows $\mathbf{a}_j^\top$, $\mathbf{b} = (b_1, \ldots, b_m)^\top \in \mathbb{R}^m$, and ϕ is differentiable. Common choices are the squared loss $\phi(t) = \frac{1}{2}t^2$, the Huber loss $\phi(t) = \frac{t^2}{2\delta}$ if $|t| \leq \delta$ and $\phi(t) = |t| - \frac{\delta}{2}$ otherwise ($\delta > 0$) for robust regression, and the logistic loss $\phi(t) = \log(1 + e^t)$ for binary classification.[1] For the GLM function above, assume $\mathbf{A} \in \mathbb{R}^{m\times d}_{+}$ and ϕ is convex and L_ϕ-smooth. Using Lemma 3.5, we obtain $M \;\leq\; L_\phi \left\|\mathrm{off}(\mathbf{A}^\top \mathbf{A})\right\|_2$. It isrequires ϕ to be convex and L_ϕ-smooth which is statisfied by all three of our examples with $L_\phi = 1$, $\frac{1}{\delta}$, and $\frac{1}{4}$, respectively.

Note that this bound relates$M$to the off-diagonal part of the covariance matrix (more precisely, the Gram matrix), excluding each feature's alignment with itself. As the features become closer to orthogonal, the coupling weakens, and the problem becomes closer to separable, even when this structure is not explicit in the objective. Without the positivity assumption, we can instead use Theorem 3.4, which bounds M with an additional$\sqrt{n}$ factor.

[1] Labels are absorbed into $\mathbf{a}_j$ and $\mathbf{b} = \mathbf{0}$; see Appendix C.8.

4 PROXIMAL-POINT COORDINATE DESCENT SCHEME

We begin by minimizing the upper bound $\Omega(\bar{\mathbf{x}}, \mathbf{y})$ in (4) at the anchor point $\bar{\mathbf{x}}$. It is then straightforward to verify that this is equivalent to minimizing the proximal-point subproblem for $\lambda > 0$,

$$\min_{\mathbf{x}^{(i)}} \left\{ g_{i,\bar{\mathbf{x}}}(\mathbf{x}^{(i)}) \;:=\; f(\bar{\mathbf{x}} + \mathbf{U}_i(\mathbf{x}^{(i)} - \bar{\mathbf{x}}^{(i)})) + \frac{\lambda}{2} \|\mathbf{x}^{(i)} - \bar{\mathbf{x}}^{(i)}\|^2 \right\}, \tag{5}$$

for each block $i \in \{1, \dots, n\}$, where $\mathbf{x}^{(i)}, \bar{\mathbf{x}}^{(i)} \in \mathbb{R}^{d_i}$.At each round, client i minimizes its local objective $g_{i,\bar{x}}(x^{(i)})$, with all clients working in parallel. We call this method the Proximal-Point Coordinate Descent Scheme (PiCS). Since finding an exact minimizer can be expensive, we allow inexact solutions, also making the local solvers more robust. The following inexactness condition suffices to retain the communication rates of exact solves up to constant factors. Let $x_+^{(i)} \in \mathbb{R}^{d_i}$ be any point satisfying:

$$\|\nabla g_{i,\bar{\mathbf{x}}}(\mathbf{x}_+^{(i)})\| \;\le\; \frac{\lambda}{2} \|\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)}\|. \tag{6}$$

Combining the coordinate outputs gives the update $\mathbf{x}_+ := (\mathbf{x}_+^{(1)}, \dots, \mathbf{x}_+^{(n)}) \in \mathbb{R}^d$. For $\lambda \ge 2M$ and coordinate-wise convex f, this condition guarantees the decrease stated below. We derive our convergence guarantees from this key inequality.

Assumption 4.1. For each $\mathbf{x} \in \mathbb{R}^d$ and each $i \in \{1, \dots, n\}$, the function $\mathbf{h} \mapsto f(\mathbf{x} + \mathbf{U}_i\mathbf{h})$ is convex on $\mathbb{R}^{d_i}$.

Lemma 4.2. *Assume coordinate-wise convexity (Assumption 4.1) and global cross-coupling with constant M (Assumption 3.1). Let $\mathbf{x}_+$ be one PiCS step from any $\bar{\mathbf{x}} \in \mathbb{R}^d$ with $\lambda > 0$ satisfying $\lambda \ge 2M$ and inexactness condition* (6)*. Then*

$$f(\mathbf{x}_+) \le f(\bar{\mathbf{x}}) - \frac{\lambda}{4} \|\mathbf{x}_+ - \bar{\mathbf{x}}\|^2 \le f(\bar{\mathbf{x}}) - \frac{1}{20\lambda} \|\nabla f(\mathbf{x}_+)\|^2. \tag{7}$$

As described Section 2 PiCS runs many local steps on all clients to approximately solve the local problems defined above. Clients then return their local solutions and server combines to form $\mathbf{x}_+$. Starting from $\mathbf{x}_0$, PiCS repeats this update with $\bar{\mathbf{x}} = \mathbf{x}_t$ and $\mathbf{x}_{t+1} = \mathbf{x}_+$ for $t = 0, 1, \dots, T-1$. The server sends each client i the information needed to query its partial gradient oracle ($\mathrm{O}_i^{\mathbf{x}_t}$). In general, the regularization coefficient can vary between iterations, so we denote it by λ_t. When $M > 0$ is known, the conceptually ideal fixed choice $\lambda_t = 2M$ gives the largest decrease guaranteed by the decrease lemma (Lemma 4.2). Algorithm 1 summarizes the method. Note that any local solver satisfying (6) can be used.

Line search. Since M is rarely known a priori, we use the doubling/halving line search of Nesterov (2013, Section 3). At iteration t, clients compute a candidate $\mathbf{x}_{t+1}$ satisfying (6) at $\mathbf{x}_t$ with a trial coefficient $\hat{\lambda}_t > 0$. The server accepts $\mathbf{x}_{t+1}$ if $f(\mathbf{x}_t) - f(\mathbf{x}_+) \ge \frac{1}{20\lambda_t} \|\nabla f(\mathbf{x}_+)\|^2$; otherwise, it doubles $\hat{\lambda}_t$ and retries. After acceptance, start the next trial at $\hat{\lambda}_{t+1} = \lambda_t/2$. Lemma 4.2 guarantees acceptance whenever $\lambda_t \ge \bar{\lambda} := 2M$, although smaller values may also pass. Starting from $0 < \hat{\lambda}_0 \le \bar{\lambda}$, every accepted coefficient satisfies $\lambda_t \le 2\bar{\lambda}$, since only trials below $\bar{\lambda}$ can be rejected. Counting all trials and all rounds, it is known for this standard line search, $N_{\mathrm{ls}} \le 2(T+1) + \log_2 \frac{\bar{\lambda}}{\hat{\lambda}_0}$. Thus, line search preserves the communication complexity up to a constant factor and an additive logarithmic cost. For simplicity, throughout the rest of the paper, we use the unified coefficient $\bar{M} = \Theta(M)$ such that $\lambda_t \le \bar{M}$ in both cases of fixed choice of λ_t and line search ($2M$ and $4M$, respectively).

4.1 MINIMIZING GRADIENT NORM

We first consider coordinate-wise convex functions, which need not be globally convex. Our goal is to find a point $\hat{\mathbf{x}}$ with $\|\nabla f(\hat{\mathbf{x}})\| \le \varepsilon$.

A standard example is matrix factorization, $\|\prod_{i=1}^n \mathbf{W}_i - \mathbf{A}\|_F^2$, which is convex in each factor with the others fixed, but generally nonconvex jointly. Summing the bound from the decrease lemma (Lemma 4.2) gives the following bound.

Algorithm 1 Proximal-Point Coordinate Descent Scheme (PiCS)

1: **Input:** $\mathbf{x}_0 \in \mathbb{R}^d$, number of communication rounds T, proximal coefficients $\lambda_t > 0$
2: **for** $t = 0, 1, \ldots, T-1$ **do**
3: Server broadcasts $\mathbf{x}_t$ and λ_t to all n clients
4: **For each block** $i \in \{1, \ldots, n\}$ **in parallel:**
5: $\mathbf{x}_{t+1}^{(i)} \approx \arg\min_{\mathbf{x}^{(i)} \in \mathbb{R}^{d_i}} \{ f(\mathbf{x}_t + \mathbf{U}_i(\mathbf{x}^{(i)} - \mathbf{x}_t^{(i)})) + \frac{\lambda_t}{2} \|\mathbf{x}^{(i)} - \mathbf{x}_t^{(i)}\|^2 \}$ satisfying (6) for $\mathbf{x}_t$ and λ_t.
6: Server sets $\mathbf{x}_{t+1} \leftarrow (\mathbf{x}_{t+1}^{(1)}, \ldots, \mathbf{x}_{t+1}^{(n)})$
7: **Output:** $\mathbf{x}_T$

Theorem 4.3. *Assume coordinate-wise convexity (Assumption 4.1), coordinate-wise smoothness (Assumption 7.1), and global cross-coupling with constant $M > 0$ (Assumption 3.1). Run the PiCS iteration with coefficients λ_t, requiring the bound from the decrease lemma (Lemma 4.2) and local solves meeting* (6). *Then for any $T \geq 1$, $\frac{1}{T}\sum_{t=1}^{T} \|\nabla f(\mathbf{x}_t)\|^2 \leq \frac{20\bar{M}(f(\mathbf{x}_0) - f^\star)}{T}$. In particular, $T \geq \frac{20\bar{M}(f(\mathbf{x}_0) - f^\star)}{\varepsilon^2}$ communication rounds suffice to obtain an ε-stationary point, i.e., a point $\hat{\mathbf{x}}$ satisfying $\|\widetilde{\nabla} f(\hat{\mathbf{x}})\| \leq \varepsilon$.*

4.2 Minimizing function value for convex functions

We assume $D_0 := \sup\{\|\mathbf{x} - \mathbf{x}^\star\| : f(\mathbf{x}) \leq f(\mathbf{x}_0)\} < +\infty$. When f is convex, we measure progress by the function gap and aim to find $\hat{\mathbf{x}}$ such that $f(\hat{\mathbf{x}}) - f^\star \leq \varepsilon$.

Assumption 4.4. The function f is convex.

Theorem 4.5. *Assume coordinate-wise convexity (Assumption 4.1), global cross-coupling with constant M (Assumption 3.1), and convexity of f (Assumption 4.4). Run the PiCS iteration with coefficients λ_t, requiring the bound from the decrease lemma (Lemma 4.2) and local solves meeting* (6). *Assume $M > 0$ and $D_0 < \infty$. Then for all $T \geq 1$, $f(\mathbf{x}_T) - f^\star \leq \frac{40\bar{M} D_0^2}{T}$. To obtain $f(\mathbf{x}_T) - f^\star \leq \varepsilon$, it suffices to run PiCS for $T \geq \frac{40\bar{M} D_0^2}{\varepsilon}$ communication rounds.*

4.3 PL condition

Under the Polyak–Łojasiewicz condition, we continue to target the function-gap criterion $f(\hat{\mathbf{x}}) - f^\star \leq \varepsilon$ and achieve linear convergence.

Assumption 4.6. The function f satisfies the Polyak-Łojasiewicz (PL) condition with constant $\mu > 0$, i.e. if for all $\mathbf{x} \in \mathbb{R}^d$, $\frac{1}{2}\|\nabla f(\mathbf{x})\|^2 \geq \mu\,(f(\mathbf{x}) - f^\star)$.

Strongly convex functions are an important subclass of PL functions.

Theorem 4.7. *Assume coordinate-wise convexity (Assumption 4.1), coordinate-wise smoothness (Assumption 7.1), global cross-coupling with constant $M > 0$ (Assumption 3.1), and the PL condition (Assumption 4.6). Run the PiCS iteration with coefficients λ_t, requiring the bound from the decrease lemma (Lemma 4.2) and local solves meeting* (6). *Then the iterates converge linearly in communication rounds: $f(\mathbf{x}_T) - f^\star \leq \left(\frac{10\bar{M}}{\mu + 10\bar{M}}\right)^T (f(\mathbf{x}_0) - f^\star)$. Thus, to obtain $f(\mathbf{x}_T) - f^\star \leq \varepsilon$, it suffices to run PiCS for $T \geq \left(1 + \frac{10\bar{M}}{\mu}\right) \log \frac{f(\mathbf{x}_0) - f^\star}{\varepsilon}$ communication rounds.*

Discussion 4.8. The key observation here is that all communication complexities *are proportional to M*. It means that as the function gets closer to a separable function, PiCS becomes more communication-efficient and can minimize any function in a constant number of communications when $M = 0$.

5 Accelerated PiCS

For convex objectives, we can also develop a accelerated version of our PiCS similar to Nesterov (2013). The key inequality is Lemma 5.1. This is the inequality required by Nesterov's scheme;

once established, the accelerated-rate proof follows a similar argument. Each iteration approximately solves the block subproblems at the momentum point $\mathbf{y}_t$ to obtain $\mathbf{x}_{t+1}$, then uses $\nabla f(\mathbf{x}_{t+1})$ to update $\mathbf{v}_t$. The weights satisfy $A_t = \sum_{j=1}^{t} a_j$, with $A_0 = 0$. We assume $R_0 := \|\mathbf{x}_0 - \mathbf{x}^\star\| < +\infty$. With $\lambda_t > 0$ and $\lambda'_t := 20\lambda_t$ we propse Algorithm 2.

Algorithm 2 Accelerated PiCS

Input: initial iterate $\mathbf{x}_0 \in \mathbb{R}^d$, number of iterations T, coefficients $\lambda_t > 0$; $A_0 \leftarrow 0$, $\mathbf{v}_0 \leftarrow \mathbf{x}_0$
for $t = 0, 1, \dots, T-1$ **do**
  set $\lambda'_t \leftarrow 20\lambda_t$; choose $a_{t+1} > 0$ with $\frac{1}{2}\lambda'_t a_{t+1}^2 = A_t + a_{t+1}$; set $A_{t+1} \leftarrow A_t + a_{t+1}$
  $\mathbf{y}_t \leftarrow \frac{A_t \mathbf{x}_t + a_{t+1}\mathbf{v}_t}{A_{t+1}}$
  For each client $i \in \{1, \dots, n\}$ in parallel, compute
    $\mathbf{x}_{t+1}^{(i)} \approx \arg\min_{\mathbf{z} \in \mathbb{R}^{d_i}} \left\{ f\left(\mathbf{y}_t + \mathbf{U}_i(\mathbf{z} - \mathbf{y}_t^{(i)})\right) + \frac{\lambda_t}{2}\|\mathbf{z} - \mathbf{y}_t^{(i)}\|^2 \right\}$ satisfying (6) for $\mathbf{y}_t$ and λ_t.
  Server sets $\mathbf{x}_{t+1} \leftarrow (\mathbf{x}_{t+1}^{(1)}, \dots, \mathbf{x}_{t+1}^{(n)})$.
  collect $\nabla_i f(\mathbf{x}_{t+1})$ from each client i and set $\mathbf{v}_{t+1} \leftarrow \mathbf{v}_t - a_{t+1}\nabla f(\mathbf{x}_{t+1})$
Output: $\mathbf{x}_T$

Lemma 5.1. *Assume cross-coupling with constant M (Assumption 3.1). Let $\mathbf{x}_+ := \mathbf{T}(\bar{\mathbf{x}})$ be one PiCS step from any $\bar{\mathbf{x}} \in \mathbb{R}^d$ with proximal coefficient $\lambda > 0$, in which every block subproblem* (5) *is solved to the inexactness criterion* (6)*. Then, for $\lambda \geq 4M$,*

$$\langle \nabla f(\mathbf{x}_+), \bar{\mathbf{x}} - \mathbf{x}_+ \rangle \;\geq\; \frac{\lambda}{4}\,\|\bar{\mathbf{x}} - \mathbf{x}_+\|^2 \;\geq\; \frac{1}{20\lambda}\,\|\nabla f(\mathbf{x}_+)\|^2 . \tag{8}$$

Similar to PiCS, if M is know ,we choose $\lambda_t = 4M$, otherwise, linesearch Accept a trial when $\langle \nabla f(\mathbf{x}_{t+1}), \mathbf{y}_t - \mathbf{x}_{t+1} \rangle \geq \frac{1}{20\lambda_t}\|\nabla f(\mathbf{x}_{t+1})\|^2$. Use the line search of Section 4 with threshold $\bar{\lambda} := 4M$ (Lemma 5.1), giving $\lambda_t \leq \bar{M} := 2\bar{\lambda} = 8M$. Similar to the PiCS, the total number of trials in all rounds changes the communication bound only by a constant factor and an additive $\mathcal{O}(\log(\bar{\lambda}/\hat{\lambda}_0))$ initialization cost, which we omit for the sake of simplicity.

Theorem 5.2. *Assume convexity (Assumption 4.4), coordinate-wise smoothness (Assumption 7.1), and global cross-coupling with constant $M > 0$ (Assumption 3.1). Run the accelerated scheme (Algorithm 2) with coefficients $0 < \lambda_t \leq \bar{M}$ satisfying* (8) *and local solves meeting the inexactness criterion* (6)*. Then for every $T \geq 1$, $f(\mathbf{x}_T) - f^\star \leq \frac{R_0^2}{2A_T} \leq \frac{20\bar{M}R_0^2}{T^2}$. For $\varepsilon > 0$, $T \geq \sqrt{\frac{20\bar{M}R_0^2}{\varepsilon}}$ suffices to obtain $f(\mathbf{x}_T) - f^\star \leq \varepsilon$.*

For convex objectives satisfying the PL condition, standard restarting yields an accelerated linear rate; we omit the details for simplicity.

6 Communication lower bound

We consider two clients with coordinates $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$, initialized without loss of generality at $(\mathbf{x}_0, \mathbf{y}_0) = (\mathbf{0}, \mathbf{0})$. We construct quadratic objectives in $\mathcal{F}_{2p}(M, R)$ (Definition F.1), with cross-coupling at most M and a minimizer within distance R of the origin. We study the linear-span procedures described in Definition F.2 and Assumptions F.3 to F.5. Each client may take arbitrarily many local first-order steps between communication rounds, so the lower bound holds regardless of the number of local steps performed.

Theorem 6.1. *Fix $M, R > 0$ and $T \in \mathbb{N}$, and let $p \geq 2T + 1$. Then there exists a convex quadratic $F \in \mathcal{F}_{2p}(M, R)$ such that every two-client procedure satisfying Assumptions F.3 to F.5 for T communication rounds satisfies $F(\mathbf{x}_T, \mathbf{y}_T) - F^\star \;\geq\; \frac{3MR^2}{16(T+1)^2}$.*

Hence, in sufficiently large dimensions as specified above, any method satisfying Assumptions F.3 to F.5 requires $\Omega(R\sqrt{M/\varepsilon})$ communication rounds in the worst case over $\mathcal{F}_d(M, R)$, regardless of the number of local oracle calls per communication round.

Remark 6.2. For $\bar{M} = \mathcal{O}(M)$ and $R_0 \leq R$, Algorithm 2 reaches ε-accuracy in $\mathcal{O}(R\sqrt{M/\varepsilon})$ communication rounds. This matches the lower bound in Theorem 6.1 up to constants and ***establishes communication optimality*** over $\mathcal{F}_d(M, R)$ (Definition F.1) under the considered setting and assumptions. The extension to more clients follows the same construction and is omitted for the sake of space.

7 Local Solvers and Complexity Analysis

Both PiCS and A-PiCS use local solvers to find approximate solutions. The communication guarantees in Theorems 4.3, 4.5 and 4.7 hold for any local solver satisfying the inexactness condition. In this section, we consider two iterative first-order local solvers for clients: gradient descent (GD) and accelerated gradient descent (AGD), with the latter given in Algorithm 4 (see Section G.2 for details).

Assumption 7.1. For each block $i \in \{1, \ldots, n\}$, there exists a constant $L_i \geq 0$ such that $\|\nabla_i f(\mathbf{x} + \mathbf{U}_i\mathbf{h}) - \nabla_i f(\mathbf{x})\| \leq L_i \|\mathbf{h}\|$ for all $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{h} \in \mathbb{R}^{d_i}$.

We assume that f is coordinate-wise L_i-smooth and coordinate-wise convex (Assumption 4.1). Each local objective $g_{i,\mathbf{x}_t}$ is therefore $(L_i + \lambda_t)$-smooth and λ_t-strongly convex. Each client i runs gradient descent on $g_{i,\mathbf{x}_t}$, starting from $\mathbf{x}_{t,0}^{(i)} = \mathbf{x}_t^{(i)}$ with step size $\frac{1}{L_i+\lambda_t}$. Its local updates are $\mathbf{x}_{t,k+1}^{(i)} = \mathbf{x}_{t,k}^{(i)} - \frac{1}{L_i+\lambda_t}\nabla g_{i,\mathbf{x}_t}(\mathbf{x}_{t,k}^{(i)})$. The client stops as soon as the conditions (6) (PiCS) or (8) (A-PiCS) are satisfied. Full GD pseudocode is given in Algorithm 3. At iteration t, GD requires $\mathcal{O}(\frac{L_i+\lambda_t}{\lambda_t})$ local steps (Lemma G.2), while AGD requires $\mathcal{O}(\sqrt{\frac{L_i+\lambda_t}{\lambda_t}})$ local steps (Lemma G.4). To control these local complexities, we need a lower bound on λ_t. We can ensure this with a simple modification of the line search: after acceptance, start the next trial at $\hat{\lambda}_{t+1} = \max\{\hat{\lambda}_0, \lambda_t/2\}$. This ensures $\lambda_t \geq \hat{\lambda}_0 > 0$.

8 Comparison to Baseline Methods

All methods use the distributed setting of Section 2 and complexity measures (2). Table 1 assumes f is convex (Assumption 4.4), L-smooth, coordinate-wise smooth with constants $\{L_i\}_{i=1}^n$ (Assumption 7.1), cross-coupled with constant M (Assumption 3.1), and ω-partially separable with convex components (Definition H.1), targeting $f(\hat{\mathbf{x}}) - f^\star \leq \varepsilon$. PCDM uses τ-nice sampling with $\tau = n$. Since few baselines target distributed CD, we also consider GD, GCD, and RCDM through naive use in our distributed setting, although they were not designed for it. To compare PiCS with

Table 1: Complexity comparison under the assumptions of Section 7, using $R(\varepsilon)$ and $W(\varepsilon)$ from (2). All entries are in $\mathcal{O}$ notation. The third column gives an upper bound on $\max_{i,t} K_{t,i}$ (we assume $M > 0$ for simplicity of the presentation). We assume M is known and we choose $\lambda_t = 2M$ for PiCS and $\lambda_t = 4M$ for A-PiCS.[2]

Method	Comm. rounds $R(\varepsilon)$	Local steps per comm.	$W(\varepsilon)$	Reference
GD	$\frac{D_0^2 L}{\varepsilon}$	1	$\frac{D_0^2 L}{\varepsilon}$	—
GCD	$\frac{D_0^2 n L_{\max}}{\varepsilon}$	1	$\frac{D_0^2 n L_{\max}}{\varepsilon}$	(Nutini et al., 2015)
RCDM	$\frac{D_0^2 n \bar{L}}{\varepsilon}$	1	$\frac{D_0^2 n \bar{L}}{\varepsilon}$	(Nesterov, 2012)
PCDM	$\frac{\omega\, D_L^2}{\varepsilon}$	1	$\frac{\omega\, D_L^2}{\varepsilon}$	(Richtárik & Takáč, 2016b)
DBCD	$\frac{\omega\, D_L^2}{\varepsilon}$	1	$\frac{\omega\, D_L^2}{\varepsilon}$	(Mareček et al., 2015)
Hydra	$\frac{\sigma\, D_{\mathrm{diag}(\mathbf{S})}^2}{\varepsilon}$	1	$\frac{\sigma\, D_{\mathrm{diag}(\mathbf{S})}^2}{\varepsilon}$	(Richtárik & Takáč, 2016a)
PiCS-GD	$\frac{D_0^2 M}{\varepsilon}$	$\frac{L_{\max}}{M} + 1$	$\frac{D_0^2 (L_{\max}+M)}{\varepsilon}$	**(ours)**
A-PiCS-AGD	$R_0\sqrt{\frac{M}{\varepsilon}}$	$\sqrt{\frac{L_{\max}}{M}} + 1$	$R_0\sqrt{\frac{L_{\max}+M}{\varepsilon}}$	**(ours)**

[2]Here $\omega \in [1, n]$; D_0 and R_0 are defined in Sections 4.2 and 5. For $\mathbf{x} \in \mathbb{R}^d$ and $\mathbf{S} \in \mathbb{R}^{d\times d}$ with $\mathbf{S} \succeq 0$, we write $\|\mathbf{x}\|_{\mathbf{S}} := \sqrt{\langle \mathbf{S}\mathbf{x}, \mathbf{x}\rangle}$, $D_L^2 := \sum_i L_i \|\mathbf{x}_0^{(i)} - (\mathbf{x}^\star)^{(i)}\|^2$ and $D_{\mathrm{diag}(\mathbf{S})}^2 := \|\mathbf{x}_0 - \mathbf{x}^\star\|_{\mathbf{D_S}}^2$, where $\mathbf{D_S} = \mathrm{diag}(S_{11}, \ldots, S_{dd})$. DBCD and Hydra use full sampling ($\tau = n$). For a square matrix $\mathbf{S}$, $\mathrm{diag}(\mathbf{S})$ denotes its diagonal part.

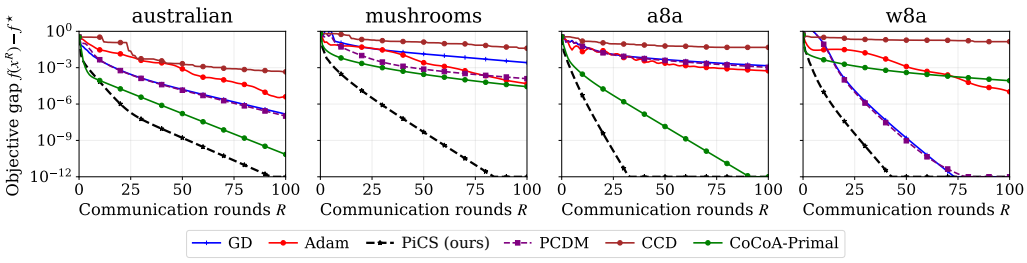


Figure 1: **Binary classification on LIBSVM benchmarks.** Objective gap versus communication rounds on `australian`, `mushrooms`, `a8a`, and `w8a`, using two workers.

the baselines, we first relate M to their smoothness and structural parameters. Expected Separable Overapproximation (ESO) analyses often express bounds through the partial-separability degree ω (Richtárik & Takáč, 2016b; Qu & Richtárik, 2016; Tappenden et al., 2018). Lemma H.2 relates it to cross-coupling: $\boxed{M \le \omega L_{\max}}$, where $L_{\max} := \max_i L_i$. Hydra (Richtárik & Takáč, 2016a) uses $\mathbf{S}$-smoothness, the normalized spectral parameter σ, and $s_{\max} := \max_i S_{ii}$ (Definition H.3); for convex f, $\boxed{M \le \sigma s_{\max}}$ (Lemma H.4). The weighted distances satisfy $D^2_{\mathrm{diag}(\mathbf{S})} \le s_{\max} D_0^2$ for Hydra and $D_L^2 \le L_{\max} D_0^2$ for PCDM, allowing comparison under a common Euclidean distance bound. Our analysis can also be extended to weighted norms, as used by PCDM and Hydra, but we omit this extension for simplicity. For convex, twice continuously differentiable, L-smooth f satisfying Assumption 7.1, Lemma H.5 gives $\boxed{M \le n\bar{L}}$, where $\bar{L} := \frac{1}{n}\sum_{i=1}^n L_i$. Finally, Lemma H.6 gives $\boxed{M \le \left(\sum_{i=1}^n \ell_i^2\right)^{1/2}}$, where ℓ_i is a Lipschitz constant of $\nabla_i f$ over all blocks, while L_i concerns only block i. We compare FedBCD's (Liu et al., 2022) nonconvex rates here (Table 1 covers convex function values), using full gradients on coordinate-wise convex objectives with the above Lipschitz bounds, $M > 0$, and $\Delta_f := f(\mathbf{x}_0) - f^\star < \infty$. Let T count communication rounds, K local steps per client per round, and $\mathbf{z}_r \in \mathbb{R}^d$ stack the blocks after local step r, with $\mathbf{z}_0 = \mathbf{x}_0$. The preprint bound (Liu et al., 2020, Theorem 2) is $\frac{1}{KT}\sum_{r=0}^{KT-1} \|\nabla f(\mathbf{z}_r)\|^2 \le \mathcal{O}\big(\frac{(\sum_i \ell_i^2)^{1/2}\Delta_f}{T}\big)$, whereas Theorem 4.3 gives PiCS $\frac{1}{T}\sum_{t=1}^T \|\nabla f(\mathbf{x}_t)\|^2 \le \frac{40M\Delta_f}{T}$ at communication iterates, for the fixed choice $\lambda_t = 2M$. These bounds and $M \le L$ (Lemma C.1) give PiCS no worse smoothness dependence, up to universal constants, than FedBCD and the table baselines under their respective assumptions and a common Euclidean distance bound for convex rates. Under the stated assumptions, PiCS matches or improves the worst-case communication bounds in Table 1, up to constants. For nearly block-separable functions, M can be arbitrarily small while the baseline coefficients need not be. PiCS and A-PiCS denote our basic and accelerated schemes (Algorithms 1 and 2); the suffixes GD and AGD identify the local solver (Section 7).

9 Experiments

We compare the performance of PiCS with baseline methods on LIBSVM classification. Figure 1 illustrates the performance of logistic regression with ℓ_2 regularization on four LIBSVM benchmarks (Chang & Lin, 2011) using two workers and compare against GD, Adam, PCDM, CCD, and primal CoCoA. Additional experiments on line-search performance, sparse problems, neural-network training, and other settings appear in Section I.

10 Conclusion

We introduced cross-coupling M and developed PiCS with inexact local solvers, adaptive line search, and an accelerated variant matching our communication lower bound up to constants. Under the stated comparison assumptions, our communication bounds match or improve those of the baselines up to constants, with potentially substantial gains when M is small. A main limitation is the $\sqrt{n}$ factor in the general bound relating M to the off-diagonal Hessian norm; avoiding this factor currently requires additional structural assumptions. Reducing this gap and obtaining simpler, sharper estimates of M under weaker assumptions remain open questions.

AI USE STATEMENT

We used generative AI tools to assist with implementing experiments, generating figures, searching the literature for related work and results, polishing the text, and correcting spelling and grammar. We also used AI to help identify potential flaws in the proofs, alongside human review of the proofs. We did not use these tools to draft complete sections or other substantial portions of the paper, develop the theory, design the experiments, or interpret experimental results.

We checked cited papers to verify references and claims, reviewed the edited text to ensure that its meaning was preserved, and examined figures and experimental results in special cases where the expected behavior was theoretically clear. The authors take responsibility for the entire submission, including all AI-assisted material.

REFERENCES

Amir Beck and Luba Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013. doi: 10.1137/120887679.

Joseph K. Bradley, Aapo Kyrola, Danny Bickson, and Carlos Guestrin. Parallel coordinate descent for L1-regularized loss minimization. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 321–328, 2011.

Timothy Castiglia, Shiqiang Wang, and Stacy Patterson. Flexible vertical federated learning with heterogeneous parties. *IEEE Transactions on Neural Networks and Learning Systems*, 35(12): 17878–17892, 2024. doi: 10.1109/TNNLS.2023.3309701.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011.

Tianyi Chen, Xiao Jin, Yuejiao Sun, and Wotao Yin. VAFL: A method of vertical asynchronous federated learning. In *ICML Workshop on Federated Learning for User Privacy and Data Confidentiality*, 2020. URL `https://arxiv.org/abs/2007.06081`.

Amir Daneshmand, Francisco Facchinei, Vyacheslav Kungurtsev, and Gesualdo Scutari. Hybrid random/deterministic parallel algorithms for convex and nonconvex big data optimization. *IEEE Transactions on Signal Processing*, 63:3914–3929, 2015. doi: 10.1109/TSP.2015.2436357.

Francisco Facchinei, Gesualdo Scutari, and Simone Sagratella. Parallel selective algorithms for nonconvex big data optimization. *IEEE Transactions on Signal Processing*, 63(7):1874–1889, 2015. doi: 10.1109/TSP.2015.2399858.

Olivier Fercoq and Peter Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

Gene H Golub and Charles F Van Loan. *Matrix Computations*. JHU press, 4th edition, 2013.

Luigi Grippo and Marco Sciandrone. On the convergence of the block nonlinear Gauss–Seidel method under convex constraints. *Operations Research Letters*, 26(3):127–136, 2000. doi: 10.1016/S0167-6377(99)00074-7.

Clifford Hildreth. A quadratic programming procedure. *Naval Research Logistics Quarterly*, 4(1): 79–85, 1957.

Cho-Jui Hsieh, Hsiang-Fu Yu, and Inderjit S Dhillon. PASSCoDe: Parallel asynchronous stochastic dual co-ordinate descent. In *International Conference on Machine Learning*, pp. 2370–2379. PMLR, 2015.

Xiaowen Jiang, Anton Rodomanov, and Sebastian U. Stich. Non-convex federated optimization under cost-aware client selection. *arXiv preprint arXiv:2512.05327*, 2025.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2): 1–210, 2021. doi: 10.1561/2200000083.

Guanghui Lan, Yuyuan Ouyang, and Zhe Zhang. Optimal and parameter-free gradient minimization methods for convex and nonconvex optimization. *arXiv preprint arXiv:2310.12139*, 2023. doi: 10.48550/arXiv.2310.12139.

Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated proximal coordinate gradient method. *SIAM Journal on Optimization*, 25(4):2244–2273, 2015.

Ji Liu and Stephen J Wright. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, 25(1):351–376, 2015.

Yang Liu, Yan Kang, Xinwei Zhang, Liping Li, Yong Cheng, Tianjian Chen, Mingyi Hong, and Qiang Yang. A communication efficient collaborative learning framework for distributed features. *arXiv preprint arXiv:1912.11187v6*, 2020. URL https://arxiv.org/abs/1912.11187v6.

Yang Liu, Xinwei Zhang, Yan Kang, Liping Li, Tianjian Chen, Mingyi Hong, and Qiang Yang. FedBCD: A communication-efficient collaborative learning framework for distributed features. *IEEE Transactions on Signal Processing*, 70:4277–4290, 2022. doi: 10.1109/TSP.2022.3198176.

Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3615–3634, 2024. doi: 10.1109/TKDE.2024.3352628.

Zhi-Quan Luo and Paul Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992. doi: 10.1007/BF00939948.

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.

Dhruv Mahajan, S. Sathiya Keerthi, and S. Sundararajan. A distributed block coordinate descent method for training l1 regularized linear classifiers. *Journal of Machine Learning Research*, 18 (91):1–35, 2017. URL https://jmlr.org/papers/v18/14-484.html.

Jakub Mareček, Peter Richtárik, and Martin Takáč. Distributed block coordinate descent for minimizing partially separable functions. In *Numerical Analysis and Optimization*, volume 134 of *Springer Proceedings in Mathematics & Statistics*, pp. 261–288. Springer, 2015. doi: 10.1007/978-3-319-17689-5_11.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282. PMLR, 2017.

Mojmír Mutný and Peter Richtárik. Parallel stochastic Newton method. *arXiv preprint arXiv:1705.02005*, 2017.

Ion Necoara and Dragos Clipici. Efficient parallel coordinate descent algorithm for convex optimization problems with separable constraints: Application to distributed mpc. *Journal of Process Control*, 23(3):243–253, 2013. doi: 10.1016/j.jprocont.2012.12.012.

Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012. doi: 10.1137/100802001.

Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013. doi: 10.1007/s10107-012-0629-5.

Yurii Nesterov. *Lectures on Convex Optimization*, volume 137 of *Springer Optimization and Its Applications*. Springer International Publishing, 2nd edition, 2018. doi: 10.1007/978-3-319-91578-4.

Yurii Nesterov and Sebastian U Stich. Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, 27(1):110–123, 2017.

Feng Niu, Benjamin Recht, Christopher Re, and Stephen Wright. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, volume 24, 2011.

Julie Nutini, Mark Schmidt, Issam Laradji, Michael Friedlander, and Hoyt Koepke. Coordinate descent converges faster with the Gauss-Southwell rule than random selection. In *International Conference on Machine Learning*, pp. 1632–1641. PMLR, 2015.

Zhimin Peng, Yangyang Xu, Ming Yan, and Wotao Yin. ARock: An algorithmic framework for asynchronous parallel coordinate updates. *SIAM Journal on Scientific Computing*, 38(5):A2851–A2879, 2016. doi: 10.1137/15M1024950.

Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling II: Expected separable overapproximation. *Optimization Methods and Software*, 31(5):858–884, 2016.

Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1):1–38, 2014.

Peter Richtárik and Martin Takáč. Distributed coordinate descent method for learning with big data. *Journal of Machine Learning Research*, 17(75):1–25, 2016a.

Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1):433–484, 2016b.

Mher Safaryan, Filip Hanzely, and Peter Richtárik. Smoothness matrices beat smoothness constants: Better communication compression techniques for distributed optimization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 25688–25702, 2021.

Chad Scherrer, Mahantesh Halappanavar, Ambuj Tewari, and David Haglin. Scaling up coordinate descent algorithms for large ℓ_1 regularization problems. In *Proceedings of the 29th International Conference on Machine Learning*, pp. 1407–1414. Omnipress, 2012.

Gesualdo Scutari, Francisco Facchinei, Peiran Song, Daniel P. Palomar, and Jong-Shi Pang. Decomposition by partial linearization: Parallel optimization of multi-agent systems. *IEEE Transactions on Signal Processing*, 62(3):641–656, 2014. doi: 10.1109/TSP.2013.2293126.

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

Virginia Smith, Simone Forte, Chenxin Ma, Martin Takáč, Michael I Jordan, and Martin Jaggi. Cocoa: A general framework for communication-efficient distributed optimization. *Journal of Machine Learning Research*, 18(230):1–49, 2018.

Sebastian U. Stich. Local SGD converges fast and communicates little. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=S1g2JnRcFX`.

Ruoyu Sun and Mingyi Hong. Improved iteration complexity bounds of cyclic block coordinate descent for convex problems. In *Advances in Neural Information Processing Systems*, volume 28, pp. 1306–1314, 2015.

Rachael Tappenden, Martin Takáč, and Peter Richtárik. On the complexity of parallel coordinate descent. *Optimization Methods and Software*, 33(2):372–395, 2018.

Ilya Trofimov and Alexander Genkin. Distributed coordinate descent for generalized linear models with regularization. *arXiv preprint arXiv:1611.02101*, 2016. doi: 10.48550/arXiv.1611.02101.

Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S. Rellermeyer. A survey on distributed machine learning. *ACM Computing Surveys*, 53(2):1–33, 2020. doi: 10.1145/3377454.

Stephen J Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):12:1–12:19, 2019. doi: 10.1145/3298981.

APPENDIX TABLE OF CONTENTS

A Related Work **15**

B Communication model and examples **15**

C Proofs for Section 3 **16**

C.1 Separable upper bound 17

C.2 Relationship between M and L 17

C.3 Proof of Theorem 3.2 18

C.4 Proof of Theorem 3.3 18

C.5 Proof of Theorem 3.4 18

C.6 Nonnegative matrix norm comparison 18

C.7 Proof of Lemma 3.5 19

C.8 Proof of Section 3 19

D Proofs for Section 4 **19**

D.1 Gradient bound for PiCS 19

D.2 Proof of Lemma 4.2 20

D.3 Proof of Theorem 4.3 20

D.4 Proof of Theorem 4.5 21

D.5 Proof of Theorem 4.7 21

E Proofs for Section 5 **21**

E.1 Proof of Lemma 5.1 21

E.2 Proof of Theorem 5.2 22

F Proofs for Section 6 **22**

F.1 Framework and hard instance 22

F.2 Krylov-space bounds 24

F.3 Proof of Theorem 6.1 26

G Proofs for Section 7 **26**

G.1 Gradient descent (GD) 26

G.2 Accelerated solver (ARM) 28

H Proofs for Section 7 **30**

H.1 Cross-coupling and partial separability 30

H.2 Cross-coupling under matrix smoothness 30

H.3 Sum of block smoothness 31

H.4 Root-sum-of-squares coupling bound 31

H.5 Baseline complexity comparisons 31

I Additional experiments and reproducibility **34**

A Related Work

Coordinate descent foundations and acceleration. Randomized CD was analyzed by Nesterov (2012); Nesterov & Stich (2017). These methods trace back to classical linear solvers (Golub & Van Loan, 2013): for quadratic objectives, Gauss–Seidel corresponds to sequential CD, and Jacobi to parallel updates. Our approach refines the Jacobi paradigm by correcting for interactions between coordinates. Early precursors include (Hildreth, 1957) for quadratic programming and (Luo & Tseng, 1993) for error bound analysis. Nutini et al. (2015) showed that greedy (Gauss–Southwell) selection can outperform random selection. Shalev-Shwartz & Zhang (2013) analyzed SDCA with sharp rates for regularized loss minimization; Richtárik & Takáč (2014) studied randomized CD for composite objectives. Under our communication model, these sequential methods require $\mathcal{O}(n)$ communication rounds per epoch. The APPROX framework (Fercoq & Richtárik, 2015) and APCG (Lin et al., 2015) achieve accelerated $\mathcal{O}(1/k^2)$ rates for convex objectives by combining Nesterov acceleration with coordinate updates.

Parallel and distributed variants. Tappenden et al. (2018) sharpen PCDM's rates and establish monotonicity in expectation. Hydra (Richtárik & Takáč, 2016a) extends to the distributed setting with two-level parallelism. The Parallel Stochastic Newton method (Mutný & Richtárik, 2017) achieves linear convergence under strong convexity using a matrix smoothness model. Asynchronous methods such as HOGWILD! (Niu et al., 2011) and AsySCD (Liu & Wright, 2015) allow updates without synchronization.

Parallel proximal Jacobi methods. Our method is closely related to parallel proximal Jacobi methods. Like PiCS, these methods compute coordinate updates in parallel by solving local proximal problems while keeping the other coordinates fixed. Parallel proximal Jacobi methods solve local proximal problems simultaneously, keeping the other coordinates fixed. Scutari et al. (2014) establish convergence for exact and inexact Jacobi schemes that preserve the objective's convex structure in each coordinate. Facchinei et al. (2015) allow selective parallel updates and nonsmooth separable terms. Daneshmand et al. (2015) combine random sampling with greedy coordinate selection while allowing inexact local solves. These methods use suitable regularization and step sizes to ensure convergence for convex and nonconvex problems.

B Communication model and examples

In this section, we describe in detail how our distributed system operates, including how clients and servers communicate and the constraints they face on computing and communication. We further illustrate these ideas by explaining how the setup can be used to solve a GLM.

Local first-order oracle. For an anchor point $\bar{\mathbf{x}} \in \mathbb{R}^d$, client i has access to the anchored block oracle

$$\mathrm{O}_i^{\bar{\mathbf{x}}}(\mathbf{z}) \coloneqq \Big(f\big(\bar{\mathbf{x}} + \mathbf{U}_i(\mathbf{z} - \bar{\mathbf{x}}^{(i)})\big) - f(\bar{\mathbf{x}}),\ \nabla_i f\big(\bar{\mathbf{x}} + \mathbf{U}_i(\mathbf{z} - \bar{\mathbf{x}}^{(i)})\big)\Big), \qquad \mathbf{z} \in \mathbb{R}^{d_i}. \tag{9}$$

The server has no direct access to f or to any $\mathrm{O}_i^{\bar{\mathbf{x}}}$; only client i can query its own oracle. We now formalize the information available to the server and clients within each communication round.

Example: GLMs. We use the sample-separable GLM class

$$f(\mathbf{x}) = \Phi(\mathbf{A}\mathbf{x}) = \sum_{j=1}^{m} \phi\big(\langle \mathbf{a}_j, \mathbf{x}\rangle - b_j\big) \tag{10}$$

where $\mathbf{A} \in \mathbb{R}^{m\times d}$ has rows $\mathbf{a}_1^\top, \dots, \mathbf{a}_m^\top$, $\mathbf{b} = (b_1, \dots, b_m)^\top \in \mathbb{R}^m$, and $\phi\colon \mathbb{R} \to \mathbb{R}$ is differentiable. The sum is over data points, and the columns are distributed by feature blocks, $\mathbf{A} = [\mathbf{A}_1, \dots, \mathbf{A}_n]$ with $\mathbf{A}_i \in \mathbb{R}^{m\times d_i}$. If the server provides $\mathbf{A}\bar{\mathbf{x}}$ and client i retains $\bar{\mathbf{x}}^{(i)}$, then client i can evaluate the oracle in (9) at any query $\mathbf{z} \in \mathbb{R}^{d_i}$ via

$$\begin{aligned} f\big(\bar{\mathbf{x}} + \mathbf{U}_i(\mathbf{z} - \bar{\mathbf{x}}^{(i)})\big) - f(\bar{\mathbf{x}}) &= \Phi\big(\mathbf{A}\bar{\mathbf{x}} + \mathbf{A}_i(\mathbf{z} - \bar{\mathbf{x}}^{(i)})\big) - \Phi(\mathbf{A}\bar{\mathbf{x}}), \\ \nabla_i f\big(\bar{\mathbf{x}} + \mathbf{U}_i(\mathbf{z} - \bar{\mathbf{x}}^{(i)})\big) &= \mathbf{A}_i^\top \nabla\Phi\big(\mathbf{A}\bar{\mathbf{x}} + \mathbf{A}_i(\mathbf{z} - \bar{\mathbf{x}}^{(i)})\big). \end{aligned}$$

Thus the query uses only $\mathbf{z}$, the local feature block $\mathbf{A}_i$, the current block $\bar{\mathbf{x}}^{(i)}$, the known loss map Φ, and the broadcast vector $\mathbf{A}\bar{\mathbf{x}}$.

Protocol. We adopt the algorithmic abstraction of Jiang et al. (2025), specialized to our block-coordinate setting. An algorithm proceeds in synchronous communication rounds indexed by $t \geq 0$. At initialization, the server is given a *state* $\mathcal{M}_0$—a finite ordered collection of vectors—encoding any prior information about f (e.g., a warm start). At the start of each communication round t, the server holds the current state $\mathcal{M}_t$, from which the current iterate is obtained through a deterministic *readout map*,

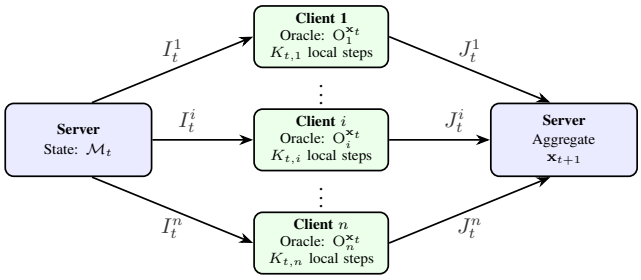


Figure 2: Formal protocol for one communication round. The server sends I_t^i to each client, client i makes $K_{t,i}$ calls to $\mathrm{O}_i^{\mathbf{x}_t}$ and returns J_t^i, and the server forms the next state.

$$\mathbf{x}_t \;\coloneqq\; \mathcal{X}(\mathcal{M}_t) \;\in\; \mathbb{R}^d. \tag{11}$$

The server then constructs from $\mathcal{M}_t$, for each client i, a *transmit set* I_t^i: a finite collection of vectors with the property that, given I_t^i, client i can evaluate the oracle $\mathrm{O}_i^{\mathbf{x}_t}(\mathbf{z})$ at every query point $\mathbf{z} \in \mathbb{R}^{d_i}$. After receiving I_t^i, client i executes a *local procedure* that produces a possibly empty sequence of query points $\mathbf{x}_{t,0}^i, \mathbf{x}_{t,1}^i, \ldots, \mathbf{x}_{t,K_{t,i}-1}^i \in \mathbb{R}^{d_i}$ and accumulates the resulting oracle responses into a *local history*

$$\mathcal{H}_{t,k}^i \;\coloneqq\; \Big(I_t^i,\; \big\{\mathbf{x}_{t,j}^{(i)},\; \mathrm{O}_i^{\mathbf{x}_t}(\mathbf{x}_{t,j}^{(i)})\big\}_{j=0}^{k-1}\Big),\; k = 0, 1, \ldots, K_{t,i}. \tag{12}$$

The local procedure is specified by three (possibly randomized) rules: a *query rule* that selects $\mathbf{x}_{t,k}^i$ as a function of the current history $\mathcal{H}_{t,k}^i$, a *termination rule* that sets the number of local oracle calls $K_{t,i} \geq 0$, and an *output rule* that produces the *response* J_t^i, a finite collection of vectors, from the final history $\mathcal{H}_{t,K_{t,i}}^i$. The server aggregates the responses $\{J_t^i\}_{i=1}^n$ into the next state $\mathcal{M}_{t+1}$, which determines the next iterate $\mathbf{x}_{t+1} = \mathcal{X}(\mathcal{M}_{t+1})$. Figure 2 summarizes one communication round.

Complexity metrics. The two complexity metrics are defined in (2). The formal protocol above specifies the query counts $K_{t,i}$ that are used in the parallel bottleneck complexity.

Example: CCD. Under the communication model above, cyclic coordinate descent (CCD) (Beck & Tetruashvili, 2013) fits naturally into our protocol. Let $i_t = (t \bmod n) + 1$ denote the active block in communication round t. The server state is $\mathcal{M}_t = \{\mathbf{x}_t\}$, with readout $\mathcal{X}(\mathcal{M}_t) = \mathbf{x}_t$. The server sends $I_t^{i_t} = \{\mathbf{x}_t\}$ and $I_t^i = \emptyset$ for every $i \neq i_t$. Given $I_t^{i_t}$, client i_t can evaluate its coordinate-wise oracle $\mathrm{O}_{i_t}^{\mathbf{x}_t}$; a single query at $\mathbf{x}_t^{(i_t)}$ returns the block gradient $\nabla_{i_t} f(\mathbf{x}_t)$. Hence $K_{t,i_t} = 1$ and $K_{t,i} = 0$ for $i \neq i_t$. The active client computes $\mathbf{x}_{t+1}^{(i_t)} = \mathbf{x}_t^{(i_t)} - \frac{1}{L_{i_t}} \nabla_{i_t} f(\mathbf{x}_t)$ and returns $J_t^{i_t} = \{\mathbf{x}_{t+1}^{(i_t)}\}$, while inactive clients return $J_t^i = \emptyset$. The server then forms $\mathbf{x}_{t+1}$ by replacing block i_t of $\mathbf{x}_t$ with $\mathbf{x}_{t+1}^{(i_t)}$ and sets $\mathcal{M}_{t+1} = \{\mathbf{x}_{t+1}\}$.

C Proofs for Section 3

We derive the separable upper bound Ω and relate M to the Hessian.

C.1 SEPARABLE UPPER BOUND

We derive (4) for arbitrary $\mathbf{x}, \mathbf{h} \in \mathbb{R}^d$.

$$
\begin{aligned}
f(\mathbf{x}+\mathbf{h}) - f(\mathbf{x}) &= \int_0^1 \langle \nabla f(\mathbf{x}+t\mathbf{h}), \mathbf{h}\rangle \, dt = \sum_{i=1}^n \int_0^1 \langle \nabla_i f(\mathbf{x}+t\mathbf{h}), \mathbf{h}^{(i)}\rangle \, dt \\
&= \sum_{i=1}^n \int_0^1 \langle \nabla_i f(\mathbf{x}+t\mathbf{U}_i\mathbf{h}^{(i)}), \mathbf{h}^{(i)}\rangle \, dt + \int_0^1 \sum_{i=1}^n \langle \nabla_i f(\mathbf{x}+t\mathbf{h}) - \nabla_i f(\mathbf{x}+t\mathbf{U}_i\mathbf{h}^{(i)}), \mathbf{h}^{(i)}\rangle \, dt \\
&\le \sum_{i=1}^n \big[f(\mathbf{x}+\mathbf{U}_i\mathbf{h}^{(i)}) - f(\mathbf{x})\big] + \|\mathbf{h}\| \int_0^1 \Big(\sum_{i=1}^n \|\nabla_i f(\mathbf{x}+t\mathbf{h}-\mathbf{P}_{-i}(t\mathbf{h})) - \nabla_i f(\mathbf{x}+t\mathbf{h})\|^2\Big)^{1/2} dt \\
&\le \sum_{i=1}^n \big[f(\mathbf{x}+\mathbf{U}_i\mathbf{h}^{(i)}) - f(\mathbf{x})\big] + \frac{M}{2}\|\mathbf{h}\|^2 =: \Omega(\mathbf{x}, \mathbf{x}+\mathbf{h}) - f(\mathbf{x}).
\end{aligned}
$$

Here we used the fundamental theorem of calculus, Cauchy–Schwarz, and $\mathbf{x} + t\mathbf{U}_i\mathbf{h}^{(i)} = \mathbf{x} + t\mathbf{h} - \mathbf{P}_{-i}(t\mathbf{h})$; cross-coupling (Assumption 3.1) is applied at $\mathbf{x} + t\mathbf{h}$ with displacement $-t\mathbf{h}$.

C.2 RELATIONSHIP BETWEEN M AND L

For a convex L-smooth function f, with $L > 0$, cocoercivity (Nesterov, 2018, Theorem 2.1.5) can be equivalently written as follows: for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$,

$$
\left\| \frac{2}{L}\big(\nabla f(\mathbf{u}) - \nabla f(\mathbf{v})\big) - (\mathbf{u}-\mathbf{v}) \right\| \le \|\mathbf{u}-\mathbf{v}\| . \tag{13}
$$

Lemma C.1. *Let f be convex and L-smooth with $L > 0$. Then f satisfies Assumption 3.1, and its smallest cross-coupling constant satisfies*

$$
M \le L.
$$

Proof. Fix $\mathbf{x}, \mathbf{h} \in \mathbb{R}^d$. Then

$$
\begin{aligned}
&\frac{2}{L}\left(\sum_{i=1}^n \|\nabla_i f(\mathbf{x}+\mathbf{P}_{-i}\mathbf{h}) - \nabla_i f(\mathbf{x})\|^2\right)^{\frac{1}{2}} \\
&= \left(\sum_{i=1}^n \left\| \mathbf{U}_i^\top \Big[\frac{2}{L}\big(\nabla f(\mathbf{x}+\mathbf{P}_{-i}\mathbf{h}) - \nabla f(\mathbf{x}+\mathbf{h})\big) + \mathbf{U}_i\mathbf{h}^{(i)}\Big] + \mathbf{U}_i^\top \Big[\frac{2}{L}\big(\nabla f(\mathbf{x}+\mathbf{h}) - \nabla f(\mathbf{x})\big) - \mathbf{h}\Big] \right\|^2\right)^{\frac{1}{2}} \\
&\le \left(\sum_{i=1}^n \left\| \frac{2}{L}\big(\nabla f(\mathbf{x}+\mathbf{P}_{-i}\mathbf{h}) - \nabla f(\mathbf{x}+\mathbf{h})\big) + \mathbf{U}_i\mathbf{h}^{(i)} \right\|^2\right)^{\frac{1}{2}} + \left\| \frac{2}{L}\big(\nabla f(\mathbf{x}+\mathbf{h}) - \nabla f(\mathbf{x})\big) - \mathbf{h} \right\| \\
&\le \left(\sum_{i=1}^n \left\| \mathbf{U}_i\mathbf{h}^{(i)} \right\|^2\right)^{\frac{1}{2}} + \|\mathbf{h}\| = 2\|\mathbf{h}\| .
\end{aligned}
$$

where the first inequality uses the triangle inequality and orthogonality of the blocks, and the last follows from (13). By (3), this gives $M \le L$. □

Lemma C.2. *For any matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ and any block $1 \le i \le n$:*

$$
\mathbf{U}_i^\top \operatorname{off}(\mathbf{A}) = \mathbf{U}_i^\top \mathbf{A}\mathbf{P}_{-i}.
$$

Proof. Since $\mathbf{U}_i^\top\mathbf{U}_j = 0$ for $j \ne i$ and $\mathbf{U}_i^\top\mathbf{U}_i = I_{d_i}$:

$$
\mathbf{U}_i^\top \operatorname{off}(\mathbf{A}) = \mathbf{U}_i^\top \left(\mathbf{A} - \sum_{j=1}^n \mathbf{U}_j\mathbf{U}_j^\top\mathbf{A}\mathbf{U}_j\mathbf{U}_j^\top\right) = \mathbf{U}_i^\top\mathbf{A} - \mathbf{U}_i^\top\mathbf{A}\mathbf{U}_i\,\mathbf{U}_i^\top = \mathbf{U}_i^\top\mathbf{A}\mathbf{P}_{-i} \qquad \square
$$

C.3 Proof of Theorem 3.2

Proof. Fix $\mathbf{x} \in \mathbb{R}^d$ and a unit vector $\mathbf{h} \in \mathbb{R}^d$. Then, for any $t > 0$, we have, according to (3),

$$\sum_{i=1}^{n} \left\| \frac{1}{t} \mathbf{U}_i^\top [\nabla f(\mathbf{x} + t\mathbf{P}_{-i}\mathbf{h}) - \nabla f(\mathbf{x})] \right\|^2 \leq M^2.$$

As $t \to 0$, we obtain

$$M^2 \geq \sum_{i=1}^{n} \left\| \mathbf{U}_i^\top \nabla^2 f(\mathbf{x})\, \mathbf{P}_{-i}\mathbf{h} \right\|^2 \overset{\text{Lemma C.2}}{=} \sum_{i=1}^{n} \left\| \mathbf{U}_i^\top \operatorname{off}\left(\nabla^2 f(\mathbf{x})\right) \mathbf{h} \right\|^2 = \left\| \operatorname{off}\left(\nabla^2 f(\mathbf{x})\right) \mathbf{h} \right\|^2.$$

Since the unit vector $\mathbf{h}$ is arbitrary, we conclude that $\left\|\operatorname{off}(\nabla^2 f(\mathbf{x}))\right\| \leq M$. □

C.4 Proof of Theorem 3.3

Proof. Since $\nabla^2 f \equiv \mathbf{A}$, the proof of the off-Hessian bound in Theorem 3.4 gives, for every $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ with $\mathbf{h} = \mathbf{y} - \mathbf{x}$,

$$\sum_{i=1}^{n} \left\| \nabla_i f(\mathbf{x}) - \nabla_i f(\mathbf{y} + \mathbf{U}_i(\mathbf{x}^{(i)} - \mathbf{y}^{(i)})) \right\|^2 = \sum_{i=1}^{n} \left\| \mathbf{U}_i^\top \operatorname{off}(\mathbf{A})\, \mathbf{h} \right\|^2 = \left\| \operatorname{off}(\mathbf{A})\, \mathbf{h} \right\|^2.$$

Thus $M \leq \|\operatorname{off}(\mathbf{A})\|$. The reverse inequality follows from Theorem 3.2, because $\sup_{\mathbf{x} \in \mathbb{R}^d} \left\|\operatorname{off}(\nabla^2 f(\mathbf{x}))\right\| = \|\operatorname{off}(\mathbf{A})\|$ for a quadratic. Therefore $M = \|\operatorname{off}(\mathbf{A})\|$. The same argument holds for any symmetric $\mathbf{A}$. □

C.5 Proof of Theorem 3.4

Proof. Fix $\mathbf{x}, \mathbf{h} \in \mathbb{R}^d$. For each block i, let

$$\mathbf{J}_i \coloneqq \int_0^1 \nabla^2 f(\mathbf{x} + t\mathbf{P}_{-i}\mathbf{h})\, dt.$$

Applying Lemma C.2 gives

$$\nabla_i f(\mathbf{x} + \mathbf{P}_{-i}\mathbf{h}) - \nabla_i f(\mathbf{x}) = \mathbf{U}_i^\top \mathbf{J}_i \mathbf{P}_{-i}\mathbf{h} = \mathbf{U}_i^\top \operatorname{off}(\mathbf{J}_i)\mathbf{h}.$$

Hence

$$\sum_{i=1}^{n} \|\nabla_i f(\mathbf{x} + \mathbf{P}_{-i}\mathbf{h}) - \nabla_i f(\mathbf{x})\|^2 = \sum_{i=1}^{n} \left\| \mathbf{U}_i^\top \operatorname{off}(\mathbf{J}_i)\mathbf{h} \right\|^2 \leq \|\mathbf{h}\|^2 \sum_{i=1}^{n} \|\operatorname{off}(\mathbf{J}_i)\|^2 \leq n\widetilde{M}^2 \|\mathbf{h}\|^2. \tag{14}$$

The last inequality follows from the assumption on the off-diagonal part of the Hessian and the linearity of $\operatorname{off}(\cdot)$. □

C.6 Nonnegative matrix norm comparison

Lemma C.3. *Let $\mathbf{X}, \mathbf{Y} \in \mathbb{R}_+^{p \times q}$ satisfy $\mathbf{X} \leq \mathbf{Y}$ entry-wise. Then $\|\mathbf{X}\|_2 \leq \|\mathbf{Y}\|_2$.*

Proof. For any $\mathbf{A} \in \mathbb{R}_+^{p \times q}$, the spectral norm can be attained by non-negative unit vectors. By the variational characterization of the spectral norm and the triangle inequality:

$$\begin{aligned} \|\mathbf{A}\|_2 &= \max_{\|\mathbf{u}\|=\|\mathbf{v}\|=1} \langle \mathbf{A}\mathbf{v}, \mathbf{u} \rangle = \max_{\|\mathbf{u}\|=\|\mathbf{v}\|=1} \sum_{i,j} \mathbf{A}_{i,j}\, u_i\, v_j \\ &\leq \max_{\|\mathbf{u}\|=\|\mathbf{v}\|=1} \sum_{i,j} \mathbf{A}_{i,j}\, |u_i|\, |v_j| = \max_{\|\mathbf{u}\|=\|\mathbf{v}\|=1} \langle \mathbf{A}\,|\mathbf{v}|, |\mathbf{u}| \rangle \leq \max_{\substack{\mathbf{u} \geq 0,\, \mathbf{v} \geq 0 \\ \|\mathbf{u}\|=\|\mathbf{v}\|=1}} \langle \mathbf{A}\mathbf{v}, \mathbf{u} \rangle. \end{aligned}$$

Thus,

$$\|\mathbf{X}\|_2 = \max_{\substack{\mathbf{u},\mathbf{v} \geq 0 \\ \|\mathbf{u}\|=\|\mathbf{v}\|=1}} \langle \mathbf{X}\mathbf{v}, \mathbf{u} \rangle \leq \max_{\substack{\mathbf{u},\mathbf{v} \geq 0 \\ \|\mathbf{u}\|=\|\mathbf{v}\|=1}} \langle \mathbf{Y}\mathbf{v}, \mathbf{u} \rangle = \|\mathbf{Y}\|_2. \qquad \square$$

C.7 Proof of Lemma 3.5

Proof. Since $|\nabla^2 f(\mathbf{z})| \le \mathbf{Q}$ entry-wise, zeroing the diagonal blocks preserves every off-diagonal entry, so $|\operatorname{off}(\nabla^2 f(\mathbf{z}))| \le \operatorname{off}(\mathbf{Q})$ entry-wise; Lemma C.3 then gives $\left\|\operatorname{off}(\nabla^2 f(\mathbf{z}))\right\|_2 \le \|\operatorname{off}(\mathbf{Q})\|_2$.

For the cross-coupling bound, fix $\mathbf{x}, \mathbf{h} \in \mathbb{R}^d$, and for each $i \in \{1, \dots, n\}$ let $\mathbf{J}_i \coloneqq \int_0^1 \nabla^2 f(\mathbf{x} + t\,\mathbf{P}_{-i}\mathbf{h})\,dt$. By the fundamental theorem of calculus, $\nabla_i f(\mathbf{x} + \mathbf{P}_{-i}\mathbf{h}) - \nabla_i f(\mathbf{x}) = \mathbf{U}_i^\top \mathbf{J}_i\,\mathbf{P}_{-i}\mathbf{h}$. Therefore,

$$\begin{aligned}
&\sum_{i=1}^n \|\nabla_i f(\mathbf{x} + \mathbf{P}_{-i}\mathbf{h}) - \nabla_i f(\mathbf{x})\|^2 \\
&= \sum_{i=1}^n \left\|\mathbf{U}_i^\top \mathbf{J}_i\,\mathbf{P}_{-i}\mathbf{h}\right\|^2 \\
&\overset{Lemma\ C.2}{=} \sum_{i=1}^n \left\|\mathbf{U}_i^\top \operatorname{off}(\mathbf{J}_i)\,\mathbf{h}\right\|^2 \\
&\overset{(\star)}{\le} \sum_{i=1}^n \left\|\mathbf{U}_i^\top \operatorname{off}(\mathbf{Q})\,|\mathbf{h}|\right\|^2 = \left\|\operatorname{off}(\mathbf{Q})\,|\mathbf{h}|\right\|^2 \le \|\operatorname{off}(\mathbf{Q})\|_2^2\,\|\mathbf{h}\|^2,
\end{aligned}$$

where the step marked $(\star)$ is obtained as follows. Since $\operatorname{off}(\cdot)$ is linear and $|\nabla^2 f| \le \mathbf{Q}$ entry-wise,

$$\begin{aligned}
|\operatorname{off}(\mathbf{J}_i)| &= \left|\operatorname{off}\left(\int_0^1 \nabla^2 f(\mathbf{x} + t\,\mathbf{P}_{-i}\mathbf{h})\,dt\right)\right| \\
&\le \int_0^1 |\operatorname{off}(\nabla^2 f(\mathbf{x} + t\,\mathbf{P}_{-i}\mathbf{h}))|\,dt \;\le\; \operatorname{off}(\mathbf{Q}) \quad \text{entry-wise.}
\end{aligned} \tag{15}$$

$$(15) \implies |\operatorname{off}(\mathbf{J}_i)\,\mathbf{h}| \;\le\; |\operatorname{off}(\mathbf{J}_i)|\,|\mathbf{h}| \;\le\; \operatorname{off}(\mathbf{Q})\,|\mathbf{h}| \quad \text{entry-wise.} \tag{16}$$

By Lemma C.3,

$$(16) \implies \left\|\mathbf{U}_i^\top \operatorname{off}(\mathbf{J}_i)\,\mathbf{h}\right\|_2 \;\le\; \left\|\mathbf{U}_i^\top \operatorname{off}(\mathbf{Q})\,|\mathbf{h}|\right\|_2, \tag{17}$$

where $\mathbf{U}_i^\top$ preserves the entry-wise inequality because it removes the same coordinates from both sides. Hence $M \le \|\operatorname{off}(\mathbf{Q})\|_2$. □

C.8 Proof of Section 3

Let f be a GLM of the form (10), with $\mathbf{A} \in \mathbb{R}_+^{m \times d}$ and ϕ convex and L_ϕ-smooth. Then

$$M \le L_\phi\,\left\|\operatorname{off}(\mathbf{A}^\top \mathbf{A})\right\|_2. \tag{18}$$

First, suppose that ϕ is twice continuously differentiable. The Hessian has the form $\nabla^2 f(\mathbf{x}) = \mathbf{A}^\top \mathbf{D}(\mathbf{x})\mathbf{A}$, where $\mathbf{D}(\mathbf{x})$ is diagonal with entries $\phi''(\langle \mathbf{a}_j, \mathbf{x}\rangle - b_j) \in [0, L_\phi]$. Since $\mathbf{A} \ge 0$,

$$0 \;\le\; \nabla^2 f(\mathbf{x}) = \mathbf{A}^\top \mathbf{D}(\mathbf{x})\,\mathbf{A} \;\le\; \mathbf{A}^\top (L_\phi \mathbf{I}_m)\,\mathbf{A} \;=\; L_\phi\,\mathbf{A}^\top \mathbf{A} \;=:\; \mathbf{Q} \qquad \text{entry-wise.}$$

Applying Lemma 3.5 proves the claim.

D Proofs for Section 4

In this section, we derive the nonconvex, convex, and PL convergence guarantees.

D.1 Gradient bound for PiCS

Lemma D.1. *Suppose f satisfies Assumption 3.1. For $\mathbf{x}_+^{(i)}$ satisfying the inexactness criterion* (6) *for all i,*

$$\|\nabla f(\mathbf{x}_+)\|^2 \;\le\; \left(\frac{9\lambda^2}{2} + 2M^2\right)\|\mathbf{x}_+ - \bar{\mathbf{x}}\|^2.$$

Proof. Adding and subtracting $\nabla_i f(\bar{\mathbf{x}} + U_i(\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)}))$ in each block:

$$\begin{aligned}\|\nabla f(\mathbf{x}_+)\|^2 &= \sum_{i=1}^n \|\nabla_i f(\mathbf{x}_+)\|^2 \\ &\le 2\sum_{i=1}^n \left\|\nabla_i f(\bar{\mathbf{x}} + U_i(\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)}))\right\|^2 + 2\sum_{i=1}^n \left\|\nabla_i f(\mathbf{x}_+) - \nabla_i f(\bar{\mathbf{x}} + U_i(\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)}))\right\|^2 \\ &\overset{Assumption\ 3.1}{\le} 2\sum_{i=1}^n \left\|\nabla_i f(\bar{\mathbf{x}} + U_i(\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)}))\right\|^2 + 2M^2 \|\mathbf{x}_+ - \bar{\mathbf{x}}\|^2. \end{aligned} \tag{19}$$

By (5) and $\nabla g_{i,\bar{\mathbf{x}}}(\mathbf{x}_+^{(i)}) = \nabla_i f\big(\mathbf{x}_+ + \mathbf{P}_{-i}(\bar{\mathbf{x}} - \mathbf{x}_+)\big) + \lambda(\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)})$:

$$\begin{aligned}\left\|\nabla_i f(\bar{\mathbf{x}} + U_i(\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)}))\right\| &= \left\|\lambda(\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)}) - \nabla g_{i,\bar{\mathbf{x}}}(\mathbf{x}_+^{(i)})\right\| \\ &\le \lambda\left\|\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)}\right\| + \left\|\nabla g_{i,\bar{\mathbf{x}}}(\mathbf{x}_+^{(i)})\right\| \overset{(6)}{\le} \frac{3\lambda}{2}\left\|\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)}\right\|, \end{aligned} \tag{20}$$

Substituting this into the previous inequality, we get the result. □

D.2 Proof of Lemma 4.2

Proof. From (5), $\nabla g_{i,\bar{\mathbf{x}}}(\mathbf{x}_+^{(i)}) = \nabla_i f(\bar{\mathbf{x}} + U_i(\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)})) + \lambda(\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)})$. Applying coordinate-wise convexity (Assumption 4.1) and the inexactness condition (6), we obtain

$$\begin{aligned} f(\bar{\mathbf{x}} + U_i(\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)})) - f(\bar{\mathbf{x}}) &\le \langle \nabla_i f(\bar{\mathbf{x}} + U_i(\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)})), \mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)}\rangle \\ &= -\lambda\left\|\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)}\right\|^2 + \langle \nabla g_{i,\bar{\mathbf{x}}}(\mathbf{x}_+^{(i)}), \mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)}\rangle \\ &\overset{(6)}{\le} -\frac{\lambda}{2}\left\|\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)}\right\|^2. \end{aligned} \tag{21}$$

Combining this with (4), we obtain

$$\begin{aligned} f(\mathbf{x}_+) &\le f(\bar{\mathbf{x}}) + \frac{M}{2}\|\mathbf{x}_+ - \bar{\mathbf{x}}\|^2 + \sum_{i=1}^n \big[f(\bar{\mathbf{x}} + U_i(\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)})) - f(\bar{\mathbf{x}})\big] \\ &\overset{(21)}{\le} f(\bar{\mathbf{x}}) - \frac{\lambda - M}{2}\|\mathbf{x}_+ - \bar{\mathbf{x}}\|^2 \overset{Lemma\ D.1}{\le} f(\bar{\mathbf{x}}) - \frac{(\lambda - M)\ \|\nabla f(\mathbf{x}_+)\|^2}{9\lambda^2 + 4M^2}, \end{aligned} \tag{22}$$

The preceding estimates hold for every $\lambda > M$. □

D.3 Proof of Theorem 4.3

Proof of Theorem 4.3. Using the bound from the decrease lemma (Lemma 4.2) and summing over $t = 0, \dots, T-1$ gives

$$\frac{1}{20\bar{M}}\sum_{t=1}^T \|\nabla f(\mathbf{x}_t)\|^2 \le \sum_{t=0}^{T-1}\big[f(\mathbf{x}_t) - f(\mathbf{x}_{t+1})\big] = f(\mathbf{x}_0) - f(\mathbf{x}_T) \le f(\mathbf{x}_0) - f^\star.$$

Multiplying by $\frac{20\bar{M}}{T}$, we obtain

$$\frac{1}{T}\sum_{t=1}^T \|\nabla f(\mathbf{x}_t)\|^2 \le \frac{20\bar{M}\big[f(\mathbf{x}_0) - f^\star\big]}{T}. \tag{23}$$

Thus, at least one of $\mathbf{x}_1, \dots, \mathbf{x}_T$ is ε-stationary whenever $T \ge 1$ and $T \ge \frac{20\bar{M}[f(\mathbf{x}_0) - f^\star]}{\varepsilon^2}$. If $f(\mathbf{x}_0) = f^\star$, the initial point is already stationary. □

D.4 Proof of Theorem 4.5

Proof of Theorem 4.5. Write $f_t \coloneqq f(\mathbf{x}_t) - f^\star$. Lemma 4.2 gives $f(\mathbf{x}_{t+1}) \le f(\mathbf{x}_t)$, so all iterates belong to the initial sublevel set. By convexity, $f_t \le \langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^\star \rangle \le D_0 \|\nabla f(\mathbf{x}_t)\|$. It suffices to consider $f_T > 0$. Set $A \coloneqq 20\bar{M} D_0^2$. By (7) and convexity,

$$f_t - f_{t+1} \ge \frac{\|\nabla f(\mathbf{x}_{t+1})\|^2}{20\bar{M}} \ge \frac{f_{t+1}^2}{A}. \tag{24}$$

By Lemma D.1, $\lambda_0 \le \bar{M}$, and $\|\mathbf{x}_1 - \mathbf{x}_0\| \le 2D_0$,

$$f_1 \le D_0 \|\nabla f(\mathbf{x}_1)\| \le \sqrt{5}\,\bar{M} D_0 \|\mathbf{x}_1 - \mathbf{x}_0\| \le 2\sqrt{5}\,\bar{M} D_0^2 \le A.$$

Thus, for $0 \le t < T$, monotonicity gives $0 < f_{t+1} \le f_1 \le A$, and

$$\frac{1}{f_{t+1}} - \frac{1}{f_t} \ge \frac{1}{f_{t+1}} - \frac{1}{f_{t+1} + \frac{f_{t+1}^2}{A}} = \frac{1}{A + f_{t+1}} \ge \frac{1}{2A}.$$

Summing over $t = 0, \dots, T-1$ yields

$$\frac{1}{f_T} \ge \frac{1}{f_0} + \frac{T}{2A} \ge \frac{T}{2A}, \qquad f(\mathbf{x}_T) - f^\star = f_T \le \frac{2A}{T} = \frac{40\bar{M} D_0^2}{T}. \qquad \square$$

D.5 Proof of Theorem 4.7

Proof. By Lemma 4.2 and Assumption 4.6, for each $t \ge 0$, we have:

$$f(\mathbf{x}_t) - f(\mathbf{x}_{t+1}) \overset{(7)}{\ge} \frac{\|\nabla f(\mathbf{x}_{t+1})\|^2}{20\bar{M}} \overset{\text{PL}}{\ge} \frac{\mu\,[f(\mathbf{x}_{t+1}) - f^\star]}{10\bar{M}}.$$

$\square$

E Proofs for Section 5

We first prove the key inequality 8 used by accelerated PiCS, and then use it to establish its convergence rate.

E.1 Proof of Lemma 5.1

Proof. For each i, set $\mathbf{z}_i \coloneqq \bar{\mathbf{x}} + \mathbf{U}_i(\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)})$.

Squaring (6) and summing gives

$$\begin{aligned}\frac{\lambda^2}{4}\|\bar{\mathbf{x}} - \mathbf{x}_+\|^2 &\ge \sum_{i=1}^n \left\|\nabla_i f(\mathbf{z}_i) + \lambda(\mathbf{x}_+^{(i)} - \bar{\mathbf{x}}^{(i)})\right\|^2 \\ &= \sum_{i=1}^n \|\nabla_i f(\mathbf{z}_i)\|^2 + \lambda^2\|\bar{\mathbf{x}} - \mathbf{x}_+\|^2 - 2\lambda \sum_{i=1}^n \langle \nabla_i f(\mathbf{z}_i), \bar{\mathbf{x}}^{(i)} - \mathbf{x}_+^{(i)} \rangle. \end{aligned} \tag{25}$$

The reverse triangle inequality and (6) give $\|\nabla_i f(\mathbf{z}_i)\| \ge \frac{\lambda}{2}\left\|\bar{\mathbf{x}}^{(i)} - \mathbf{x}_+^{(i)}\right\|$. Hence

$$\begin{aligned}\langle \nabla f(\mathbf{x}_+), \bar{\mathbf{x}} - \mathbf{x}_+ \rangle &= \sum_{i=1}^n \langle \nabla_i f(\mathbf{z}_i), \bar{\mathbf{x}}^{(i)} - \mathbf{x}_+^{(i)} \rangle + \sum_{i=1}^n \langle \nabla_i f(\mathbf{x}_+) - \nabla_i f(\mathbf{z}_i), \bar{\mathbf{x}}^{(i)} - \mathbf{x}_+^{(i)} \rangle \\ &\overset{(25)}{\ge} \frac{\lambda}{2}\|\bar{\mathbf{x}} - \mathbf{x}_+\|^2 - \Big(\sum_{i=1}^n \|\nabla_i f(\mathbf{x}_+) - \nabla_i f(\mathbf{z}_i)\|^2\Big)^{1/2} \|\bar{\mathbf{x}} - \mathbf{x}_+\| \overset{(3)}{\ge} \Big(\frac{\lambda}{2} - M\Big)\|\bar{\mathbf{x}} - \mathbf{x}_+\|^2. \end{aligned} \tag{26}$$

$$\langle \nabla f(\mathbf{x}_+), \bar{\mathbf{x}} - \mathbf{x}_+ \rangle \overset{\lambda \ge 4M}{\ge} \frac{\lambda}{4}\|\bar{\mathbf{x}} - \mathbf{x}_+\|^2 \overset{Lemma\ D.1,\ \lambda \ge 4M}{\ge} \frac{1}{20\lambda}\|\nabla f(\mathbf{x}_+)\|^2. \qquad \square$$

E.2 Proof of Theorem 5.2

Proof. Assume $0 < \lambda_t \le \bar{M}$ and (8). Write $g \coloneqq \nabla f(\mathbf{x}_{t+1})$ and recall $\lambda_t' a_{t+1}^2 = 2A_{t+1}$. By convexity and the updates in Algorithm 2,

$$
\begin{aligned}
& A_t f(\mathbf{x}_t) + a_{t+1} f^\star + \frac{1}{2}\left\|\mathbf{x}^\star - \mathbf{v}_t\right\|^2 \\
& \overset{\text{Assumption 4.4}}{\ge} A_t\big[f(\mathbf{x}_{t+1}) + \langle g, \mathbf{x}_t - \mathbf{x}_{t+1}\rangle\big] + a_{t+1}\big[f(\mathbf{x}_{t+1}) + \langle g, \mathbf{x}^\star - \mathbf{x}_{t+1}\rangle\big] + \frac{1}{2}\left\|\mathbf{x}^\star - \mathbf{v}_t\right\|^2 \\
& = A_t\big[f(\mathbf{x}_{t+1}) + \langle g, \mathbf{x}_t - \mathbf{x}_{t+1}\rangle\big] + a_{t+1}\big[f(\mathbf{x}_{t+1}) + \langle g, \mathbf{v}_{t+1} - \mathbf{x}_{t+1}\rangle\big] \\
& \quad + \frac{1}{2}\left\|\mathbf{v}_{t+1} - \mathbf{v}_t\right\|^2 + \frac{1}{2}\left\|\mathbf{x}^\star - \mathbf{v}_{t+1}\right\|^2 \\
& = A_{t+1} f(\mathbf{x}_{t+1}) + \frac{1}{2}\left\|\mathbf{x}^\star - \mathbf{v}_{t+1}\right\|^2 + A_{t+1}\langle g, \frac{A_t\mathbf{x}_t + a_{t+1}\mathbf{v}_t}{A_{t+1}} - \mathbf{x}_{t+1}\rangle \\
& \quad + a_{t+1}\langle g, \mathbf{v}_{t+1} - \mathbf{v}_t\rangle + \frac{1}{2}\left\|\mathbf{v}_{t+1} - \mathbf{v}_t\right\|^2 \\
& = A_{t+1} f(\mathbf{x}_{t+1}) + \frac{1}{2}\left\|\mathbf{x}^\star - \mathbf{v}_{t+1}\right\|^2 + A_{t+1}\left[\langle g, \mathbf{y}_t - \mathbf{x}_{t+1}\rangle - \frac{a_{t+1}^2}{2A_{t+1}}\left\|g\right\|^2\right] \\
& \overset{(8)}{\ge} A_{t+1} f(\mathbf{x}_{t+1}) + \frac{1}{2}\left\|\mathbf{x}^\star - \mathbf{v}_{t+1}\right\|^2.
\end{aligned}
$$

Subtracting $A_{t+1}f^\star$ gives $\Phi_{t+1} \le \Phi_t$ for $\Phi_t \coloneqq A_t[f(\mathbf{x}_t) - f^\star] + \frac{1}{2}\left\|\mathbf{x}^\star - \mathbf{v}_t\right\|^2$. Since $A_0 = 0$ and $\mathbf{v}_0 = \mathbf{x}_0$, telescoping yields

$$
A_T[f(\mathbf{x}_T) - f^\star] \le \Phi_T \le \Phi_0 = \frac{1}{2}\left\|\mathbf{x}^\star - \mathbf{x}_0\right\|^2 = \frac{1}{2}R_0^2. \tag{27}
$$

The accumulators grow as

$$
\sqrt{A_{t+1}} - \sqrt{A_t} \;=\; \frac{a_{t+1}}{\sqrt{A_{t+1}} + \sqrt{A_t}} \;\ge\; \frac{a_{t+1}}{2\sqrt{A_{t+1}}} \overset{a_{t+1} = \sqrt{\frac{2A_{t+1}}{\lambda_t'}}}{=} \frac{1}{\sqrt{2\lambda_t'}} \;\ge\; \frac{1}{\sqrt{40\bar{M}}}, \tag{28}
$$

Thus $A_T \ge \frac{T^2}{40\bar{M}}$. Combining with (27),

$$
f(\mathbf{x}_T) - f^\star \;\le\; \frac{\left\|\mathbf{x}^\star - \mathbf{x}_0\right\|^2}{2A_T} \;\le \frac{20\bar{M}R_0^2}{T^2}. \tag{29}
$$

□

F Proofs for Section 6

We construct a two-client quadratic problem and, using this hard problem, establish a lower bound on the communication.

F.1 Framework and hard instance

Definition F.1. Let $\mathcal{F}_d(M, R)$ denote the class of differentiable functions $F\colon \mathbb{R}^d \to \mathbb{R}$ that satisfy convexity (Assumption 4.4) and global cross-coupling (Assumption 3.1) with respect to the n-block partition $\mathbf{x} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)})$, whose smallest valid cross-coupling constant is at most M, and whose minimizer set satisfies $\operatorname{dist}\big(\mathbf{0}, \arg\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x})\big) \le R$. For $F \in \mathcal{F}_d(M, R)$, write $F^\star \coloneqq \min_{\mathbf{x}} F(\mathbf{x})$. An output $\hat{\mathbf{x}}$ is ε-accurate if $F(\hat{\mathbf{x}}) - F^\star \le \varepsilon$.

This section proves a lower bound on communication for the two-client specialization of the protocol in Section B. Fix $M, R > 0$ and $p \in \mathbb{N}$. We consider zero initialization $(\mathbf{x}_0, \mathbf{y}_0) = (\mathbf{0}, \mathbf{0})$.

Definition F.2. Let $T \in \mathbb{N}$ and $F \in \mathcal{F}_{2p}(M, R)$. Consider the two-client specialization $n = 2$, $d_1 = d_2 = p$, of Section B. For every communication round $t \in \{0, \dots, T-1\}$, let $(\mathbf{x}_{t,0}, \mathbf{y}_{t,0})$ be the server iterate, and write $K_t^x := K_{t,1}$ and $K_t^y := K_{t,2}$. Set $\mathcal{Z}_0^x = \mathcal{Z}_0^y := \varnothing$. The query histories satisfy

$$\mathcal{Z}_{t+1}^x := \mathcal{Z}_t^x \cup \left\{ \mathbf{z}_{t,k}^x = (\mathbf{x}_{t,k}, \mathbf{y}_{t,0}) \right\}_{k=0}^{K_t^x - 1},$$
$$\mathcal{Z}_{t+1}^y := \mathcal{Z}_t^y \cup \left\{ \mathbf{z}_{t,\ell}^y = (\mathbf{x}_{t,0}, \mathbf{y}_{t,\ell}) \right\}_{\ell=0}^{K_t^y - 1}.$$

We use $\operatorname{span}(\varnothing) = \{\mathbf{0}\}$.

Assumption F.3. For every communication round $t \in \{0, \dots, T-1\}$, the local variables queried by clients 1 and 2 satisfy, for all respective valid local indices k and ℓ,

$$\mathbf{x}_{t,k} \in \mathbf{x}_0 + \operatorname{span}\Big(\{\nabla_{\mathbf{x}} F(\mathbf{z}) \mid \mathbf{z} \in \mathcal{Z}_t^x\} \cup \{\nabla_{\mathbf{x}} F(\mathbf{z}_{t,j}^x) \mid 0 \le j < k\} \Big),$$
$$\mathbf{y}_{t,\ell} \in \mathbf{y}_0 + \operatorname{span}\Big(\{\nabla_{\mathbf{y}} F(\mathbf{z}) \mid \mathbf{z} \in \mathcal{Z}_t^y\} \cup \{\nabla_{\mathbf{y}} F(\mathbf{z}_{t,j}^y) \mid 0 \le j < \ell\} \Big).$$

Assumption F.4. For every communication round $t \in \{0, \dots, T-1\}$, the remote variables held fixed by clients 1 and 2 satisfy

$$\mathbf{y}_{t,0} \in \mathbf{y}_0 + \operatorname{span}\{\nabla_{\mathbf{y}} F(\mathbf{z}) \mid \mathbf{z} \in \mathcal{Z}_t^y\}, \qquad \mathbf{x}_{t,0} \in \mathbf{x}_0 + \operatorname{span}\{\nabla_{\mathbf{x}} F(\mathbf{z}) \mid \mathbf{z} \in \mathcal{Z}_t^x\}.$$

Assumption F.5. For every communication round $t \in \{0, \dots, T-1\}$, the candidate solution $(\mathbf{x}_{t+1,0}, \mathbf{y}_{t+1,0})$ generated by the server at the end of communication round t satisfies

$$\mathbf{x}_{t+1,0} \in \mathbf{x}_0 + \operatorname{span}\{\nabla_{\mathbf{x}} F(\mathbf{z}) \mid \mathbf{z} \in \mathcal{Z}_{t+1}^x\}, \qquad \mathbf{y}_{t+1,0} \in \mathbf{y}_0 + \operatorname{span}\{\nabla_{\mathbf{y}} F(\mathbf{z}) \mid \mathbf{z} \in \mathcal{Z}_{t+1}^y\}.$$

Definition F.6. For $m \in \mathbb{N}$, $\mathbf{C} \in \mathbb{R}^{m \times p}$, and $\mathbf{b} \in \mathbb{R}^m$, let $\alpha \ge 0$ and define the quadratic $F \colon \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ by

$$F(\mathbf{x}, \mathbf{y}) := \frac{\alpha}{2} \|\mathbf{x} - \mathbf{y}\|^2 + \langle \mathbf{C}^\top \mathbf{C} \mathbf{x}, \mathbf{y} \rangle - \frac{1}{\sqrt{2}} \langle \mathbf{C}^\top \mathbf{b}, \mathbf{x} + \mathbf{y} \rangle. \tag{30}$$

This quadratic is the hard problem used below.

The construction uses the following least-squares instance.

Proposition F.7. *Let $L, R > 0$, and let $m, p, T \in \mathbb{N}$ satisfy*

$$1 \le T \le \frac{\min\{m-1, p\} - 1}{2}.$$

Then there exists

$$\mathbf{C} \in \mathbb{R}^{m \times p}, \qquad \|\mathbf{C}\| \le L, \qquad \mathbf{b} \in \mathbb{R}^m,$$

such that, with

$$\mathcal{K}_T(\mathbf{C}^\top \mathbf{C}, \mathbf{C}^\top \mathbf{b}) := \operatorname{span}\left\{ (\mathbf{C}^\top \mathbf{C})^j \mathbf{C}^\top \mathbf{b} \;\middle|\; j = 0, \dots, T-1 \right\},$$

we have

$$\min_{\mathbf{u} \in \mathcal{K}_T(\mathbf{C}^\top \mathbf{C}, \mathbf{C}^\top \mathbf{b})} \frac{1}{2} \|\mathbf{C}\mathbf{u} - \mathbf{b}\|^2 \ge \frac{3 L^2 R^2}{32 (T+1)^2}.$$

Moreover, the linear system

$$\mathbf{C}\mathbf{u} = \mathbf{b}$$

has a solution $\mathbf{u}^\star \in \mathbb{R}^p$ satisfying

$$\|\mathbf{u}^\star\| \le R.$$

We next embed the least-squares residual into a convex two-block quadratic.

Lemma F.8. *Let F be the hard problem in* (30)*, with $2\alpha \ge \|\mathbf{C}\|^2$, and suppose $\mathbf{b} \in \operatorname{range}(\mathbf{C})$. For $\mathbf{u} := \frac{\mathbf{x}+\mathbf{y}}{\sqrt{2}}$, the function F is convex, its optimal value is $F^\star = -\frac{1}{2}\|\mathbf{b}\|^2$, and*

$$F(\mathbf{x}, \mathbf{y}) - F^\star \ge \frac{1}{2} \|\mathbf{C}\mathbf{u} - \mathbf{b}\|^2.$$

Proof. We prove the lemma by first applying a rotation transformation $\mathbf{Q}$ to $\mathbf{x}$ and $\mathbf{y}$ to make the Hessian of F block-diagonal. We define $\mathbf{Q}$ as:

$$\mathbf{Q} \coloneqq \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{I}_p & \mathbf{I}_p \\ \mathbf{I}_p & -\mathbf{I}_p\end{bmatrix} \in \mathbb{R}^{2p\times 2p}, \quad \begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix} \coloneqq \mathbf{Q}\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}\mathbf{x}+\mathbf{y}\\ \mathbf{x}-\mathbf{y}\end{bmatrix}, \qquad \mathbf{u},\mathbf{v}\in\mathbb{R}^p.$$

By applying this rotation transformation, we can rewrite F as a function of $\mathbf{u}$ and $\mathbf{v}$:

$$\begin{aligned}
F(\mathbf{x},\mathbf{y}) &\overset{(30)}{=} \frac{\alpha}{2}\left\|\sqrt{2}\mathbf{v}\right\|^2 + \frac{1}{2}\langle \mathbf{C}^\top\mathbf{C}(\mathbf{u}+\mathbf{v}), \mathbf{u}-\mathbf{v}\rangle - \langle \mathbf{C}^\top\mathbf{b}, \mathbf{u}\rangle \\
&= \alpha\|\mathbf{v}\|^2 + \frac{1}{2}\langle \mathbf{C}^\top\mathbf{C}\mathbf{u}, \mathbf{u}\rangle - \frac{1}{2}\langle \mathbf{C}^\top\mathbf{C}\mathbf{u}, \mathbf{v}\rangle \\
&\quad + \frac{1}{2}\langle \mathbf{C}^\top\mathbf{C}\mathbf{v}, \mathbf{u}\rangle - \frac{1}{2}\langle \mathbf{C}^\top\mathbf{C}\mathbf{v}, \mathbf{v}\rangle - \langle \mathbf{C}^\top\mathbf{b}, \mathbf{u}\rangle \\
&= \alpha\|\mathbf{v}\|^2 + \frac{1}{2}\|\mathbf{C}\mathbf{u}\|^2 - \frac{1}{2}\|\mathbf{C}\mathbf{v}\|^2 - \langle \mathbf{C}\mathbf{u}, \mathbf{b}\rangle + \frac{1}{2}\|\mathbf{b}\|^2 - \frac{1}{2}\|\mathbf{b}\|^2 \\
&= \frac{1}{2}\|\mathbf{C}\mathbf{u}-\mathbf{b}\|^2 + \frac{1}{2}\langle (2\alpha\mathbf{I}_p - \mathbf{C}^\top\mathbf{C})\mathbf{v}, \mathbf{v}\rangle - \frac{1}{2}\|\mathbf{b}\|^2
\end{aligned}$$

Since $2\alpha \geq \|\mathbf{C}\|^2$, both quadratic terms are convex and the second is nonnegative. Thus,

$$\geq \frac{1}{2}\|\mathbf{C}\mathbf{u}-\mathbf{b}\|^2 - \frac{1}{2}\|\mathbf{b}\|^2.$$

Thus, $F^\star \geq -\frac{1}{2}\|\mathbf{b}\|^2$. This lower bound is attained because the two quadratic terms are minimized at $\mathbf{C}\mathbf{u} = \mathbf{b}$ and $\mathbf{v} = 0$, respectively. Therefore

$$F^\star = -\frac{1}{2}\|\mathbf{b}\|^2. \qquad \square$$

We now choose $\alpha \coloneqq \frac{\|\mathbf{C}\|^2}{2}$, the smallest value satisfying $2\alpha \geq \|\mathbf{C}\|^2$. The cross-coupling follows from the paper's exact quadratic characterization.

Lemma F.9. *Let $M_F > 0$, and let F be the hard problem in* (30) *with $\alpha = \frac{\|\mathbf{C}\|^2}{2}$ and $\|\mathbf{C}\| \leq \sqrt{2M_F}$. By the proof of Theorem 3.3, the smallest cross-coupling constant of F is $\frac{\|\mathbf{C}\|^2}{2}$, and*

$$\frac{\|\mathbf{C}\|^2}{2} \leq M_F.$$

Proof of Lemma F.9.

$$\operatorname{off}(\nabla^2 F) = \begin{bmatrix} 0 & \mathbf{C}^\top\mathbf{C} - \alpha\mathbf{I}_p \\ \mathbf{C}^\top\mathbf{C} - \alpha\mathbf{I}_p & 0\end{bmatrix}.$$

Therefore,

$$\left\|\operatorname{off}(\nabla^2 F)\right\| = \left\|\mathbf{C}^\top\mathbf{C} - \alpha\mathbf{I}_p\right\| = \left\|\mathbf{C}^\top\mathbf{C} - \frac{\|\mathbf{C}\|^2}{2}\mathbf{I}_p\right\| = \frac{\|\mathbf{C}\|^2}{2} \leq M_F. \qquad \square$$

F.2 Krylov-space bounds

The local span restriction first controls all oracle calls made within one communication round.

Lemma F.10. *Let $T \in \mathbb{N}$, let F be the hard problem in* (30)*, and suppose the local query points satisfy Assumption F.3 for T communication rounds. Set $\mathcal{K}_0 \coloneqq \{\mathbf{0}\}$, and write $\mathcal{K}_r \coloneqq \mathcal{K}_r(\mathbf{C}^\top\mathbf{C}, \mathbf{C}^\top\mathbf{b})$. Fix $0 \leq t < T$, and suppose $K_t^x, K_t^y > 0$. Then*

$$\left.\begin{array}{c}\mathbf{x}_{t,0}, \mathbf{y}_{t,0} \in \mathcal{K}_t, \\ \nabla_{\mathbf{x}} F(\mathbf{z}) \in \mathcal{K}_t \quad \forall \mathbf{z} \in \mathcal{Z}_t^x, \\ \nabla_{\mathbf{y}} F(\mathbf{z}) \in \mathcal{K}_t \quad \forall \mathbf{z} \in \mathcal{Z}_t^y\end{array}\right\} \implies \begin{array}{c}\mathbf{x}_{t,K_t^x-1}, \mathbf{y}_{t,K_t^y-1} \in \mathcal{K}_{t+1}, \\ \nabla_{\mathbf{x}} F(\mathbf{z}) \in \mathcal{K}_{t+1} \quad \forall \mathbf{z} \in \mathcal{Z}_{t+1}^x, \\ \nabla_{\mathbf{y}} F(\mathbf{z}) \in \mathcal{K}_{t+1} \quad \forall \mathbf{z} \in \mathcal{Z}_{t+1}^y.\end{array}$$

Proof. We first use induction over the local queries of client 1. Suppose, for some $0 \le k < K_t^x - 1$, that

$$\mathbf{x}_{t,k} \in \mathcal{K}_{t+1}, \qquad \nabla_{\mathbf{x}} F(\mathbf{z}) \in \mathcal{K}_{t+1} \quad \forall \mathbf{z} \in \mathcal{Z}_t^x \cup \{\mathbf{z}_{t,j}^x : 0 \le j \le k\}.$$

Then

$$\mathbf{x}_{t,k+1} \overset{Assumption\ F.3}{\in} \mathbf{x}_0 + \operatorname{span}\Big(\{\nabla_{\mathbf{x}} F(\mathbf{z}) : \mathbf{z} \in \mathcal{Z}_t^x\} \cup \{\nabla_{\mathbf{x}} F(\mathbf{z}_{t,j}^x) : 0 \le j \le k\}\Big) \subseteq \mathcal{K}_{t+1}.$$

Consequently,

$$\nabla_{\mathbf{x}} F(\mathbf{z}_{t,k+1}^x) = \underbrace{\alpha(\mathbf{x}_{t,k+1} - \mathbf{y}_{t,0})}_{\in \mathcal{K}_{t+1}} + \underbrace{\mathbf{C}^\top \mathbf{C} \mathbf{y}_{t,0}}_{\in \mathcal{K}_{t+1}} - \underbrace{\frac{1}{\sqrt{2}} \mathbf{C}^\top \mathbf{b}}_{\in \mathcal{K}_1} \in \mathcal{K}_{t+1}.$$

Therefore, the induction step holds for $k+1$. The induction claim holds for $k = 0$, since $\mathbf{x}_{t,0} \in \mathcal{K}_t \subseteq \mathcal{K}_{t+1}$ and

$$\nabla_{\mathbf{x}} F(\mathbf{z}_{t,0}^x) = \underbrace{\alpha(\mathbf{x}_{t,0} - \mathbf{y}_{t,0})}_{\in \mathcal{K}_t \subseteq \mathcal{K}_{t+1}} + \underbrace{\mathbf{C}^\top \mathbf{C} \mathbf{y}_{t,0}}_{\in \mathcal{K}_{t+1}} - \underbrace{\frac{1}{\sqrt{2}} \mathbf{C}^\top \mathbf{b}}_{\in \mathcal{K}_1} \in \mathcal{K}_{t+1}.$$

Together with the hypothesis, this gives

$$\nabla_{\mathbf{x}} F(\mathbf{z}) \in \mathcal{K}_{t+1} \quad \forall \mathbf{z} \in \mathcal{Z}_t^x \cup \{\mathbf{z}_{t,0}^x\}.$$

This proves the base case. At $k = K_t^x - 1$,

$$\mathbf{x}_{t,K_t^x-1} \in \mathcal{K}_{t+1}, \qquad \nabla_{\mathbf{x}} F(\mathbf{z}) \in \mathcal{K}_{t+1} \quad \forall \mathbf{z} \in \mathcal{Z}_t^x \cup \{\mathbf{z}_{t,j}^x : 0 \le j < K_t^x\} = \mathcal{Z}_{t+1}^x.$$

The same induction for client 2 gives

$$\mathbf{y}_{t,K_t^y-1} \in \mathcal{K}_{t+1}, \qquad \nabla_{\mathbf{y}} F(\mathbf{z}) \in \mathcal{K}_{t+1} \quad \forall \mathbf{z} \in \mathcal{Z}_{t+1}^y. \qquad \square$$

The span restriction, therefore, permits one additional Krylov level in each communication round, regardless of the number of local steps.

Lemma F.11. *Let $T \in \mathbb{N}$, let F be the hard problem in* (30)*, and consider the two-client specialization of the communication protocol in Section B. Suppose its local query points, remote variables, and server iterates satisfy Assumptions F.3 to F.5 for T communication rounds, with $K_t^x, K_t^y > 0$ for every $0 \le t < T$. Set $\mathcal{K}_0 \coloneqq \{\mathbf{0}\}$, and write $\mathcal{K}_r \coloneqq \mathcal{K}_r(\mathbf{C}^\top \mathbf{C}, \mathbf{C}^\top \mathbf{b})$. Then*

$$\boxed{\mathbf{u}_T \coloneqq \frac{\mathbf{x}_{T,0} + \mathbf{y}_{T,0}}{\sqrt{2}} \in \mathcal{K}_T}.$$

Proof. We use induction on the communication round t to prove that

$$\mathbf{x}_{t,0}, \mathbf{y}_{t,0} \in \mathcal{K}_t, \qquad \nabla_{\mathbf{x}} F(\mathbf{z}) \in \mathcal{K}_t \quad \forall \mathbf{z} \in \mathcal{Z}_t^x, \qquad \nabla_{\mathbf{y}} F(\mathbf{z}) \in \mathcal{K}_t \quad \forall \mathbf{z} \in \mathcal{Z}_t^y.$$

Suppose, for some $0 \le t < T$, that

$$\mathbf{x}_{t,0}, \mathbf{y}_{t,0} \in \mathcal{K}_t, \qquad \nabla_{\mathbf{x}} F(\mathbf{z}) \in \mathcal{K}_t \quad \forall \mathbf{z} \in \mathcal{Z}_t^x, \qquad \nabla_{\mathbf{y}} F(\mathbf{z}) \in \mathcal{K}_t \quad \forall \mathbf{z} \in \mathcal{Z}_t^y,$$

by Lemma F.10,

$$\mathbf{x}_{t,K_t^x-1}, \mathbf{y}_{t,K_t^y-1} \in \mathcal{K}_{t+1}, \quad \nabla_{\mathbf{x}} F(\mathbf{z}) \in \mathcal{K}_{t+1} \quad \forall \mathbf{z} \in \mathcal{Z}_{t+1}^x, \quad \nabla_{\mathbf{y}} F(\mathbf{z}) \in \mathcal{K}_{t+1} \quad \forall \mathbf{z} \in \mathcal{Z}_{t+1}^y.$$

Therefore,

$$\mathbf{x}_{t+1,0} \overset{Assumption\ F.5}{\in} \mathbf{x}_0 + \operatorname{span}\{\nabla_{\mathbf{x}} F(\mathbf{z}) : \mathbf{z} \in \mathcal{Z}_{t+1}^x\} \subseteq \mathcal{K}_{t+1},$$

and

$$\mathbf{y}_{t+1,0} \overset{Assumption\ F.5}{\in} \mathbf{y}_0 + \operatorname{span}\{\nabla_{\mathbf{y}} F(\mathbf{z}) : \mathbf{z} \in \mathcal{Z}_{t+1}^y\} \subseteq \mathcal{K}_{t+1}.$$

Thus, the induction hypothesis holds with $t+1$. Hence, for every $0 \le t < T$,

$$\mathbf{x}_{t,K_t^x-1} \in \mathcal{K}_{t+1}, \qquad \mathbf{y}_{t,K_t^y-1} \in \mathcal{K}_{t+1}.$$

The initialization and the empty query histories give $\mathbf{x}_{0,0} = \mathbf{y}_{0,0} = \mathbf{0} \in \mathcal{K}_0$, $\mathcal{Z}_0^x = \mathcal{Z}_0^y = \varnothing$, which proves the base case $t = 0$. At $t = T$, the induction gives $\mathbf{x}_{T,0}, \mathbf{y}_{T,0} \in \mathcal{K}_T$, and therefore

$$\mathbf{u}_T = \frac{\mathbf{x}_{T,0} + \mathbf{y}_{T,0}}{\sqrt{2}} \in \frac{1}{\sqrt{2}} \mathcal{K}_T + \frac{1}{\sqrt{2}} \mathcal{K}_T = \mathcal{K}_T. \qquad \square$$

F.3 Proof of Theorem 6.1

The preceding ingredients give the lower bound with the paper's cross-coupling parameter M.

Proof of Theorem 6.1. Set $m := p+1$. Since $p \geq 2T+1$, we have $1 \leq T \leq (\min\{m-1, p\}-1)/2$. Apply Proposition F.7 with $L = \sqrt{2M}$. There exist $\mathbf{C} \in \mathbb{R}^{m\times p}$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{u}^\star \in \mathbb{R}^p$ such that

$$\|\mathbf{C}\| \leq \sqrt{2M}, \qquad \mathbf{C}\mathbf{u}^\star = \mathbf{b}, \qquad \|\mathbf{u}^\star\| \leq R,$$

and

$$\min_{\mathbf{u}\in\mathcal{K}_T(\mathbf{C}^\top\mathbf{C},\mathbf{C}^\top\mathbf{b})} \frac{1}{2}\|\mathbf{C}\mathbf{u}-\mathbf{b}\|^2 \geq \frac{3MR^2}{16(T+1)^2}.$$

Using these $\mathbf{C}$ and $\mathbf{b}$, set $\alpha := \|\mathbf{C}\|^2/2$ and define F by (30). Write $(\mathbf{x}_T, \mathbf{y}_T) = (\mathbf{x}_{T,0}, \mathbf{y}_{T,0})$ for the final server iterate. Since

$$\mathbf{C}\mathbf{u}^\star = \mathbf{b} \Longrightarrow \mathbf{b} \in \mathrm{range}(\mathbf{C}).$$

By Lemma F.8, F is convex and $F^\star = -\frac{1}{2}\|\mathbf{b}\|^2$. Set $\mathbf{x}^\star = \mathbf{y}^\star := \frac{\mathbf{u}^\star}{\sqrt{2}}$. Then $F(\mathbf{x}^\star, \mathbf{y}^\star) = -\frac{1}{2}\|\mathbf{b}\|^2 = F^\star$. Consequently,

$$\|\mathbf{z}^*\| \leq \sqrt{\|\mathbf{x}^\star\|^2 + \|\mathbf{y}^\star\|^2} = \|\mathbf{u}^\star\| \leq R.$$

By Lemma F.9, the smallest cross-coupling constant of F is $M_F = \frac{\|\mathbf{C}\|^2}{2} \leq M$. Thus $F \in \mathcal{F}_{2p}(M, R)$. Hence, without loss of generality, $K_t^x, K_t^y > 0$ for every $0 \leq t < T$. By Lemma F.11,

$$\mathbf{u}_T = \frac{\mathbf{x}_{T,0}+\mathbf{y}_{T,0}}{\sqrt{2}} \in \mathcal{K}_T(\mathbf{C}^\top\mathbf{C}, \mathbf{C}^\top\mathbf{b}).$$

The embedding and the hard residual bound now give

$$\begin{aligned} F(\mathbf{x}_{T,0}, \mathbf{y}_{T,0}) - F^\star &\overset{Lemma\ F.8}{\geq} \frac{1}{2}\|\mathbf{C}\mathbf{u}_T - \mathbf{b}\|^2 \\ &\geq \min_{\mathbf{u}\in\mathcal{K}_T(\mathbf{C}^\top\mathbf{C},\mathbf{C}^\top\mathbf{b})} \frac{1}{2}\|\mathbf{C}\mathbf{u}-\mathbf{b}\|^2 \overset{Proposition\ F.7}{\geq} \frac{3MR^2}{16(T+1)^2}. \end{aligned}$$

Consequently, an ε-accurate output satisfies

$$F(\mathbf{x}_{T,0}, \mathbf{y}_{T,0}) - F^\star \leq \varepsilon \Longrightarrow \varepsilon \geq \frac{3MR^2}{16(T+1)^2} \Longrightarrow T+1 \geq \frac{\sqrt{3}}{4}R\sqrt{\frac{M}{\varepsilon}}. \qquad \square$$

G Proofs for Section 7

G.1 Gradient descent (GD)

Algorithm 1 allows any local solver returning a block update that meets (6), and the communication guarantees of Theorems 4.3, 4.5 and 4.7 hold for every such choice. Algorithm 3 uses local gradient descent; Lemma G.2 bounds the number of gradient evaluations per client needed to satisfy (6).

Algorithm 3 PiCS with Gradient Descent Local Solver

Input: initial iterate $\mathbf{x}_0 \in \mathbb{R}^d$, number of communication rounds T, regularizer $\lambda_t > 0$, block smoothness constants $\{L_i\}_{i=1}^n$
for $t = 0, 1, \ldots, T-1$ **do**
  Server broadcasts $\mathbf{x}_t$ and λ_t to all n workers
  In parallel for each block $i \in \{1, \ldots, n\}$**:**
    form local objective $g_{i,\mathbf{x}_t}(\mathbf{x}^{(i)}) = f\big(\mathbf{x}_t + \mathbf{U}_i(\mathbf{x}^{(i)} - \mathbf{x}_t^{(i)})\big) + \frac{\lambda_t}{2} \left\|\mathbf{x}^{(i)} - \mathbf{x}_t^{(i)}\right\|^2$
    initialize $\mathbf{x}_{t,0}^{(i)} \leftarrow \mathbf{x}_t^{(i)}$, $k \leftarrow 0$
    repeat *(gradient descent on $g_{i,\mathbf{x}_t}$)*
      $\mathbf{x}_{t,k+1}^{(i)} \leftarrow \mathbf{x}_{t,k}^{(i)} - \frac{1}{L_i + \lambda_t} \nabla g_{i,\mathbf{x}_t}(\mathbf{x}_{t,k}^{(i)})$
      $k \leftarrow k + 1$
    until $\left\|\nabla g_{i,\mathbf{x}_t}(\mathbf{x}_{t,k}^{(i)})\right\| \le \frac{\lambda_t}{2} \left\|\mathbf{x}_{t,k}^{(i)} - \mathbf{x}_t^{(i)}\right\|$ (cf. (6))
    set $\mathbf{x}_{t+1}^{(i)} \leftarrow \mathbf{x}_{t,k}^{(i)}$
  Server aggregates $\mathbf{x}_{t+1} = (\mathbf{x}_{t+1}^{(1)}, \ldots, \mathbf{x}_{t+1}^{(n)})$
Output: $\mathbf{x}_T$

We first record the standard gradient-norm rate of gradient descent on smooth convex functions.

Lemma G.1. *Let g be L-smooth and convex with a minimizer $\mathbf{x}_g^\star$. Gradient descent with step size $\frac{1}{L}$, started from any $\bar{\mathbf{x}} \in \mathbb{R}^d$ and run for $N \ge 1$ steps, produces an iterate $\mathbf{x}_+$ with*

$$\|\nabla g(\mathbf{x}_+)\| \;\le\; \frac{2LR}{N}, \qquad R \coloneqq \left\|\bar{\mathbf{x}} - \mathbf{x}_g^\star\right\|. \tag{31}$$

See Nesterov (2018).

Lemma G.2. *At outer iteration t, assume $\lambda_t > 0$. With the local solver and termination criterion of Section G.1, the first iteration N_i at which worker i stops satisfies*

$$N_i \;\le\; \left\lceil \frac{6(L_i + \lambda_t)}{\lambda_t} \right\rceil. \tag{32}$$

Proof. For brevity write $g \coloneqq g_{i,\mathbf{x}_t}$, $\bar{\mathbf{x}} \coloneqq \mathbf{x}_t^{(i)} \in \mathbb{R}^{d_i}$, and $L \coloneqq L_i + \lambda_t$; let $\mathbf{x}_g^\star \coloneqq \arg\min_{\mathbf{z} \in \mathbb{R}^{d_i}} g(\mathbf{z})$ and $R \coloneqq \left\|\bar{\mathbf{x}} - \mathbf{x}_g^\star\right\|$. The local solver (Section G.1) is gradient descent on g with step size $\frac{1}{L}$, started from $\bar{\mathbf{x}}$, and run until its output $\mathbf{x}_+$ satisfies the inexactness criterion $\|\nabla g(\mathbf{x}_+)\| \le \frac{\lambda_t}{2} \|\mathbf{x}_+ - \bar{\mathbf{x}}\|$. Recall g is L-smooth and λ_t-strongly convex.

We first use strong convexity to reduce the inexactness criterion to a gradient-norm bound. By λ_t-strong convexity, $\left\|\mathbf{x}_+ - \mathbf{x}_g^\star\right\| \le \frac{\|\nabla g(\mathbf{x}_+)\|}{\lambda_t}$, and the reverse triangle inequality then gives $\|\mathbf{x}_+ - \bar{\mathbf{x}}\| \ge R - \frac{\|\nabla g(\mathbf{x}_+)\|}{\lambda_t}$. Hence, if $\|\nabla g(\mathbf{x}_+)\| \le \frac{\lambda_t R}{3}$, then $\|\mathbf{x}_+ - \bar{\mathbf{x}}\| \ge \frac{2R}{3}$ and

$$\|\nabla g(\mathbf{x}_+)\| \;\le\; \frac{\lambda_t R}{3} \;=\; \frac{\lambda_t}{2} \cdot \frac{2R}{3} \;\le\; \frac{\lambda_t}{2} \|\mathbf{x}_+ - \bar{\mathbf{x}}\|,$$

satisfying the criterion. It thus suffices to ensure

$$\|\nabla g(\mathbf{x}_+)\| \;\le\; \frac{\lambda_t R}{3}. \tag{33}$$

By Lemma G.1, after N inner GD steps the output $\mathbf{x}_+$ satisfies $\|\nabla g(\mathbf{x}_+)\| \le \frac{2LR}{N}$. This is below $\frac{\lambda_t R}{3}$ as soon as $N \ge \left\lceil \frac{6L}{\lambda_t} \right\rceil$, so the inexactness criterion holds by that step. Hence $N_i \le \left\lceil \frac{6(L_i + \lambda_t)}{\lambda_t} \right\rceil$, giving (32). □

G.2 ACCELERATED SOLVER (ARM)

This appendix is self-contained and independent of the remainder of the paper. We revisit a variant of accelerated gradient descent tailored to *gradient-norm minimization*: the accumulative regularization method (ARM) of Lan et al. (2023). Because the section stands on its own, every symbol below is defined locally and is *not* to be identified with any identically named quantity in the main text.

Let $f\colon \mathbb{R}^p \to \mathbb{R}$ be convex and L-smooth with respect to the Euclidean norm $\|\cdot\|$. Consider the minimization problem

$$\min_{\mathbf{w}\in\mathbb{R}^p} \; f(\mathbf{w}), \tag{34}$$

with nonempty set of minimizers $\mathcal{W}^\star$ and a minimizer $\mathbf{w}^\star \in \mathcal{W}^\star$. Given an initial point $\mathbf{v} \in \mathbb{R}^p$, write $\operatorname{dist}(\mathbf{v}, \mathcal{W}^\star) = \min_{\mathbf{w}^\star\in\mathcal{W}^\star} \|\mathbf{v} - \mathbf{w}^\star\|$ for its distance to the optimal set. The objective is to drive the *gradient norm* below a target $\varepsilon > 0$: to compute a point $\mathbf{w} \in \mathbb{R}^p$ such that

$$\|\nabla f(\mathbf{w})\| \;\le\; \varepsilon, \tag{35}$$

We target a *relative* accuracy, with the tolerance proportional to the distance to optimality. Fixing a relative accuracy $\delta > 0$ and any upper bound $D \ge \operatorname{dist}(\mathbf{v}, \mathcal{W}^\star)$ on that distance, we instantiate the subroutine with

$$\varepsilon \;=\; \delta\, D. \tag{36}$$

When the bound is tight, $D = \operatorname{dist}(\mathbf{v}, \mathcal{W}^\star)$, and the target (35) becomes $\|\nabla f(\mathbf{w})\| \le \delta \operatorname{dist}(\mathbf{v}, \mathcal{W}^\star)$.

Algorithm 4 restates the accumulative regularization method of Lan et al. (2023). It solves a sequence of τ proximal subproblems with geometrically increasing regularization $\sigma^{(k)}$; each subproblem is solved approximately by an inner run of Nesterov's accelerated gradient method. The routine takes the target accuracy ε, the smoothness constant L, and a distance bound $D \ge \operatorname{dist}(\mathbf{v}, \mathcal{W}^\star)$ as inputs; its stage count, regularization schedule, interpolation weight, and inner budgets are exactly those of the guarantee (37).

Algorithm 4 $\mathrm{ARM}(\nabla f, \mathbf{v}, \varepsilon \mid L, D)$ — accumulative regularization method (Lan et al., 2023)

Set $\tau = 1 + \lceil \log_4(\frac{LD}{\varepsilon}) \rceil$.
Set $\sigma^{(k)} = 4^{\,k-2}\,\frac{\varepsilon}{D}$ and $N_k = \lceil 16\sqrt{\frac{L}{\sigma^{(k)}}} \rceil$ for $k = 1, \dots, \tau$.
Initialize $\bar{\mathbf{w}}^{(0)} = \mathbf{w}^{(0)} = \mathbf{v}$.
for $k = 1, \dots, \tau$ **do**
  $\gamma^{(k)} = \frac{3}{4}$
  $\bar{\mathbf{w}}^{(k)} = (1 - \gamma^{(k)})\, \bar{\mathbf{w}}^{(k-1)} + \gamma^{(k)}\, \mathbf{w}^{(k-1)}$
  % Begin inner subroutine: Nesterov's accelerated gradient method
  Initialize $\mathbf{x}_k^{(0)} = \mathbf{w}^{(k-1)}$, $\mathbf{y}_k^{(0)} = \mathbf{w}^{(k-1)}$, $t_0 = 1$, and $L_k = L + \sigma^{(k)}$.
  for $i = 0, \dots, N_k - 1$ **do**
    $\nabla f^{(k)}(\mathbf{y}_k^{(i)}) = \nabla f(\mathbf{y}_k^{(i)}) + \sigma^{(k)}\big(\mathbf{y}_k^{(i)} - \bar{\mathbf{w}}^{(k)}\big)$
    $\mathbf{x}_k^{(i+1)} = \mathbf{y}_k^{(i)} - \frac{1}{L_k} \nabla f^{(k)}(\mathbf{y}_k^{(i)})$
    $t_{i+1} = \frac{1+\sqrt{1+4t_i^2}}{2}$
    $\mathbf{y}_k^{(i+1)} = \mathbf{x}_k^{(i+1)} + \frac{t_i - 1}{t_{i+1}}\big(\mathbf{x}_k^{(i+1)} - \mathbf{x}_k^{(i)}\big)$
  % End inner subroutine
  $\mathbf{w}^{(k)} = \mathbf{x}_k^{(N_k)}$
return $\mathbf{w}^{(\tau)}$

The following gradient-norm guarantee for Algorithm 4 is Lan et al. (2023, Theorem 2.1). The inner accelerated routine enters only through an absolute constant $c_{\mathcal{A}}$, which equals 2 for the Nesterov method used here. Accordingly, the inner budget $N_k = \lceil 16\sqrt{\frac{L}{\sigma^{(k)}}} \rceil$ of Algorithm 4 is the budget N_k of (37) evaluated at $c_{\mathcal{A}} = 2$, since $8\sqrt{2c_{\mathcal{A}}} = 16$.

Theorem G.3. *Let f be convex and L-smooth, let $D \geq \operatorname{dist}(\mathbf{v}, \mathcal{W}^\star)$ be an upper bound on the distance from $\mathbf{v}$ to the optimal set, and let the target satisfy $\varepsilon \leq LD$. With the stage count, regularization schedule, interpolation weights, and inner budgets*

$$\tau = 1 + \lceil \log_4(\frac{LD}{\varepsilon}) \rceil, \quad \sigma^{(k)} = 4^{\,k-2}\,\frac{\varepsilon}{D}, \quad \gamma^{(k)} = \frac{3}{4}, \quad N_k = \left\lceil 8\sqrt{\frac{2c_{\mathcal{A}}L}{\sigma^{(k)}}}\,\right\rceil, \tag{37}$$

Algorithm 4 returns a point $\mathbf{w}^{(\tau)}$ satisfying

$$\left\|\nabla f(\mathbf{w}^{(\tau)})\right\| \;\leq\; \varepsilon, \tag{38}$$

using at most $2\big(1 + 8\sqrt{2c_{\mathcal{A}}}\big)\sqrt{\frac{LD}{\varepsilon}}$ gradient evaluations of ∇f.

The proof is given in Lan et al. (2023, Theorem 2.1).

The schedule (37) and the gradient budget depend on ε and D only through the ratio $\frac{\varepsilon}{D}$. With the relative target $\varepsilon = \delta D$ of (36) this ratio equals δ, so the method is scale-free—it requires only the relative accuracy δ and the smoothness L, not the distance bound D or the problem scale—and the gradient budget becomes

$$\sum_{k=1}^{\tau} N_k \;\leq\; 2\big(1 + 8\sqrt{2c_{\mathcal{A}}}\big)\sqrt{\frac{L}{\delta}}, \tag{39}$$

independent of D, with the output obeying $\left\|\nabla f(\mathbf{w}^{(\tau)})\right\| \leq \delta D$; when the bound is tight, $D = \operatorname{dist}(\mathbf{v}, \mathcal{W}^\star)$ and this is $\leq \delta \operatorname{dist}(\mathbf{v}, \mathcal{W}^\star)$.

We now record the per-worker bound when the local solver is the accumulative regularization method of Theorem G.3 rather than gradient descent. Because that method minimizes the gradient norm at the optimal accelerated rate, the per-worker count drops from $\mathcal{O}(\frac{L_i+\lambda_t}{\lambda_t})$ to $\mathcal{O}(\sqrt{\frac{L_i+\lambda_t}{\lambda_t}})$.

Lemma G.4. *At outer iteration t, assume $\lambda_t > 0$. With the local solver replaced by the accumulative regularization method (Algorithm 4, inner constant $c_{\mathcal{A}} = 2$) run at relative accuracy $\delta = \frac{\lambda_t}{3}$, and the termination criterion* (6)*, the first iteration N_i at which worker i stops satisfies*

$$N_i \;\leq\; 34\sqrt{\frac{3(L_i + \lambda_t)}{\lambda_t}} \;=\; \mathcal{O}\left(\sqrt{\frac{L_i + \lambda_t}{\lambda_t}}\right). \tag{40}$$

Proof. Write $g := g_{i,\mathbf{x}_t}$, $\bar{\mathbf{x}} := \mathbf{x}_t^{(i)} \in \mathbb{R}^{d_i}$, $L := L_i + \lambda_t$, $\mathbf{x}_g^\star := \arg\min_{\mathbf{z}\in\mathbb{R}^{d_i}} g(\mathbf{z})$, and $D := \left\|\bar{\mathbf{x}} - \mathbf{x}_g^\star\right\| = \operatorname{dist}(\bar{\mathbf{x}}, \arg\min g)$; recall g is L-smooth and λ_t-strongly convex, with no nonsmooth term. If $D = 0$, the initial point already satisfies (6), so $N_i = 0$. Assume $D > 0$. By the same strong-convexity reduction as in the proof of Lemma G.2, the inexactness criterion (6) is satisfied once

$$\|\nabla g(\mathbf{x}_+)\| \;\leq\; \frac{\lambda_t D}{3},$$

i.e. once g is minimized to relative gradient-norm accuracy $\delta = \frac{\lambda_t}{3}$ measured against the distance D. Apply Theorem G.3 to g with smoothness $L = L_i + \lambda_t$, the tight distance bound D (exact, as the strongly convex g has a unique minimizer), and relative accuracy $\delta = \frac{\lambda_t}{3}$: the hypothesis $\varepsilon = \delta D = \frac{\lambda_t D}{3} \leq LD$ holds because $\frac{\lambda_t}{3} \leq L_i + \lambda_t$, and the schedule (37) depends only on δ and L, hence is independent of the unknown D. The relative gradient budget (39) then bounds the inner gradient steps by

$$\begin{aligned} N_i &\leq 2\big(1 + 8\sqrt{2c_{\mathcal{A}}}\big)\sqrt{\frac{L}{\delta}} = 2\big(1 + 8\sqrt{2\cdot 2}\big)\sqrt{\frac{3(L_i + \lambda_t)}{\lambda_t}} \\ &= 34\sqrt{\frac{3(L_i + \lambda_t)}{\lambda_t}}, \end{aligned}$$

where $c_{\mathcal{A}} = 2$ for the Nesterov inner loop of Algorithm 4. □

H PROOFS FOR SECTION 7

We relate cross-coupling to the smoothness, separability, and other parameters used by the baselines, then derive the complexity bounds in Table 1.

H.1 CROSS-COUPLING AND PARTIAL SEPARABILITY

Definition H.1. A function $f\colon \mathbb{R}^d \to \mathbb{R}$ is *ω-partially separable* if it admits a decomposition $f(\mathbf{x}) = \sum_{J\in\mathcal{J}} f_J(\mathbf{x})$, where $\mathcal{J}$ is a finite family of nonempty subsets of $\{1,\dots,n\}$, each f_J depends only on the blocks $\{\mathbf{x}^{(i)} : i \in J\}$. We define $\omega := \max_{J\in\mathcal{J}} |J|$.

The partial-separability bound is proved in Section H.1.

Lemma H.2. *Let $f = \sum_{J\in\mathcal{J}} f_J$ be ω-partially separable, where every f_J is convex and twice continuously differentiable. Suppose f satisfies coordinate-wise smoothness (Assumption 7.1) and is L-smooth, and define $L_{\max} := \max_i L_i$. Then*

$$\boxed{M \le \omega L_{\max}}. \tag{41}$$

Proof. Fix $\mathbf{z} \in \mathbb{R}^d$ and set $\mathbf{H} := \nabla^2 f(\mathbf{z})$. For $J \in \mathcal{J}$, set $\mathbf{H}_J := \nabla^2 f_J(\mathbf{z}) \succeq 0$. Then $\mathbf{H} = \sum_{J\in\mathcal{J}} \mathbf{H}_J$, and for every $\mathbf{w} \in \mathbb{R}^d$,

$$\begin{aligned}
\langle \mathbf{H}\mathbf{w}, \mathbf{w}\rangle &= \sum_{J\in\mathcal{J}} \sum_{i,j\in J} \langle (\mathbf{H}_J)_{ij}\mathbf{w}^{(j)}, \mathbf{w}^{(i)}\rangle \\
&\le \sum_{J\in\mathcal{J}} \sum_{i,j\in J} \sqrt{\langle (\mathbf{H}_J)_{ii}\mathbf{w}^{(i)}, \mathbf{w}^{(i)}\rangle}\, \sqrt{\langle (\mathbf{H}_J)_{jj}\mathbf{w}^{(j)}, \mathbf{w}^{(j)}\rangle} \\
&= \sum_{J\in\mathcal{J}} \left(\sum_{i\in J} \sqrt{\langle (\mathbf{H}_J)_{ii}\mathbf{w}^{(i)}, \mathbf{w}^{(i)}\rangle} \right)^2 \\
&\le \sum_{J\in\mathcal{J}} |J| \sum_{i\in J} \langle (\mathbf{H}_J)_{ii}\mathbf{w}^{(i)}, \mathbf{w}^{(i)}\rangle \\
&\le \omega \sum_{i=1}^{n} \langle (\mathbf{H})_{ii}\mathbf{w}^{(i)}, \mathbf{w}^{(i)}\rangle \\
&\le \omega \sum_{i=1}^{n} L_i \left\|\mathbf{w}^{(i)}\right\|^2 \le \omega L_{\max} \|\mathbf{w}\|^2 .
\end{aligned} \tag{42}$$

Taking the supremum over $\|\mathbf{w}\| = 1$ and $\mathbf{z} \in \mathbb{R}^d$ gives $L \le \omega L_{\max}$. Combining this with $M \le L$ from Lemma C.1 proves the claim. □

H.2 CROSS-COUPLING UNDER MATRIX SMOOTHNESS

Definition H.3. Following Richtárik & Takáč (2016a, Eq. (2)), let $\mathbf{S} \in \mathbb{R}^{d\times d}$ be symmetric and positive definite. A differentiable function $f\colon \mathbb{R}^d \to \mathbb{R}$ is **S**-*smooth* if

$$f(\mathbf{x}+\mathbf{h}) \le f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{h}\rangle + \frac{1}{2}\langle \mathbf{S}\mathbf{h}, \mathbf{h}\rangle, \qquad \mathbf{x}, \mathbf{h} \in \mathbb{R}^d. \tag{43}$$

With $\mathbf{D}_{\mathbf{S}} := \operatorname{diag}(S_{11}, \dots, S_{dd}) \in \mathbb{R}^{d\times d}$, define the normalized matrix $\mathbf{Q} := \mathbf{D}_{\mathbf{S}}^{-\frac{1}{2}} \mathbf{S} \mathbf{D}_{\mathbf{S}}^{-\frac{1}{2}} \in \mathbb{R}^{d\times d}$ and, as in Richtárik & Takáč (2016a, Eqs. (4) and (7)),

$$\sigma := \max\{\langle \mathbf{Q}\mathbf{z}, \mathbf{z}\rangle : \mathbf{z} \in \mathbb{R}^d,\ \langle \mathbf{z}, \mathbf{z}\rangle \le 1\} = \lambda_{\max}(\mathbf{Q}). \tag{44}$$

We denote the largest diagonal entry of **S** by $s_{\max} := \max_{1\le i\le d} S_{ii} = \|\mathbf{D}_{\mathbf{S}}\|$.

Lemma H.4. *Let f be convex and **S**-smooth, with **S**, σ, and $s_{\max}$ as in Definition H.3. Then f satisfies Assumption 3.1, and its smallest cross-coupling constant satisfies*

$$M \le \sigma s_{\max}.$$

Proof. By convexity and (43), f is $\|\mathbf{S}\|$-smooth. Hence,

$$M \overset{\text{Lemma C.1}}{\leq} \|\mathbf{S}\| = \left\|\mathbf{D}_{\mathbf{S}}^{\frac{1}{2}}\mathbf{Q}\mathbf{D}_{\mathbf{S}}^{\frac{1}{2}}\right\| \leq \|\mathbf{D}_{\mathbf{S}}\|\,\|\mathbf{Q}\| \overset{(44)}{=} \sigma s_{\max}. \qquad \square$$

H.3 Sum of block smoothness

Lemma H.5. *Let f be convex, twice continuously differentiable, and L-smooth, and satisfy coordinate-wise smoothness (Assumption 7.1), and define $\bar{L} \coloneqq \frac{1}{n}\sum_{i=1}^n L_i$. Then f satisfies Assumption 3.1, and its smallest cross-coupling constant satisfies*

$$M \leq n\bar{L}.$$

Proof. For a positive semidefinite Hessian, its norm is bounded by the sum of its diagonal-block norms, so coordinate-wise smoothness gives $\sup_{\mathbf{z}\in\mathbb{R}^d}\left\|\nabla^2 f(\mathbf{z})\right\| \leq \sum_{i=1}^n L_i = n\bar{L}$. Taking L to be this global smoothness constant and applying Lemma C.1,

$$L = 0 \implies \nabla^2 f \equiv 0 \implies M = 0, \qquad L > 0 \implies M \overset{Lemma\ C.1}{\leq} L \leq n\bar{L}. \qquad \square$$

H.4 Root-sum-of-squares coupling bound

Lemma H.6. *Let f be differentiable. Suppose that, for each $i \in \{1,\dots,n\}$, there exists $\ell_i \geq 0$ such that $\|\nabla_i f(\mathbf{x}) - \nabla_i f(\mathbf{y})\| \leq \ell_i \|\mathbf{x}-\mathbf{y}\|$ for all $\mathbf{x},\mathbf{y} \in \mathbb{R}^d$.[2] Then f satisfies Assumption 3.1, and its smallest cross-coupling constant satisfies*

$$M \leq \Big(\sum_{i=1}^n \ell_i^2\Big)^{1/2}.$$

Proof. For all $\mathbf{x},\mathbf{h} \in \mathbb{R}^d$,

$$\sum_{i=1}^n \|\nabla_i f(\mathbf{x}+\mathbf{P}_{-i}\mathbf{h}) - \nabla_i f(\mathbf{x})\|^2 \leq \sum_{i=1}^n \ell_i^2 \|\mathbf{P}_{-i}\mathbf{h}\|^2 \leq \left(\sum_{i=1}^n \ell_i^2\right)\|\mathbf{h}\|^2.$$

The claim follows from (3). $\square$

H.5 Baseline complexity comparisons

This appendix derives the complexity bounds in Table 1. All rows use the framework defined in Section 2 on the comparison class of Section 7. Unless stated otherwise, the generic instantiation uses the state $\mathcal{M}_t = \{\mathbf{x}_t\}$ with identity readout and sends $\mathbf{x}_t$ to each active client; one oracle call to (9) returns the anchored value difference and block gradient at a single query point, so one block-gradient step costs one call. Each subsection below specifies only the method-specific active set S_t and local procedure. The local work in round t is $\max_{1\leq i\leq n} K_{t,i}$, and $W(\varepsilon)$ sums this quantity over the $R(\varepsilon)$ communication rounds. Throughout, D_0 is as in Section 4.2. The PiCS bounds assume $M > 0$ and, as in the table, are $\varepsilon \to 0$ asymptotics for fixed problem constants.

H.5.1 Gradient Descent (GD)

Method. GD minimizes f by full-gradient steps with constant step size:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta\,\nabla f(\mathbf{x}_t), \qquad \eta = \frac{1}{L}.$$

Adaptation to our setup. $S_t = \{1,\dots,n\}$. The server broadcasts $I_t^i = \{\mathbf{x}_t\}$ to all clients. In parallel, each client i computes $\mathbf{x}_{t+1}^{(i)} = \mathbf{x}_t^{(i)} - \eta\,\nabla_i f(\mathbf{x}_t)$ via a single oracle call ($K_{t,i} = 1$) and returns $J_t^i = \{\mathbf{x}_{t+1}^{(i)}\}$.

[2] FedBCD uses L_i for these constants (Liu et al., 2020, Assumption A1); they correspond to our ℓ_i.

Complexity. For L-smooth convex f, GD with $\eta = \frac{1}{L}$ achieves $f(\mathbf{x}_T) - f^\star \le \frac{LD_0^2}{2T}$ (Nesterov, 2012, Eq. (2.4)), so the iteration count to reach accuracy ε is

$$T_{\mathrm{GD}}(\varepsilon) \in \mathcal{O}\left(\frac{L\,D_0^2}{\varepsilon}\right).$$

Each iteration is one communication round with one oracle call per client, so

$$R_{\mathrm{GD}}(\varepsilon) = W_{\mathrm{GD}}(\varepsilon) = T_{\mathrm{GD}}(\varepsilon) \in \mathcal{O}\left(\frac{L\,D_0^2}{\varepsilon}\right).$$

H.5.2 Greedy Coordinate Descent (GCD)

Method. GCD updates the single block with the largest block-gradient norm (the Gauss–Southwell rule, Nutini et al., 2015):

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_{i_t}\,\mathbf{U}_{i_t}\nabla_{i_t} f(\mathbf{x}_t), \qquad i_t = \operatorname*{arg\,max}_{1\le i\le n} \|\nabla_i f(\mathbf{x}_t)\|, \quad \eta_{i_t} = \frac{1}{L_{i_t}}.$$

Adaptation to our setup. $S_t = \{1,\dots,n\}$: the selection rule needs all block-gradient norms, so the server broadcasts $I_t^i = \{\mathbf{x}_t\}$ to every client. In parallel, each client i computes $\nabla_i f(\mathbf{x}_t)$ via a single oracle call ($K_{t,i} = 1$) and returns its candidate update together with its norm, $J_t^i = \left\{\mathbf{x}_t^{(i)} - \frac{1}{L_i}\nabla_i f(\mathbf{x}_t),\ \|\nabla_i f(\mathbf{x}_t)\|\right\}$. The server applies the candidate of the winning block i_t and keeps all other blocks unchanged.

Complexity. For smooth convex objectives, the Gauss–Southwell rule with coordinate step sizes has sublinear rate $f(\mathbf{x}_T) - f^\star = \mathcal{O}(\frac{nL_{\max}D_0^2}{T})$ (Nutini et al., 2015, Sec. 2). Thus

$$R_{\mathrm{GCD}}(\varepsilon) \in \mathcal{O}\left(\frac{n\,L_{\max}\,D_0^2}{\varepsilon}\right).$$

Each iteration is one communication round in which all n clients make one oracle call, so $W_{\mathrm{GCD}}(\varepsilon) = R_{\mathrm{GCD}}(\varepsilon)$ and

$$W_{\mathrm{GCD}}(\varepsilon) \in \mathcal{O}\left(\frac{n\,L_{\max}\,D_0^2}{\varepsilon}\right).$$

Remark H.7. The cyclic rule $i_t = (t \bmod n) + 1$ of the Section B example does *not* attain this communication complexity: the best general convex guarantee for cyclic block-gradient updates is $f(\mathbf{x}_k) - f^\star \le \frac{4L(1+n)D_0^2}{k+\frac{8}{n}}$ after k full cycles (Beck & Tetruashvili, 2013, Corollary 3.8), i.e. $\mathcal{O}(\frac{n^2 L\,D_0^2}{\varepsilon})$ communication rounds at one block per communication round, and closing the gap to the randomized or greedy rules is conjectured impossible for general convex objectives (Sun & Hong, 2015).

H.5.3 Randomized Coordinate Descent (RCDM)

Method. RCDM updates a single block per iteration, sampled with importance probabilities $p_i \propto L_i$:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_{i_t}\,\mathbf{U}_{i_t}\nabla_{i_t} f(\mathbf{x}_t), \qquad i_t \sim p, \quad \eta_{i_t} = \frac{1}{L_{i_t}}.$$

Adaptation to our setup. $S_t = \{i_t\}$ with $i_t \sim p$ drawn at the server ($p_i \propto L_i$). The server broadcasts $I_t^{i_t} = \{\mathbf{x}_t\}$ to client i_t only ($I_t^i = \emptyset$ for $i \ne i_t$). Client i_t takes one block-gradient step via a single oracle call ($K_{t,i_t} = 1$) and returns $J_t^{i_t} = \{\mathbf{x}_{t+1}^{(i_t)}\}$; all other clients idle.

Complexity. With importance sampling, RCDM achieves $\mathbb{E}[f(\mathbf{x}_T)] - f^\star \le \frac{2(\sum_{i=1}^n L_i)\,D_0^2}{T+4}$ (Nesterov, 2012, Theorem 1), so

$$T_{\mathrm{RCDM}}(\varepsilon) \in \mathcal{O}\left(\frac{n\,\bar{L}\,D_0^2}{\varepsilon}\right).$$

Each iteration is one communication round with one oracle call by a single active client, so

$$R_{\mathrm{RCDM}}(\varepsilon) = W_{\mathrm{RCDM}}(\varepsilon) = T_{\mathrm{RCDM}}(\varepsilon) \in \mathcal{O}\left(\frac{n\,\bar{L}\,D_0^2}{\varepsilon}\right).$$

H.5.4 Parallel Coordinate Descent (PCDM)

Method. PCDM samples a mini-batch S_t of τ blocks per iteration (τ-nice sampling) and updates them in parallel with stepsizes inflated by an ESO factor $\beta(\tau,\omega)$:

$$\mathbf{x}_{t+1}^{(i)} = \mathbf{x}_t^{(i)} - \eta_i\,\nabla_i f(\mathbf{x}_t) \text{ for } i \in S_t, \qquad \eta_i = \frac{1}{\beta(\tau,\omega)\,L_i}.$$

Adaptation to our setup. We specialize PCDM to $\tau = n$, so $S_t = \{1,\dots,n\}$. The server broadcasts $I_t^i = \{\mathbf{x}_t\}$ to all clients; each client i takes one block-gradient step via a single oracle call ($K_{t,i} = 1$) and returns $J_t^i = \{\mathbf{x}_{t+1}^{(i)}\}$.

Complexity. For τ-nice sampling, $\beta(\tau,\omega) = 1 + \frac{(\tau-1)(\omega-1)}{\max(1,n-1)}$; setting $\tau = n$ gives $\beta = \omega$. The underlying ESO with $\beta = \omega$ requires the partially separable decomposition to have convex pieces (its proof applies Jensen's inequality to each piece); the trivial decomposition whose single piece is f itself shows that every convex f qualifies with $\omega = n$. Theorem 19 of Richtárik & Takáč (2016b) with $w_i = L_i$ is naturally stated in the L-weighted norm and yields, with $D_L^2 := \sum_{i=1}^n L_i \left\|\mathbf{x}_0^{(i)} - (\mathbf{x}^\star)^{(i)}\right\|^2 \le L_{\max} D_0^2$,

$$T_{\mathrm{PCDM}}(\varepsilon) \in \mathcal{O}\left(\frac{\omega\,D_L^2}{\varepsilon}\right).$$

Each iteration is one communication round with one oracle call per client. Thus,

$$R_{\mathrm{PCDM}}(\varepsilon) = W_{\mathrm{PCDM}}(\varepsilon) = T_{\mathrm{PCDM}}(\varepsilon) \in \mathcal{O}\left(\frac{\omega\,D_L^2}{\varepsilon}\right).$$

In the worst case ($\omega = n$), PCDM offers no improvement in communication complexity over serial CD.

H.5.5 Distributed block coordinate descent (DBCD)

We specialize Algorithm 1 of Mareček et al. (2015) to one block per client, all clients active, and zero nonsmooth regularizer. Their generic distributed ESO formula (Theorem 4) gives $\beta = n$ in this setting. For this full-sampling specialization, we replace their generic parameter with the sharper ESO $\beta = \omega$, $w_i = L_i$, from Section H.5.4; this is the choice used in Table 1. The resulting update is exactly the full-sampling PCDM update,

$$\mathbf{x}_{t+1}^{(i)} = \mathbf{x}_t^{(i)} - \frac{1}{\omega L_i}\nabla_i f(\mathbf{x}_t), \qquad i = 1,\dots,n.$$

Each client makes one block-gradient oracle call and returns its updated block, so $K_{t,i} = 1$. The PCDM bound therefore gives $R(\varepsilon) = W(\varepsilon) \in \mathcal{O}(\frac{\omega D_L^2}{\varepsilon})$. Keeping the original generic ESO would instead give $\mathcal{O}(\frac{nD_L^2}{\varepsilon})$.

H.5.6 Hydra

We specialize Algorithm 1 of Richtárik & Takáč (2016a) to zero nonsmooth regularizer and full sampling. Their Table 2 gives $\beta = \sigma$ when all scalar coordinates are updated, so, using Definition H.3,

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{1}{\sigma}\mathbf{D}_{\mathbf{S}}^{-1}\nabla f(\mathbf{x}_t).$$

With full sampling, the update is independent of the partition and also applies to unequal block sizes. Each client evaluates its block gradient once and simultaneously updates all scalar coordinates in its

block. Thus $K_{t,i} = 1$, and each iteration is one communication round. This is GD with respect to the weighted Euclidean norm $\|\mathbf{x}\|_{\mathbf{D_S}} = \sqrt{\langle \mathbf{D_S}\mathbf{x}, \mathbf{x}\rangle}$, in which f is σ-smooth. Its complexity is therefore

$$R(\varepsilon) = W(\varepsilon) \in \mathcal{O}\left(\frac{\sigma\, D^2_{\mathrm{diag}(\mathbf{S})}}{\varepsilon}\right).$$

Here $D_{\mathrm{diag}(\mathbf{S})} = \|\mathbf{x}_0 - \mathbf{x}^\star\|_{\mathbf{D_S}}$ is the initial distance in this norm. The table retains the weighted distance $D_{\mathrm{diag}(\mathbf{S})}$; the comparison in Section 7 uses $D^2_{\mathrm{diag}(\mathbf{S})} \le s_{\max} D_0^2$, analogously to $D_L^2 \le L_{\max} D_0^2$ for PCDM.

H.5.7 PiCS (Ours)

Algorithms 1 and 2 describe PiCS and A-PiCS, and Section 7 gives the local solvers. The PiCS entries in Table 1 combine the communication bounds in Theorems 4.5 and 5.2 with the local-step bounds in Section 7. The work bounds follow from (2); A-PiCS also uses one gradient query per client in its gradient-collection round.

I Additional experiments and reproducibility

We extend the experiments in Section 9 to neural-network training and sparse high-dimensional problems, then examine the line-search parameter in more detail.

Line-search and coupling ratios on LIBSVM. As we use the adaptive line search in Section 4, we track local curvature to help explain its parameter choices. Figure 3 compares fixed-parameter and line-search PiCS on `a1a` and plots $\frac{\lambda_t}{2}$ alongside the local off-diagonal Hessian norm. The right panels report local Hessian-based coupling ratios across 14 datasets. These measurements describe curvature along the sampled trajectories, not the global constant M, and do not by themselves establish tighter worst-case complexity bounds.

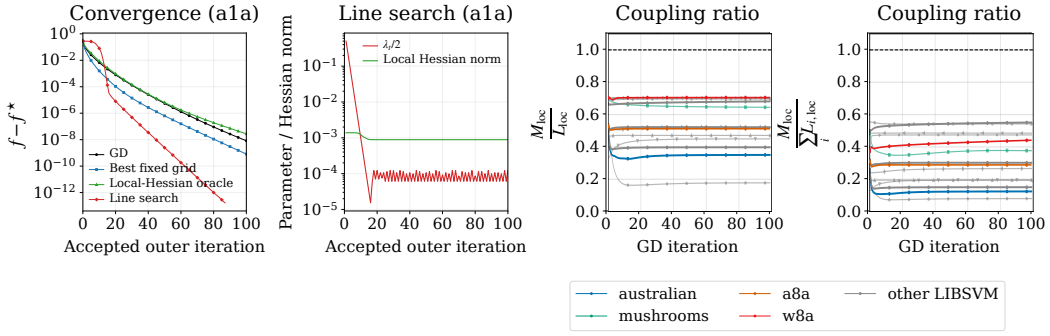


Figure 3: **PiCS line-search and empirical coupling on LIBSVM.** Left: objective gaps and the line-search parameter compared with the local off-diagonal Hessian norm on `a1a`. Right: local Hessian-based coupling ratios across 14 datasets. Here $M_{\mathrm{loc}} = \left\|\mathrm{off}(\nabla^2 f(\mathbf{x}))\right\|_2$; L_{loc} and $L_{i,\mathrm{loc}}$ are the full and block Hessian norms at the same iterate, not global constants.

Sparse high-dimensional LIBSVM: communication and local iterations. We vary the PiCS block count on `url` and `rcv1.binary` datasets and compare against GD with line search. Figure 4 separates communication rounds from local solver iterations and a combined cost model. The last column shows whether communication savings outweigh extra local work. We use an illustrative weight of 10. More blocks do not consistently reduce communication. PiCS uses more local iterations than GD but reaches the target at a lower weighted total cost for every displayed block count.

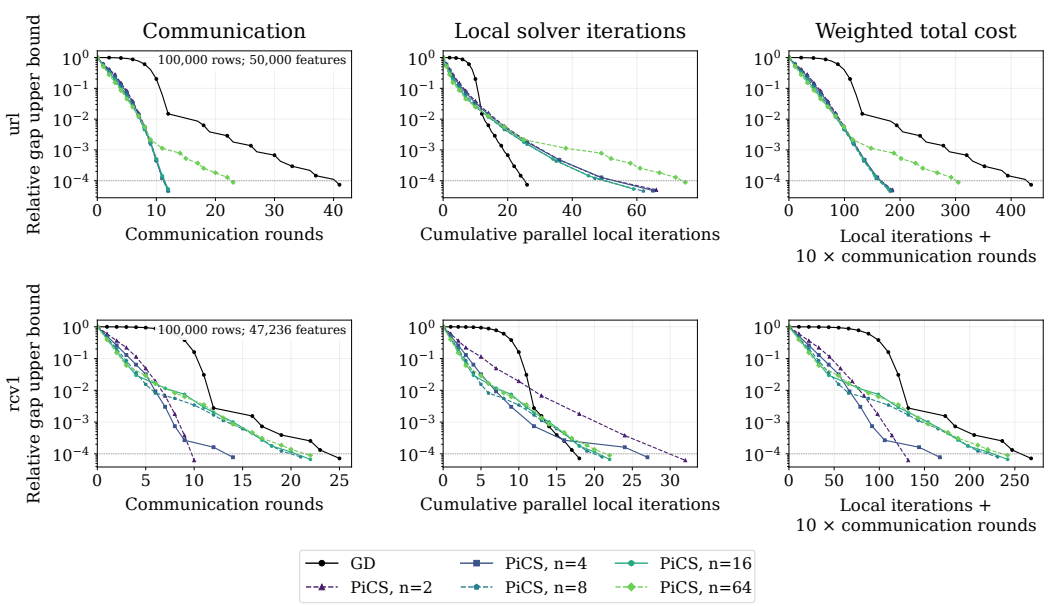


Figure 4: **Line-search PiCS versus GD on sparse LIBSVM problems.** Rows: `url` and `rcv1.binary`. Columns: relative gap upper bound versus communication rounds, cumulative parallel local iterations, and local iterations plus ten times the communication rounds. The weight 10 illustrates relatively costly communication rather than an empirically calibrated cost. PiCS uses $n \in \{2, 4, 8, 16, 64\}$. Counts include rejected trials; local iterations sum the maximum client iteration count per trial. The cost axes do not measure elapsed time or all gradient evaluations.

Local steps in neural-network training. Figure 5 examines how much communication can be saved by taking several local steps between synchronizations when training ResNet34 on CIFAR-10. With $K = 12$ local steps, training reaches the target loss in 31 communication rounds, compared with 63 for $K = 1$; increasing K further gives no consistent improvement. This experiment omits the explicit proximal term, so it illustrates the local-work versus communication trade-off rather than testing the PiCS convergence guarantees.

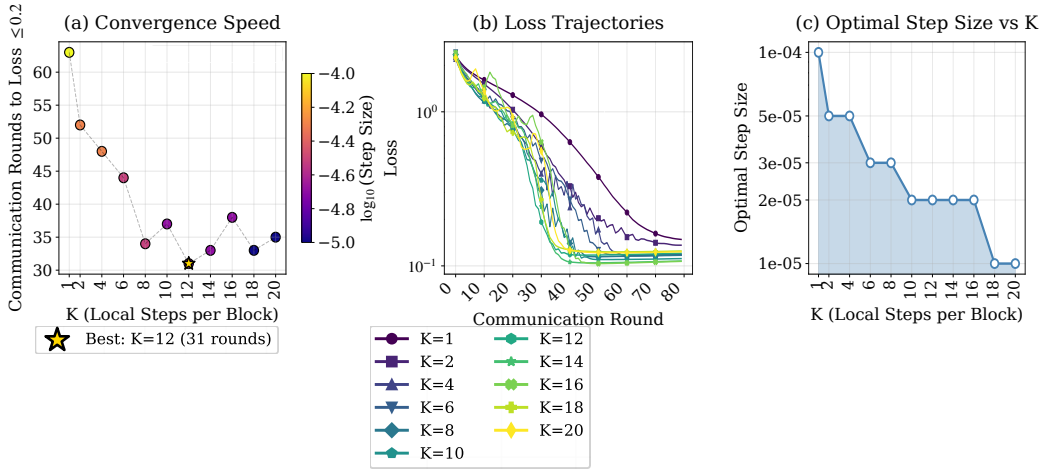


Figure 5: **Local steps in ResNet34 training on CIFAR-10. (a)** Communication rounds to training loss 0.2 versus local steps K; color indicates the selected step size and the star marks the best plotted configuration. **(b)** Training-loss trajectories. **(c)** Selected step size for each K.

What the line search selects. Figures 6 to 8 examine convergence, parameter trajectories, and sensitivity to the number of blocks using the acceptance condition in Section 4. The selected $\frac{\lambda_t}{2}$ can remain below the local off-diagonal Hessian norm; it should not be read as an estimate of the global M.

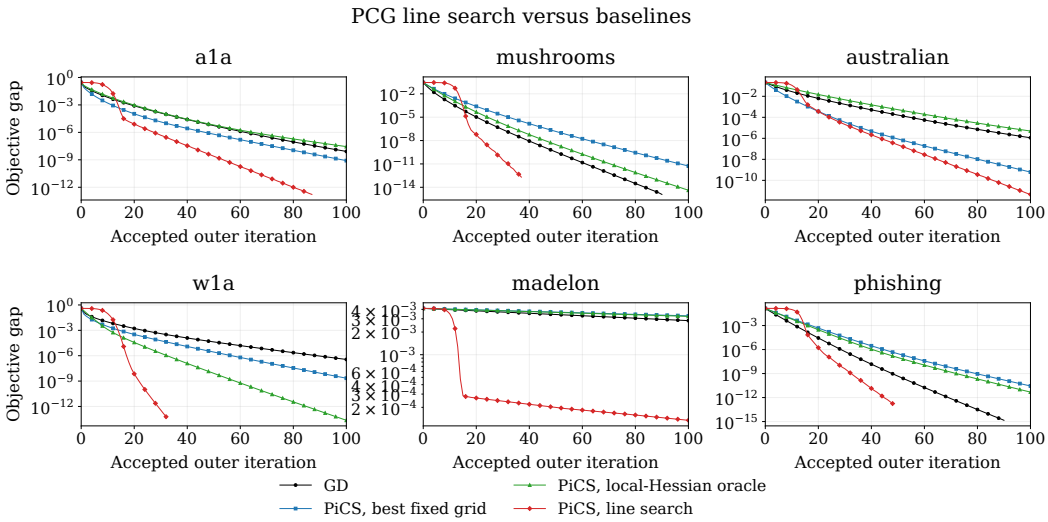


Figure 6: **Convergence with fixed, oracle, and line-search proximal parameters.** Objective gap versus accepted outer iterations on six LIBSVM datasets. Black circles: GD; blue squares: the fixed parameter with the smallest final objective on the tested grid; green triangles: local-Hessian oracle; red diamonds: line search. Gaps below the reference uncertainty are omitted.

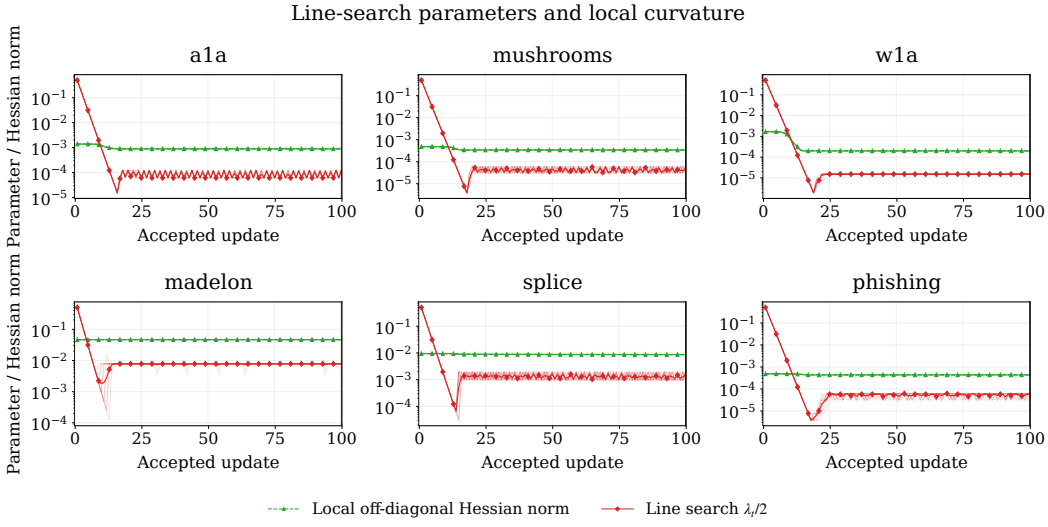


Figure 7: **Line-search parameters and local curvature along the iterates.** Red diamonds show $\frac{\lambda_t}{2}$; green triangles show $\|\mathrm{off}(\mathbf{H}(\mathbf{x}_t))\|_2$. Curves report geometric means over random initializations.

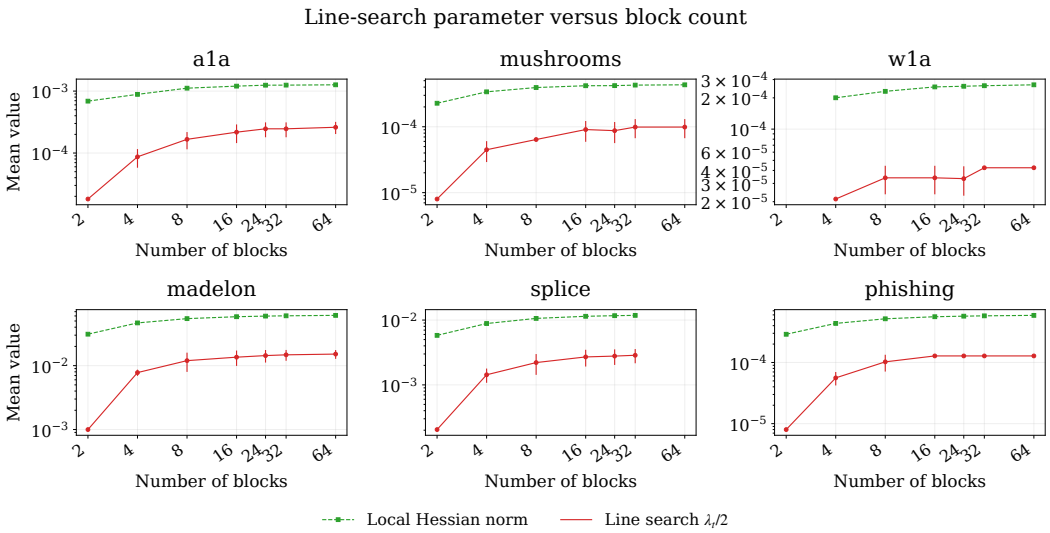


Figure 8: **Dependence on the number of blocks.** Red circles show the final-half mean of $\frac{\lambda_t}{2}$ with one-standard-deviation bars; green squares show the iteration-averaged local off-diagonal Hessian norm. Both axes use logarithmic scales.