Exact Decentralized Optimization via Explicit ℓ_1 Consensus Penalties

Hong Wang*

November 28, 2025

Abstract

Consensus optimization enables autonomous agents to solve joint tasks through peer-to-peer exchanges alone. Classical decentralized gradient descent is appealing for its minimal state but fails to achieve exact consensus with fixed stepsizes unless additional trackers or dual variables are introduced. We revisit penalty methods and introduce a decentralized two-layer framework that couples an outer penalty-continuation loop with an inner plug-and-play saddle-point solver. Any primal-dual routine that satisfies simple stationarity and communication conditions can be used; when instantiated with a proximal-gradient solver, the framework yields the DP^{2}G algorithm, which reaches exact consensus with constant stepsizes, stores only one dual residual per agent, and requires exactly two short message exchanges per inner iteration. An explicit ℓ_1 penalty enforces agreement and, once above a computable threshold, makes the penalized and constrained problems equivalent. Leveraging the Kurdyka-Łojasiewicz property, we prove global convergence, vanishing disagreement, and linear rates for strongly convex objectives under any admissible inner solver. Experiments on distributed least squares, logistic regression, and elastic-net tasks across various networks demonstrate that DP^{2}G outperforms DGD-type methods in both convergence speed and communication efficiency, is competitive with gradient-tracking approaches while using less memory, and naturally accommodates composite objectives.

1 Introduction

Consensus optimization is central to modern networked systems including distributed sensing, distributed learning, smart grids, and cooperative robotics. In these settings, n autonomous agents aim to solve

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

using only local computation and peer-to-peer communication. Practical deployments must satisfy three core requirements: minimal per-agent memory, low communication overhead, and robust convergence under realistic networking constraints [6, 10, 11, 12]. Failing to meet these constraints increases implementation costs and undermines system resilience.

Classical decentralized gradient descent (DGD) meets low-memory requirements but achieves exact consensus only with vanishing stepsizes [6, 10]. Such diminishing stepsizes degrade practical performance and complicate parameter tuning, while fixed stepsizes result in persistent

[*] School of Artificial Intelligence, Shenzhen Technology University, 3002 Lantian Road, Pingshan District, Shenzhen Guangdong, China, 518118. Email: hitwanghong@163.com

disagreement. Gradient-tracking and primal-dual schemes [9, 8, 13] recover exactness with constant steps by equipping each node with auxiliary states (gradient trackers, dual multipliers, or both), thereby doubling memory usage and increasing communication overhead. These drawbacks motivate algorithms that preserve the simplicity of DGD while delivering the accuracy of more advanced methods.

Penalty methods provide that bridge. Quadratic penalties and augmented Lagrangians are analytically convenient but require large penalty weights and full dual updates, both of which erode efficiency in bandwidth-limited settings. Explicit ℓ_1 penalties instead enjoy an exact-penalty property: once the coefficient exceeds a computable threshold, the penalized and constrained formulations share the same solutions. The main technical challenges arise from managing the resulting nonseparable disagreement term and coordinating the penalty schedule without centralized control, particularly when aiming for convergence guarantees beyond convex objectives.

We address these challenges using a modular two-layer framework. The outer layer runs a fully decentralized penalty-continuation scheme that keeps each agent's state to one primal vector and one dual residual, matching the memory footprint of DGD while steering the penalty toward the exactness threshold. The inner layer is a plug-and-play saddle-point solver that must guarantee a verifiable decrease in a composite optimality residual; our theory is independent of the specific routine provided it meets simple stationarity, communication, and warm-start requirements. Instantiating the inner layer with a primal-dual proximal-gradient method yields the DP^{2}G algorithm evaluated in our experiments, yet the same interface accommodates Chambolle–Pock variants, accelerated saddle-point updates, or inexact ADMM iterations. Once the penalty saturates, the framework attains the accuracy of constrained consensus methods without sacrificing the simplicity of DGD.

Contributions. The paper makes three main contributions.

- **Exact-penalty perspective.** We recast consensus optimization by penalizing disagreement through the operator $Z = (I - W) \otimes I_m$ with an explicit ℓ_1 term and, using Hoffman's bound, derive a computable threshold beyond which the penalized and constrained formulations are equivalent.

- **Modular two-layer framework.** We formalize the penalty-continuation outer loop and the plug-and-play inner saddle solver, specifying the interface in terms of residual reduction, communication accounting, and warm-start requirements. When the inner routine is a primal-dual proximal-gradient method, we obtain DP^{2}G, which stores a single primal and dual vector per agent and needs only two neighbor exchanges per inner iteration.

- **Global guarantees and evidence.** Leveraging the Kurdyka-Łojasiewicz (KŁ) property of the penalized objective, we adapt the Lyapunov analysis of preconditioned primal-dual gradient methods [5] to the distributed setting and prove global convergence to consensual critical points, with linear rates under strong convexity for *any* admissible inner solver. The same analysis covers semi-algebraic nonconvex objectives, and our experiments on least squares, logistic regression, and elastic-net tasks across several topologies corroborate the theoretical predictions.

1.1 Related Work and Recent Advances

Gradient tracking. EXTRA [9] pioneered gradient-correction steps that cancel consensus bias and allow fixed stepsizes. Subsequent work generalized the idea to directed graphs (Push-DIGing [8]), introduced Nesterov-type acceleration (NIDS [13]), and analyzed unbalanced graphs or nonconvex objectives [14, 15]. Recent efforts pursue optimal gradient complexity [28], modular frameworks that mix local computation with tracking [31, 32], and variance-reduced stochastic variants for large-scale learning [16, 17, 30, 33]. All such methods keep auxiliary gradient trackers in memory and in flight, which doubles the storage relative to DGD and motivates our search for tracker-free yet exact algorithms.

ADMM and primal-dual algorithms. ADMM handles consensus by introducing dual variables and augmented penalties [4]. It is robust but often slower than first-order trackers on smooth problems [9], motivating communication-efficient refinements based on compression, event-triggered communication, or adaptive penalty updates [18, 19, 20]. Linear rates are available under strong convexity [21, 22], albeit at the cost of storing both primal and dual vectors per agent. Our framework borrows the primal-dual perspective but decouples the penalty schedule (outer layer) from the particular saddle solver (inner layer), allowing lighter per-node states when DP^{2}G is selected.

Penalty methods. Classical quadratic penalties and augmented Lagrangians can degrade conditioning or require full dual updates, whereas nonquadratic penalties such as ℓ_1 and elastic-net terms offer robustness to noise and can encourage sparse disagreement corrections [23, 24, 25]. Recent studies leveraged penalties for constrained consensus [27] or bilevel formulations [29], yet most rely on diminishing penalty sequences or heuristics without exact-penalty guarantees. Our contribution is to couple explicit ℓ_1 continuation with a plug-and-play saddle solver, yielding fixed-stepsize guarantees while keeping the memory footprint comparable to DGD.

1.2 Notation

Bold lowercase letters (e.g., $\mathbf{x}_i$) denote vectors and uppercase letters (e.g., W) denote matrices. The operator $\mathrm{col}(\cdot)$ stacks its arguments column-wise; $\mathbf{1}$ and $\mathbf{0}$ are all-ones and all-zeros vectors of compatible dimensions. The Kronecker product is written $\otimes$, and $\mathbb{R}^m$ denotes the m-dimensional Euclidean space equipped with norm $\|\cdot\|$. We reserve $\|\cdot\|_1$ for the ℓ_1 norm. For a closed set $\mathcal{X}$, $\mathrm{dist}(x, \mathcal{X})$ is the Euclidean distance from x to $\mathcal{X}$, and $\mathrm{Proj}_{\mathcal{X}}(x)$ is the Euclidean projection. The limiting subdifferential of a function g at x is $\partial g(x)$. Agent indices lie in $[n] = \{1, 2, \ldots, n\}$, the communication graph is $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and $\mathcal{N}_i$ denotes the neighbor set of agent i.

1.3 Organization

The remaining content is organized as follows. Section 2 formalizes the consensus model and the explicit ℓ_1 penalty reformulation. Section 3 details the modular two-layer framework together with its DP^{2}G instantiation. Section 4 establishes convergence guarantees, and Section 5 reports the numerical results. Section 6 summarizes the main findings and outlines future directions.

2 Consensus Model and Penalty Reformulation

2.1 Network Topology

We adopt the standard modeling framework used in decentralized optimization. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected undirected graph with node set $\mathcal{V} = [n]$ and edge set $\mathcal{E}$. Agent i exchanges information only with neighbors in $\mathcal{N}_i = \{j : (i,j) \in \mathcal{E}\}$. Communication is governed by a mixing matrix W that respects the sparsity of $\mathcal{G}$ and satisfies the conditions in Assumption 1.

Assumption 1. *The mixing matrix $W = [w_{ij}] \in \mathbb{R}^{n \times n}$ satisfies the following conditions:*

1. *For any $i \neq j$, if $(i,j) \notin \mathcal{E}$, then $w_{ij} = 0$, and $w_{ij} > 0$ otherwise.*
2. *W is symmetric, namely, $W = W^\mathsf{T}$.*
3. *W is doubly stochastic, i.e., $W\mathbf{1} = \mathbf{1}$ and $\mathbf{1}^\mathsf{T} W = \mathbf{1}^\mathsf{T}$.*
4. *$-I \prec W \preceq I$.*

Denote the eigenvalues of W by $\lambda_1(W) \geq \cdots \geq \lambda_n(W)$. Assumption 1 guarantees $-1 < \lambda_n(W) \leq \cdots \leq \lambda_2(W) < \lambda_1(W) = 1$. The spectral gap $1-\zeta$, with $\zeta = \max\{|\lambda_2(W)|, |\lambda_n(W)|\}$, is therefore strictly positive and measures how quickly consensus information diffuses through the network.

2.2 Consensus Optimization Problem

Each agent privately holds a differentiable function $f_i : \mathbb{R}^m \to \mathbb{R}$. The consensus optimization problem seeks

$$\min_{x \in \mathbb{R}^m} \; f(x) = \sum_{i=1}^{n} f_i(x), \tag{2.1}$$

where the factor $1/n$ is omitted because it does not affect optimal solutions.

Introducing local copies $\mathbf{x}_i \in \mathbb{R}^m$ and stacking $\mathbf{x} = \operatorname{col}(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ yields the equivalent constrained formulation

$$\min_{\mathbf{x} \in \mathbb{R}^{nm}} \; F(\mathbf{x}) = \sum_{i=1}^{n} f_i(\mathbf{x}_i) \quad \text{s.t.} \quad \mathbf{x}_1 = \cdots = \mathbf{x}_n. \tag{2.2}$$

Let $Z = (I_n - W) \otimes I_m$. Because W is symmetric and doubly stochastic, $I_n - W$ is symmetric and positive semidefinite with nullspace spanned by $\mathbf{1}$. Consequently,

$$Z^\mathsf{T} = Z, \qquad Z(\mathbf{1} \otimes x) = \big((I_n - W)\mathbf{1}\big) \otimes x = \mathbf{0} \quad \forall x \in \mathbb{R}^m.$$

Since $I_n - W \succeq 0$, it follows that $Z = (I_n - W) \otimes I_m \succeq 0$, and its nullspace is $\{\mathbf{1} \otimes v : v \in \mathbb{R}^m\}$, corresponding exactly to the consensus subspace where all agents agree.

Assumption 2. *For each $i \in [n]$, the function f_i is proper, closed, bounded below, and continuously differentiable with L_i-Lipschitz continuous gradient. Define $L_{\max} = \max_i\{L_i\}$.*

2.3 The ℓ_1 Consensus Penalty

We study the penalized objective

$$\Phi_\rho(\mathbf{x}) = F(\mathbf{x}) + \rho \left\| Z\mathbf{x} \right\|_1 , \tag{2.3}$$

where $\rho > 0$ is the penalty parameter and $Z = (I - W) \otimes I_m$. Denoting $\mathbf{u} = Z\mathbf{x}$ and

$$\mathbf{u}_i := (Z\mathbf{x})_i = (1 - w_{ii})\mathbf{x}_i - \sum_{j \in \mathcal{N}_i} w_{ij}\mathbf{x}_j, \tag{2.4}$$

we have $\|Z\mathbf{x}\|_1 = \sum_{i=1}^n \|\mathbf{u}_i\|_1$. Importantly, each component of $\mathbf{u}_i$ depends on both $\mathbf{x}_i$ and the neighbor variables $\mathbf{x}_j$, so the penalty is *not* separable across agents. Any update that manipulates a single block $\mathbf{x}_i$ must account for the contribution of $\mathbf{x}_i$ to $\mathbf{u}_i$ and to the rows $\mathbf{u}_r$ of its in-neighbors.

Lemma 2.1 (Local structure of the penalty subgradient)**.** *Let $g_i(\mathbf{x}) \in (\partial_{\mathbf{x}} \|Z\mathbf{x}\|_1)_i$ be a subgradient of the penalty with respect to agent i. Then*

$$g_i(\mathbf{x}) = (1 - w_{ii}) \operatorname{sign}(\mathbf{u}_i) - \sum_{r : i \in \mathcal{N}_r} w_{ri} \operatorname{sign}(\mathbf{u}_r), \tag{2.5}$$

where the $\operatorname{sign}(\cdot)$ *operator acts component-wise and produces subgradients in* $[-1, 1]$ *when the corresponding residual coordinate is zero.*

Proof. Recall that $\|Z\mathbf{x}\|_1 = \sum_{r=1}^n \|\mathbf{u}_r\|_1$ with $\mathbf{u}_r = (1 - w_{rr})\mathbf{x}_r - \sum_{j \in \mathcal{N}_r} w_{rj}\mathbf{x}_j$. Fix i and perturb only the block $\mathbf{x}_i$ by $h \in \mathbb{R}^m$. The resulting change in the penalty is

$$\begin{aligned} \|Z(\mathbf{x} + e_i \otimes h)\|_1 - \|Z\mathbf{x}\|_1 = {} & \Big\| \mathbf{u}_i + (1 - w_{ii})h \Big\|_1 - \|\mathbf{u}_i\|_1 \\ & + \sum_{r : i \in \mathcal{N}_r} \Big(\big\| \mathbf{u}_r - w_{ri}h \big\|_1 - \|\mathbf{u}_r\|_1 \Big), \end{aligned}$$

where e_i is the ith unit vector in $\mathbb{R}^n$. Each term in the sum is convex in h and admits the subgradient representation

$$\partial_h \|\mathbf{u}_i + (1 - w_{ii})h\|_1 = (1 - w_{ii}) \operatorname{sign}(\mathbf{u}_i), \qquad \partial_h \|\mathbf{u}_r - w_{ri}h\|_1 = -w_{ri} \operatorname{sign}(\mathbf{u}_r),$$

with the convention that $\operatorname{sign}(0)$ is the interval $[-1, 1]$ applied component-wise. Summing these contributions yields precisely (2.5). Because the limiting subdifferential of a finite sum of convex functions is the sum of the individual limiting subdifferentials, the stated expression describes every element of $(\partial_{\mathbf{x}} \|Z\mathbf{x}\|_1)_i$. □

Assumption 3 (Level-boundedness)**.** *The objective function of* (2.2) *F has bounded level sets. Equivalently, the penalized objective function Φ_ρ has bounded level sets.*

Theorem 2.2 (Exactness of ℓ_1 penalty)**.** *Suppose Assumption 2 holds and* (2.1) *has a nonempty solution set $\mathcal{X}^*$. Then there exists $\bar{\rho} > 0$ such that for any $\rho \geq \bar{\rho}$:*

(a) Every consensual optimal point $\mathbf{x}^* = \mathbf{1} \otimes x^*$ with $x^* \in \mathcal{X}^*$ is a local minimizer of Φ_ρ.

(b) If $\mathbf{x}^\dagger$ is a local minimizer of Φ_ρ, then $Z\mathbf{x}^\dagger = \mathbf{0}$ and its consensus component $x^\dagger$ solves (2.1).

Proof. Let $\mathcal{C} = \mathrm{null}(Z) = \{\mathbf{1} \otimes x : x \in \mathbb{R}^m\}$ and fix any consensual solution $x^* \in \mathcal{X}^*$. Set $\mathbf{x}^* = \mathbf{1} \otimes x^*$, so that $\mathbf{x}^* \in \mathcal{C}$ and $Z\mathbf{x}^* = \mathbf{0}$. Because each f_i is continuously differentiable, F is differentiable and therefore locally Lipschitz. Let $L_F > 0$ denote a Lipschitz constant of F on a compact neighborhood $\mathcal{V}$ of $\mathbf{x}^*$ whose existence follows from continuity.

Hoffman's bound for the linear system $Z\mathbf{x} = 0$ [26] yields a constant $C > 0$ such that

$$\mathrm{dist}(\mathbf{x}, \mathcal{C}) \le C\,\|Z\mathbf{x}\|_1 \qquad \forall\, \mathbf{x} \in \mathbb{R}^{nm}. \tag{2.6}$$

Part (a). For any $\mathbf{x} \in \mathcal{V}$ we use the triangle inequality and the Lipschitz property to write $F(\mathbf{x}) \ge F(\mathbf{x}^*) - L_F\,\mathrm{dist}(\mathbf{x}, \mathcal{C})$. Combining this inequality with (2.6) we obtain

$$\Phi_\rho(\mathbf{x}) = F(\mathbf{x}) + \rho\|Z\mathbf{x}\|_1 \ge F(\mathbf{x}^*) + (\rho - CL_F)\|Z\mathbf{x}\|_1.$$

Whenever $\rho \ge CL_F$, the right-hand side is minimized at $\|Z\mathbf{x}\|_1 = 0$, i.e., at $\mathbf{x} \in \mathcal{C}$. Because $\mathbf{x}^*$ belongs to $\mathcal{C}$ and is optimal for (2.2), we conclude that $\mathbf{x}^*$ is a local minimizer of Φ_ρ for every $\rho \ge CL_F$. The constant $\bar{\rho} = CL_F$ therefore satisfies statement (a).

Part (b). Let $\mathbf{x}^\dagger$ be any local minimizer of Φ_ρ with $\rho > \bar{\rho}$. Pick a compact neighborhood $\mathcal{W}$ of $\mathbf{x}^\dagger$ on which F is Lipschitz with constant $L_{\mathcal{W}}$. Take $\hat{\mathbf{x}}$ to be the Euclidean projection of $\mathbf{x}^\dagger$ onto $\mathcal{C}$. Then $Z\hat{\mathbf{x}} = \mathbf{0}$ and $\hat{\mathbf{x}} \in \mathcal{W}$ for $\mathcal{W}$ small enough. Hence

$$F(\hat{\mathbf{x}}) \le F(\mathbf{x}^\dagger) + L_{\mathcal{W}}\,\|\hat{\mathbf{x}} - \mathbf{x}^\dagger\| = F(\mathbf{x}^\dagger) + L_{\mathcal{W}}\,\mathrm{dist}(\mathbf{x}^\dagger, \mathcal{C}) \tag{2.7}$$

$$\le F(\mathbf{x}^\dagger) + L_{\mathcal{W}} C\|Z\mathbf{x}^\dagger\|_1, \tag{2.8}$$

where the last step uses (2.6). Optimality of $\mathbf{x}^\dagger$ on $\mathcal{W}$ yields

$$F(\mathbf{x}^\dagger) + \rho\|Z\mathbf{x}^\dagger\|_1 \le F(\hat{\mathbf{x}}) + \rho\|Z\hat{\mathbf{x}}\|_1 = F(\hat{\mathbf{x}}). \tag{2.9}$$

Substituting (2.8) into (2.9) gives $(\rho - CL_{\mathcal{W}})\|Z\mathbf{x}^\dagger\|_1 \le 0$. As $\rho > \bar{\rho} \ge CL_{\mathcal{W}}$, the only possibility is $Z\mathbf{x}^\dagger = \mathbf{0}$, showing that every local minimizer is consensual. Finally, when $\mathbf{x}^\dagger \in \mathcal{C}$ the penalty term vanishes and $F(\mathbf{x}^\dagger) \le F(\hat{\mathbf{x}})$ for every $\hat{\mathbf{x}} \in \mathcal{C}$ close to $\mathbf{x}^\dagger$, implying that the shared component solves (2.1). □

3 Decentralized Primal-Dual Proximal Gradient Algorithm

We now derive the DP^{2}G update and formalize the resulting two-layer architecture. The outer layer adapts the penalty parameter ρ_k at iteration k, whereas the inner layer performs primal-dual proximal-gradient steps for a fixed penalty until a verifiable optimality condition is satisfied.

3.1 Primal-dual splitting

Introduce the saddle formulation of the penalized problem

$$\min_{\mathbf{x} \in \mathbb{R}^{nm}} \max_{\mathbf{y} \in \mathcal{Y}_\rho} \mathcal{L}(\mathbf{x}, \mathbf{y}; \rho) = F(\mathbf{x}) + \langle Z\mathbf{x}, \mathbf{y}\rangle - \delta_{\mathcal{Y}_\rho}(\mathbf{y}), \tag{3.1}$$

where $\mathcal{Y}_\rho = \{\mathbf{y} \in \mathbb{R}^{nm} : \|\mathbf{y}\|_\infty \le \rho\}$ and $\delta_{\mathcal{Y}_\rho}$ is the indicator of the hypercube. Eliminating $\mathbf{y}$ recovers $\Phi_\rho(\mathbf{x})$ because the conjugate of $\delta_{\mathcal{Y}_\rho}$ is the norm $\rho\|\cdot\|_1$. The gradient of F is block-separable, while the adjoint $\mathbf{v} := Z^\top\mathbf{y}$ inherits the sparsity of the mixing matrix:

$$\mathbf{v}_i := (Z^\top\mathbf{y})_i = (1 - w_{ii})\mathbf{y}_i - \sum_{j \in \mathcal{N}_i} w_{ji}\mathbf{y}_j. \tag{3.2}$$

Hence each agent only needs the dual variables of its neighbors.

Applying the primal-dual hybrid gradient scheme with the recommended extrapolation parameter $\theta = 1$ yields

$$\mathbf{y}^{t+1} = \mathrm{Proj}_{\mathcal{Y}_{\rho_k}}\big(\mathbf{y}^t + \sigma Z \bar{\mathbf{x}}^t\big), \tag{3.3a}$$

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \alpha\big(\nabla F(\mathbf{x}^t) + Z^\top \mathbf{y}^{t+1}\big), \tag{3.3b}$$

$$\bar{\mathbf{x}}^{t+1} = \mathbf{x}^{t+1} + \big(\mathbf{x}^{t+1} - \mathbf{x}^t\big), \tag{3.3c}$$

where $\bar{\mathbf{x}}^0 = \mathbf{x}^0$. The projection $\mathrm{Proj}_{\mathcal{Y}_{\rho_k}}$ acts as a soft threshold on the disagreement residual: whenever a component of $Z\bar{\mathbf{x}}^t$ exceeds ρ_k, the corresponding dual variable saturates at $\pm\rho_k$ and memorizes the sign of the disagreement. The fixed steps obey $0 < \alpha < 1/L_{\max}$ and $0 < \sigma < 1/(\alpha(1 - \lambda_n(W))^2)$.

The local form required by agent i becomes

$$\mathbf{y}_i^{t+1} = \mathrm{Proj}_{[-\rho_k,\rho_k]}\Big(\mathbf{y}_i^t + \sigma \bar{\mathbf{u}}_i^t\Big), \tag{3.4a}$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t - \alpha\big(\nabla f_i(\mathbf{x}_i^t) + \mathbf{v}_i^{t+1}\big), \tag{3.4b}$$

$$\bar{\mathbf{x}}_i^{t+1} = \mathbf{x}_i^{t+1} + \big(\mathbf{x}_i^{t+1} - \mathbf{x}_i^t\big), \tag{3.4c}$$

where $\bar{\mathbf{u}}_i^t = (Z\bar{\mathbf{x}}^t)_i$ is computed from extrapolated neighbor information. Each inner iteration therefore still requires two neighbor exchanges: one to form $\bar{\mathbf{u}}_i^t$ and one to share the fresh dual variables $\mathbf{y}_i^{t+1}$ before evaluating $\mathbf{v}_i^{t+1} = (Z^\top \mathbf{y}^{t+1})_i$.

3.2 Two-layer adaptive architecture

The outer layer increases the penalty parameter until the exactness threshold from Theorem 2.2 is exceeded, while the inner layer executes (3.4) until a prescribed optimality tolerance is met. Communication-wise, DP^{2}G needs two neighbor exchanges per inner iteration (one for $\mathbf{x}$, one for $\mathbf{y}$) and only stores $\mathbf{x}_i$ and $\mathbf{y}_i$ locally. The full procedure is summarized in Algorithm 1.

Assumption 4 (Penalty schedule and stepsize)**.** *The penalty sequence satisfies* $\rho_0 > 0$*,* $\rho_{\max} < \infty$*, and* $\rho_{k+1} = \min\{\beta\rho_k, \rho_{\max}\}$ *with* $\beta > 1$*. The cap* $\rho_{\max}$ *is chosen so that* $\rho_{\max} \geq \bar{\rho}$ *(the exactness threshold from Theorem 2.2) and* $\rho_{\max} > G$*, where* G *is any known upper bound on* $\max_{i\in[n]} \|\nabla f_i(x)\|$ *over the sublevel set* $\{\mathbf{x} : \Phi_{\rho_0}(\mathbf{x}) \leq \Phi_{\rho_0}(\mathbf{x}^0)\}$ *guaranteed by Assumption 3. The fixed stepsizes obey* $0 < \alpha < 1/(3L_{\max})$ *and* $0 < \sigma < 1/(\alpha(1 - \lambda_n(W))^2)$*.*

Assumption 5 (Tolerance sequences)**.** *The stationarity tolerances* $\{\varepsilon_k\}$ *and consensus tolerances* $\{\delta_k\}$ *satisfy* $\varepsilon_k > 0$ *and* $\delta_k > 0$*, where both sequences are nonincreasing and satisfy* $\varepsilon_k \to 0$ *and* $\delta_k \to 0$ *as* $k \to \infty$*. Typical choices include* $\varepsilon_k = \varepsilon_0/k^{\eta_1}$ *and* $\delta_k = \delta_0/k^{\eta_2}$ *for* $\eta_1 > 0$ *and* $\eta_2 > 0$*, or exponentially decaying rules such as* $\varepsilon_k = \varepsilon_0\theta_1^k$ *and* $\delta_k = \delta_0\theta_2^k$ *with* $\theta_1, \theta_2 \in (0, 1)$*. In practice, choosing* δ_k *to decay faster than* ε_k *(e.g.,* $\eta_2 = 2\eta_1$*) ensures consensus is achieved before final stationarity.*

The stepsize bounds $\alpha < 1/(3L_{\max})$ and $\sigma < 1/(\alpha\kappa_Z^2)$ with $\kappa_Z = \|Z\|_2 = 1 - \lambda_n(W)$ are stricter than those for proximal or gradient descent with Lipschitz gradients. These restrictions ensure sufficient descent in the Lyapunov analysis (see Lemma 4.2). The inner loop performs proximal gradient steps until the optimality condition is satisfied, while the growth factor $\beta > 1$ adaptively strengthens the penalty when progress slows.

Algorithm 1 Two-layer Decentralized Primal-Dual Proximal Gradient (DP^{2}G)

```
Input: Initial $\mathbf{x}_i^0$, duals $\mathbf{y}_i^0 = \mathbf{0}$, penalty $\rho_0 > 0$, primal step $\alpha > 0$, dual step $\sigma > 0$, growth factor $\beta > 1$, cap $\rho_{\max}$, tolerances $\{\varepsilon_k\}$ and $\{\delta_k\}$.
Set $k = 0$ and $\rho_k = \rho_0$.
while not terminated do
    Set $t = 0$ and $\bar{\mathbf{x}}_i^0 = \mathbf{x}_i^k$.
    repeat                                   ▷ Inner primal-dual iterations with fixed $\rho_k$
        Exchange $\bar{\mathbf{x}}_i^t$ with neighbors and compute $\bar{\mathbf{u}}_i^t$ according to (2.4).
        Dual update: $\mathbf{y}_i^{t+1} = \mathrm{Proj}_{[-\rho_k,\rho_k]}(\mathbf{y}_i^t + \sigma\bar{\mathbf{u}}_i^t)$.
        Exchange $\mathbf{y}_i^{t+1}$ with neighbors and compute $\mathbf{v}_i^{t+1}$ according to (3.2).
        Primal update: $\mathbf{x}_i^{t+1} = \mathbf{x}_i^t - \alpha(\nabla f_i(\mathbf{x}_i^t) + \mathbf{v}_i^{t+1})$.
        Extrapolation: $\bar{\mathbf{x}}_i^{t+1} = \mathbf{x}_i^{t+1} + (\mathbf{x}_i^{t+1} - \mathbf{x}_i^t)$.
        $t \leftarrow t + 1$.
    until $\left\|\nabla f_i(\mathbf{x}_i^t) + \mathbf{v}_i^t\right\| \le \varepsilon_k$ for all $i$.
    Set $\mathbf{x}_i^{k+1} = \mathbf{x}_i^t$, $\mathbf{y}_i^{k+1} = \mathbf{y}_i^t$.
    Exchange the fresh $\mathbf{x}_i^{k+1}$ and compute $d_i^{k+1} = \left\|\mathbf{u}_i^{k+1}\right\|_1$.
    if $d_i^{k+1} \le \delta_k$ for all $i$ then
        break
    else
        Update penalty: $\rho_{k+1} = \min\{\beta\rho_k, \rho_{\max}\}$, and $k \leftarrow k + 1$.
    end if
end while
```

3.3 Stationarity certificates

The inner stopping criterion relies on the distance of the composite subgradient to zero. Because the dual variables belong to $\mathcal{Y}_{\rho_k}$, the penalty subgradient is readily available through $Z^{\mathsf{T}}\mathbf{y}$.

Lemma 3.1 (Stationarity residual)**.** *Let $(\mathbf{x}, \mathbf{y})$ satisfy $\mathbf{y} \in \rho_k \partial\|Z\mathbf{x}\|_1$. Then*

$$\operatorname{dist}\big(\mathbf{0}, \nabla F(\mathbf{x}) + \partial(\rho_k\|Z\cdot\|_1)(\mathbf{x})\big) = \left\|\nabla F(\mathbf{x}) + Z^{\mathsf{T}}\mathbf{y}\right\|. \tag{3.5}$$

For a general $\mathbf{y} \in \mathcal{Y}_{\rho_k}$ the right-hand side provides a computable upper bound,

$$\operatorname{dist}\big(\mathbf{0}, \nabla F(\mathbf{x}) + \partial(\rho_k\|Z\cdot\|_1)(\mathbf{x})\big) \le \left\|\nabla F(\mathbf{x}) + Z^{\mathsf{T}}\mathbf{y}\right\|.$$

The inequality is strict whenever $\mathbf{y}$ lies in the interior of $\mathcal{Y}_{\rho_k}$, because then $Z^{\mathsf{T}}\mathbf{y}$ does not belong to $\partial(\rho_k\|Z\cdot\|_1)(\mathbf{x})$.

Proof. Because the conjugate of $\delta_{\mathcal{Y}_{\rho_k}}$ is $\rho_k\|\cdot\|_1$, the subdifferential of the penalty reads

$$\partial(\rho_k\|Z\cdot\|_1)(\mathbf{x}) = Z^{\mathsf{T}}\big(\rho_k\partial\|Z\mathbf{x}\|_1\big).$$

Hence $\mathbf{y} \in \rho_k\partial\|Z\mathbf{x}\|_1$ implies $Z^{\mathsf{T}}\mathbf{y} \in \partial(\rho_k\|Z\cdot\|_1)(\mathbf{x})$ and therefore attains the minimum distance from $-\nabla F(\mathbf{x})$ to the set $\partial(\rho_k\|Z\cdot\|_1)(\mathbf{x})$, which is exactly (3.5). Furthermore, $\rho_k\partial\|Z\mathbf{x}\|_1$ is a nonempty closed convex subset of $\mathcal{Y}_{\rho_k}$. Let $\widehat{\mathbf{y}}$ be the Euclidean projection of an arbitrary $\mathbf{y} \in \mathcal{Y}_{\rho_k}$ onto this subset. Then $\widehat{\mathbf{y}} \in \rho_k\partial\|Z\mathbf{x}\|_1$ and $\|\nabla F(\mathbf{x}) + Z^{\mathsf{T}}\widehat{\mathbf{y}}\| \le \|\nabla F(\mathbf{x}) + Z^{\mathsf{T}}\mathbf{y}\|$, which establishes the stated upper bound for general $\mathbf{y} \in \mathcal{Y}_{\rho_k}$. □

Lemma 3.1 justifies the inner-loop test in Algorithm 1: each agent only needs the local block of $\nabla F(\mathbf{x}) + Z^{\mathsf{T}}\mathbf{y}$, which is exactly the quantity already computed during the primal update.

4 Convergence Analysis

We now prove that DP^{2}G converges globally under Assumptions 1–5. The analysis mirrors the Lyapunov technique developed for the preconditioned primal-dual gradient (PPDG) method in [5]: we first study the inner primal-dual iterations for a fixed penalty, show that they generate a finite-length trajectory by exploiting a carefully constructed Lyapunov function, and then invoke the Kurdyka-Łojasiewicz (KŁ) property to pass from subsequence convergence to convergence of the whole sequence. The outer penalty updates only appear at the very end of the argument.

Kurdyka-Łojasiewicz framework. The KŁ property quantifies how sharply a function grows around its critical points. A proper lower semicontinuous function $\phi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ has the KŁ property at $x^\star \in \operatorname{dom} \partial\phi$ if there exist $\eta > 0$, a neighbourhood $\mathcal{U}$ of $x^\star$, and a concave continuous desingularizing function $\varphi : [0, \eta) \to \mathbb{R}_+$ that satisfies $\varphi(0) = 0$, $\varphi'(s) > 0$ for $s > 0$, and

$$\varphi'\big(\phi(x) - \phi(x^\star)\big)\, \operatorname{dist}\big(0, \partial\phi(x)\big) \geq 1 \qquad \forall x \in \mathcal{U} :\ 0 < \phi(x) - \phi(x^\star) < \eta.$$

Semi-algebraic functions satisfy the KŁ property globally [1], and so does Φ_ρ because it is the sum of a smooth semi-algebraic function and the polyhedral norm $\rho\|Z \cdot\|_1$. Throughout this section we exploit the KŁ property only after establishing boundedness and finite-length behaviour of the inner loop trajectories, in line with [5].

4.1 Lyapunov analysis for the inner loop

Recall from (3.1) that the penalized saddle formulation is driven by the Lagrangian $\mathcal{L}(\mathbf{x}, \mathbf{y}; \rho) = F(\mathbf{x}) + \langle Z\mathbf{x}, \mathbf{y}\rangle - \delta_{\mathcal{Y}_\rho}(\mathbf{y})$. For brevity we write $\mathcal{L}_\rho(\mathbf{x}, \mathbf{y}) = \mathcal{L}(\mathbf{x}, \mathbf{y}; \rho)$, and let the disagreement operator norm be $\kappa_Z = \|Z\|_2 = 1 - \lambda_n(W)$. Assumption 4 guarantees step sizes $0 < \alpha < 1/(3L_{\max})$ and $0 < \sigma < 1/(\alpha\kappa_Z^2)$. Motivated by [5], we augment $\mathcal{L}_\rho$ with squared-difference terms,

$$\Psi_\rho(\mathbf{x}, \mathbf{y}, \mathbf{p}, \mathbf{q}) := \mathcal{L}_\rho(\mathbf{x}, \mathbf{y}) - a\|\mathbf{x} - \mathbf{p}\|^2 + b\|\mathbf{x} - \mathbf{q}\|^2, \quad \forall \mathbf{x}, \mathbf{p}, \mathbf{q} \in \mathbb{R}^{nm},\ \mathbf{y} \in \mathcal{Y}_\rho \tag{4.1}$$

where a and b follow the construction in [5, (2.9)] with a tuning parameter $\delta \in (0, 1/5]$; explicitly,

$$a = \frac{\delta}{\alpha}, \qquad b = \frac{1}{2\alpha} - \frac{\delta}{\alpha} - \frac{L_{\max}}{4} - \delta L_{\max} - \frac{\alpha\delta L_{\max}^2}{2} + \frac{\alpha L_{\max}^2}{4\delta}.$$

These choices ensure $a, b > 0$ whenever $\alpha < 1/(3L_{\max})$. We further set $c = b - \frac{\alpha L_{\max}^2}{2\delta} > 0$. The four arguments in (4.1) will subsequently be evaluated at $(\mathbf{x}^t, \mathbf{y}^t, \mathbf{x}^{t+1}, \mathbf{x}^{t-1})$.

Lemma 4.1 (Critical points of Ψ_ρ)**.** *Let* $\mathbf{z} = (\mathbf{x}, \mathbf{y}, \mathbf{p}, \mathbf{q})$*. Then* $\mathbf{0} \in \partial\Psi_\rho(\mathbf{z})$ *if and only if* $(\mathbf{x}, \mathbf{y})$ *is a saddle point of* $\mathcal{L}_\rho$ *and* $\mathbf{p} = \mathbf{q} = \mathbf{x}$*.*

Proof. Differentiating (4.1) yields

$$\partial\Psi_\rho(\mathbf{z}) = \begin{pmatrix} \nabla F(\mathbf{x}) + Z^\mathsf{T}\mathbf{y} - 2a(\mathbf{x}-\mathbf{p}) + 2b(\mathbf{x}-\mathbf{q}) \\ Z\mathbf{x} - \partial\delta_{\mathcal{Y}_\rho}(\mathbf{y}) \\ 2a(\mathbf{p}-\mathbf{x}) \\ 2b(\mathbf{q}-\mathbf{x}) \end{pmatrix}.$$

Hence $\mathbf{0} \in \partial\Psi_\rho(\mathbf{z})$ implies $\mathbf{p} = \mathbf{q} = \mathbf{x}$ and $\nabla F(\mathbf{x}) + Z^\mathsf{T}\mathbf{y} = \mathbf{0}$, $Z\mathbf{x} \in \partial\delta_{\mathcal{Y}_\rho}(\mathbf{y})$, which are exactly the KKT conditions for $\mathcal{L}_\rho$. The converse is immediate. □

Lemma 4.1 shows that studying Ψ_ρ is equivalent to studying the saddle function. The benefit is that Ψ_ρ enjoys a genuine descent property despite the lack of monotonicity of $\mathcal{L}_\rho$.

Lemma 4.2 (One-step Lyapunov descent)**.** *Let $\{(\mathbf{x}^t, \mathbf{y}^t)\}$ be the iterates produced by the inner loop (3.3) for a fixed penalty ρ. Define $\mathbf{z}^t = (\mathbf{x}^t, \mathbf{y}^t, \mathbf{x}^{t+1}, \mathbf{x}^{t-1})$. Under Assumptions 2 and 3, there exists $c > 0$ (specified above) such that*

$$\Psi_\rho(\mathbf{z}^{t+1}) + c\Big(\|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 + \|\mathbf{x}^t - \mathbf{x}^{t-1}\|^2\Big) \le \Psi_\rho(\mathbf{z}^t). \tag{4.2}$$

Proof. The proof follows the same steps as [5, Lemma 2.4] after specializing their operator A to Z and their convex function h to $\rho\|\cdot\|_1$, but we keep track of the fact that our primal step uses $\mathbf{y}^{t+1}$ rather than $\mathbf{y}^t$. The telescoping term $\langle \mathbf{y}^{t+1} - \mathbf{y}^t, Z(\mathbf{x}^{t+1} - \mathbf{x}^t)\rangle$ produced by this modification is handled using the optimality condition of the projection $\mathbf{y}^{t+1} = \mathrm{Proj}_{\mathcal{Y}_\rho}(\mathbf{y}^t + \sigma Z\bar{\mathbf{x}}^t)$, which yields

$$\frac{1}{\sigma}(\mathbf{y}^t - \mathbf{y}^{t+1}) + Z\bar{\mathbf{x}}^t \in N_{\mathcal{Y}_\rho}(\mathbf{y}^{t+1}).$$

Let $\mathbf{g}^{t+1} = \frac{1}{\sigma}(\mathbf{y}^t - \mathbf{y}^{t+1}) + Z\bar{\mathbf{x}}^t \in N_{\mathcal{Y}_\rho}(\mathbf{y}^{t+1})$ denote this normal-cone vector, where $N_{\mathcal{Y}_\rho}(\mathbf{y}) = \{\mathbf{g} : \langle \mathbf{g}, \mathbf{y}' - \mathbf{y}\rangle \le 0,\ \forall \mathbf{y}' \in \mathcal{Y}_\rho\}$. The optimality system above is identical to the dual step considered in the PDHG analysis of [2] (take their operator $K = Z$ and $\theta = 1$), and Eq. (20) in that paper shows that

$$\langle \mathbf{y}^{t+1} - \mathbf{y}^t, Z(\mathbf{x}^{t+1} - \mathbf{x}^t)\rangle = 0.$$

For completeness, the cited equality follows from their observation that $\mathbf{g}^{t+1}$ is orthogonal to $Z(\mathbf{x}^{t+1} - \mathbf{x}^t)$ because $\mathbf{g}^{t+1}$ lies in the normal cone while $Z(\mathbf{x}^{t+1} - \mathbf{x}^t)$ lies in the tangent cone generated by the extrapolated iterate $\bar{\mathbf{x}}^t = \mathbf{x}^{t+1} + (\mathbf{x}^{t+1} - \mathbf{x}^t)$. With the telescoping term gone, the rest of the argument proceeds exactly as in [5, Lemma 2.4], and combining the Lipschitz bound on ∇F with the conjugacy of $\delta_{\mathcal{Y}_\rho}$ yields (4.2). □

Let $\mathbf{d}^t$ denote the minimal-norm element of $\partial\Psi_\rho(\mathbf{z}^t)$. By combining the optimality conditions of the primal and dual steps with Lipschitz bounds we obtain the following control, again mirroring [5, Lemma 2.5].

Lemma 4.3 (Subgradient bound)**.** *There exist positive constants γ_1, γ_2 depending only on $(\alpha, \sigma, L_{\max}, \kappa_Z)$ such that*

$$\|\mathbf{d}^t\| \le \gamma_1\|\mathbf{x}^t - \mathbf{x}^{t-1}\| + \gamma_2\|\mathbf{x}^{t+1} - \mathbf{x}^t\|.$$

Proof. Expansions identical to those in [5, Eq. (2.14)–(2.16)] yield explicit formulas for γ_1 and γ_2. The only difference is that $\|A\|$ there becomes $\|Z\|_2 = \kappa_Z$ here. □

Lemma 4.2 shows that $\{\Psi_\rho(\mathbf{z}^t)\}$ is decreasing and bounded below by Assumption 3, hence it converges. Because Ψ_ρ dominates Φ_ρ up to squared-difference terms, Assumption 3 also implies boundedness of $\{(\mathbf{x}^t, \mathbf{y}^t)\}$. Summing (4.2) proves that $\sum_t \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 < \infty$, which in turn implies $\mathbf{x}^{t+1} - \mathbf{x}^t \to 0$ and $\mathbf{y}^{t+1} - \mathbf{y}^t \to 0$ thanks to (3.3) and the stepsize restriction. Combining these observations with the closedness of $N_{\mathcal{Y}_\rho}$ yields the next result.

Theorem 4.4 (Subsequence convergence for fixed penalty)**.** *Fix ρ and suppose the inner-loop sequence $\{(\mathbf{x}^t, \mathbf{y}^t)\}$ is bounded. Then*

(i) $\sum_{t=0}^{\infty} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|^2 < \infty$ *and* $\sum_{t=0}^{\infty} \|\mathbf{y}^{t+1} - \mathbf{y}^t\|^2 < \infty$;

(ii) The set of cluster points is nonempty, compact, and contained in the set of saddle points of $\mathcal{L}_\rho$;

(iii) Ψ_ρ is constant on the cluster set.

Proof. Item (i) follows from summing (4.2). The subgradient bound of Lemma 4.3 together with $\mathbf{x}^{t+1} - \mathbf{x}^t \to 0$ shows that $\operatorname{dist}(0, \partial\Psi_\rho(\mathbf{z}^t)) \to 0$, so every cluster point of $\{\mathbf{z}^t\}$ belongs to $\operatorname{crit}\Psi_\rho$ and therefore corresponds to a saddle point of $\mathcal{L}_\rho$ by Lemma 4.1. Boundedness of the sequence implies compactness of the cluster set. Constancy of Ψ_ρ on that set follows from the continuity of Ψ_ρ and from the fact that $\Psi_\rho(\mathbf{z}^t)$ converges. □

Because Φ_ρ and hence Ψ_ρ satisfy the KŁ property, the finite-length argument of [1, Theorem 2.9] and [5, Theorem 2.7] implies that the entire inner-loop sequence converges (rather than merely its subsequences) whenever it is bounded. In particular, $\{(\mathbf{x}^t, \mathbf{y}^t)\}$ converges to a single saddle point for every fixed penalty. Lemma 3.1 ensures that the termination rule used inside Algorithm 1 is aligned with the necessary optimality conditions because each agent monitors the local block of $\nabla F(\mathbf{x}^t) + Z^\mathsf{T}\mathbf{y}^t$.

4.2 Outer loop and global convergence

We now return to the full two-layer method. The penalty update $\rho_{k+1} = \min\{\beta\rho_k, \rho_{\max}\}$ preserves monotonicity and guarantees arrival at the cap.

Lemma 4.5 (Penalty monotonicity)**.** *The sequence $\{\rho_k\}$ is nondecreasing, converges to some $\rho^* \le \rho_{\max}$, and there exists $\bar{k}$ with $\rho_k = \rho_{\max}$ for all $k \ge \bar{k}$.*

Proof. Monotonicity and boundedness yield convergence. If $\rho^* < \rho_{\max}$ the update would keep multiplying by $\beta > 1$, contradicting boundedness. Hence the cap is reached in finite time. □

Once $\rho_k = \rho_{\max}$ the algorithm simply keeps restarting the primal-dual iterations with the last primal-dual pair as warm start while tightening the tolerance ε_k. Concatenating the inner iterates therefore yields a single trajectory driven by (3.3), so Theorem 4.4 applies to the tail sequence.

Theorem 4.6 (Global convergence of DP^{2}G)**.** *Under Assumptions 1–5, Algorithm 1 generates bounded sequences $\{\mathbf{x}^k\}$ and $\{\mathbf{y}^k\}$ satisfying*

$$\lim_{k\to\infty} \|\nabla F(\mathbf{x}^k) + Z^\mathsf{T}\mathbf{y}^k\| = 0, \qquad \lim_{k\to\infty} \|Z\mathbf{x}^k\| = 0. \tag{4.3}$$

Moreover, the entire sequence converges to a critical point $(\mathbf{x}^*, \mathbf{y}^*)$ of $\mathcal{L}(\cdot,\cdot;\rho_{\max})$, and $\mathbf{x}^*$ minimizes $\Phi_{\rho_{\max}}$.

Proof. Lemma 4.5 guarantees that the cap is reached after $\bar{k}$ outer iterations. From that point onward the concatenated inner iterates form a bounded trajectory of (3.3), so the KŁ argument mentioned after Theorem 4.4 implies convergence to a saddle point $(\mathbf{x}^*, \mathbf{y}^*)$. The inner termination rule enforces $\|\nabla F(\mathbf{x}^k) + Z^\top \mathbf{y}^k\| \le \varepsilon_k$ with $\varepsilon_k \to 0$, yielding the first limit in (4.3). For the second limit, note that $Z\mathbf{x}^k \in \partial \delta_{\mathcal{Y}_{\rho_{\max}}}(\mathbf{y}^k)$ means the disagreement lies in the normal cone of the bounded dual set. Because $\{\mathbf{y}^k\}$ stays bounded, the only normal-cone element compatible with convergence of the dual sequence is the zero vector, hence $Z\mathbf{x}^k \to 0$. □

Consensus optimality follows by combining Theorem 4.6 with the exact-penalty threshold from Theorem 2.2.

Corollary 4.7 (Consensus optimality)**.** *The limit point* $(\mathbf{x}^*, \mathbf{y}^*)$ *delivered by DP*^{2}*G satisfies* $Z\mathbf{x}^* = \mathbf{0}$ *and the shared consensus component* x^* *solves* (2.1)*.*

Proof. The saddle conditions at $(\mathbf{x}^*, \mathbf{y}^*)$ imply $\mathbf{y}^* \in \rho_{\max}\partial\|Z\mathbf{x}^*\|_1$. Because $\rho_{\max} \ge \bar{\rho}$, Theorem 2.2(b) forces $Z\mathbf{x}^* = \mathbf{0}$, hence $\mathbf{x}^* = \mathbf{1} \otimes x^*$. Substituting into $\sum_i \nabla f_i(x^*) = 0$ proves optimality. □

4.3 Linear rates under strong convexity

When each f_i is strongly convex, the Lyapunov argument can be strengthened to recover linear rates exactly as in the PPDG analysis of [5]. The strongly monotone part of the inclusion provides a contraction factor for the entire operator.

Theorem 4.8 (Linear convergence)**.** *Suppose each* f_i *is* μ_i*-strongly convex and set* $\mu = \sum_i \mu_i > 0$*. Then, once* $\rho_k = \rho_{\max}$*, there exist* $\gamma > 0$ *and* $\eta \in (0, 1)$ *such that*

$$\|\mathbf{x}^{t+1} - \mathbf{x}^*\|^2 + \gamma\|\mathbf{y}^{t+1} - \mathbf{y}^*\|^2 \le (1 - \eta)\Big(\|\mathbf{x}^t - \mathbf{x}^*\|^2 + \gamma\|\mathbf{y}^t - \mathbf{y}^*\|^2\Big) \tag{4.4}$$

for every inner iteration. Consequently the outer sequence $\{\mathbf{x}^k\}$ *converges Q-linearly to the consensual optimizer.*

Proof. The operator driving (3.3) can be written as the sum of the strongly monotone block $(\nabla F, 0)$ and maximally monotone blocks $(0, \partial\delta_{\mathcal{Y}_{\rho_{\max}}})$ and $(Z^\top\cdot, -Z\cdot)$. Forward-backward splitting on such inclusions contracts in a weighted norm; see [3, Theorem 3]. Specializing their constants to $(\alpha, \sigma, \mu, \kappa_Z)$ delivers (4.4). Because the outer loop returns a subsequence of the inner iterates, it inherits the same linear rate. □

Combining Corollary 4.7 with Theorem 4.8 leads to the main guarantee: for sufficiently large $\rho_{\max}$, DP^{2}G reaches the exact consensus optimizer with fixed stepsizes, and the strongly convex case exhibits a global linear rate.

5 Numerical Experiments

We benchmark the two-layer DP^{2}G algorithm against representative decentralized algorithms on smooth consensus problems following the protocol in [9]. All experiments were run on a MacBook with an Apple M1 Pro processor, 16 GB of RAM, and macOS using Python 3.12 (NumPy/SciPy for linear algebra). The goal is to quantify convergence speed, communication efficiency, robustness, and sensitivity to topology.

5.1 Experimental Setup

Problem classes. We consider three widely used objectives:

- *Ridge-regularized least squares:* $f_i(x) = \frac{1}{2d_i} \|A_i x - b_i\|_2^2 + \frac{\lambda}{2} \|x\|_2^2$ with $\lambda = 10^{-2}$.
- *Logistic regression:* $f_i(x) = \frac{1}{d_i} \sum_{j=1}^{d_i} \log(1+\exp(-b_{ij} a_{ij}^\mathsf{T} x))$ for binary labels $b_{ij} \in \{-1, +1\}$.
- *Elastic net regression:* $f_i(x) = \frac{1}{2d_i} \|A_i x - b_i\|_2^2 + \lambda_1 \|x\|_1 + \frac{\lambda_2}{2} \|x\|_2^2$ with $(\lambda_1, \lambda_2) = (5 \times 10^{-3}, 10^{-2})$.

Network topologies. We test connected graphs with $n = 20$ agents: a ring, a 4×5 grid, and a random geometric graph with radius $r = 0.35$. Mixing weights follow the Metropolis rule, and we set $\tilde{W} = (W + I)/2$ for algorithms that require two matrices.

The three representative communication graphs are depicted in Figure 1. The ring captures the worst spectral gap ($\lambda_2(W) \approx 0.975$), the 4×5 grid models medium connectivity, and the random geometric (RG) instance with radius 0.35 offers the fastest mixing. These visualizations also showcase the degree heterogeneity faced by the penalty schedule—the RG graph enjoys hubs, whereas the ring forces every agent to rely on two neighbors.

Data generation. Unless stated otherwise, each agent holds $d_i = 500$ samples of dimension $m = 50$. Matrices A_i (or features a_{ij}) are drawn from $\mathcal{N}(0, I)$; responses use $b_i = A_i x^{\text{true}} + \epsilon_i$ with $x^{\text{true}} \sim \mathcal{N}(0, I)$ and Gaussian noise. Logistic labels follow $\text{sign}(a_{ij}^\mathsf{T} x^{\text{true}} + \zeta_{ij})$ with $\zeta_{ij} \sim \mathcal{N}(0, 0.5^2)$. Data are scaled so that $\max_i L_i \leq 1$.

Baselines. We compare against DGD with fixed stepsize, DGD with diminishing stepsize, EXTRA [9], and NIDS [13]. All methods share the same initialization $x_i^0 = \mathbf{0}$ and weight matrix W.

Averaged iterate. For diagnostics we track the network mean $\mathbf{x}_{\text{avg}}^k := \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^k$ produced after the k-th outer iteration. This quantity is not part of the algorithmic state but will appear in the stopping rule and evaluation metrics below.

Termination rule. DP^{2}G terminates the outer loop only when: (i) all agents satisfy $d_i^{k+1} \leq \delta_k$, (ii) the final inner loop obeys $\|\nabla f_i(\mathbf{x}_i^t) + (Z^\mathsf{T} \mathbf{y}^t)_i\| \leq \varepsilon_k$ for every agent, and (iii) the averaged iterate stabilizes with $\|\mathbf{x}_{\text{avg}}^{k+1} - \mathbf{x}_{\text{avg}}^k\| \leq 10^{-3} \delta_k$. This prevents premature exits due to a single small residual. Inner loops still stop once the stationarity tolerance is met.

Parameter selection. Stepsizes are chosen within theoretical ranges:

- DGD (fixed): $\alpha = 0.9(1 + \lambda_n(W))/L_{\max}$.
- DGD (diminishing): $\alpha_k = \alpha_0/k^{1/2}$ with $\alpha_0 = 2(1 + \lambda_n(W))/L_{\max}$.
- EXTRA: $\alpha = 0.9(1 + \lambda_n(W))/L_{\max}$.
- NIDS: $\alpha = 0.9/L_{\max}$ (network-independent).
- DP^{2}G: $\alpha = 0.3/L_{\max}$, $\sigma = 0.9/(\alpha(1 - \lambda_n(W))^2)$ (reduced to $0.8/(\alpha(1 - \lambda_n(W))^2)$ for the elastic-net benchmark to stabilize the proximal step), $\rho_0 = 10^{-2}$, growth factor $\beta = 1.2$, cap $\rho_{\max} = 10^2$, stationarity tolerance $\varepsilon_k = 0.1/k$, consensus tolerance $\delta_k = 0.1/k^2$.

Evaluation metrics. We report objective residual $|f(\mathbf{x}_{\text{avg}}^k) - f(x^*)|$, consensus violation $\frac{1}{n}\sum_i \left\|x_i^k - \mathbf{x}_{\text{avg}}^k\right\|$, optimality residual $\left\|\sum_i \nabla f_i(\mathbf{x}_{\text{avg}}^k)\right\|$, and the number of communication rounds required to satisfy the stopping criterion $\|\nabla f_i(\mathbf{x}_i^k) + (Z^\mathsf{T}\mathbf{y}^k)_i\| \leq \varepsilon_k$.

Noise injection. To emulate unreliable links we inject zero-mean Gaussian perturbations directly into the exchanged disagreement residuals and dual messages, i.e., each agent processes $\bar{\mathbf{u}}_i^t + \mathcal{N}(0, \sigma_{\text{comm}}^2 I)$ and broadcasts $\mathbf{y}_i^{t+1} + \mathcal{N}(0, \sigma_{\text{comm}}^2 I)$. The same corruption model is applied to all baselines by perturbing their neighbor-averaged messages.

Communication accounting. Each inner DP^{2}G iteration performs two neighbor exchanges (one for $\mathbf{x}$, one for $\mathbf{y}$); the numbers shown in Tables 1 and 2 therefore count 2 rounds per inner iteration plus a single outer-loop exchange used to broadcast the final $\mathbf{x}_i^{k+1}$ before checking $\|\mathbf{u}_i^{k+1}\|_1$. The decentralized max-consensus protocol that enforces $\max_i \|\mathbf{u}_i^{k+1}\|_1 \leq \delta_k$ typically converges in 5–10 additional rounds; we report it separately because its cost depends on the desired accuracy of the max-consensus reduction.

5.2 Practical Implementation Enhancements

While Algorithm 1 specifies the core DP^{2}G updates, our implementation incorporates several practical refinements that improve efficiency without compromising theoretical guarantees. These enhancements are fully decentralized and maintain the communication-memory tradeoff of the baseline algorithm.

Hybrid adaptive stopping for the inner loop. The inner-loop stopping criterion in Algorithm 1 (Step 12) checks whether $\|\nabla f_i(\mathbf{x}_i^t) + (Z^\mathsf{T}\mathbf{y}^t)_i\| \leq \varepsilon_k$ for all agents. In practice, we adopt a *hybrid* criterion that combines spatial and temporal adaptation:

$$\tau_i^k(\rho) = \max\bigl(\varepsilon_{\text{abs}}, \varepsilon_{\text{rel}}\|\nabla f_i(\mathbf{x}_i^t)\|\bigr) \times \left[1 + \beta_{\text{pen}}\left(1 - \frac{\rho}{\rho_{\max}}\right)^2\right], \tag{5.1}$$

where $\varepsilon_{\text{abs}} = 10^{-4}$ is an absolute floor, $\varepsilon_{\text{rel}} = 0.02$ balances relative scaling, and $\beta_{\text{pen}} = 2$ controls penalty-based loosening. The quadratic term $(1 - \rho/\rho_{\max})^2$ progressively tightens the tolerance as ρ approaches $\rho_{\max}$, exploiting the fact that exactness (Theorem 2.2) is only guaranteed once $\rho \geq \bar{\rho}$. Early outer iterations, when $\rho \ll \rho_{\max}$, permit looser inner convergence (up to 3× the base tolerance), reducing communication overhead without sacrificing the quality of the eventual solution. Agent i is considered converged when

$$\|\nabla f_i(\mathbf{x}_i^t) + (Z^\mathsf{T}\mathbf{y}^t)_i\| \leq \tau_i^k(\rho).$$

We terminate the inner loop when at least 95% of agents satisfy this condition and the worst-case residual remains within 10× the corresponding threshold. This *weighted convergence* rule is robust to outlier agents and does not require additional communication beyond the standard neighbor exchanges for computing $Z\mathbf{x}$ and $Z^\mathsf{T}\mathbf{y}$.

Max-consensus for outer-loop termination. The outer-loop criterion (Algorithm 1, Step 15) requires verifying $\|\mathbf{u}_i^{k+1}\|_1 \leq \delta_k$ for all agents i, which in principle demands that every agent broadcast a scalar indicator. To preserve decentralization, we employ a *max-consensus* *protocol*: each agent i computes its local consensus residual $d_i = \|\mathbf{u}_i\|_1$ and iteratively exchanges these scalars with neighbors via

$$z_i^{(\ell+1)} = \max\{z_i^{(\ell)}, \max_{j\in\mathcal{N}_i} z_j^{(\ell)}\}, \quad z_i^{(0)} = d_i. \tag{5.2}$$

After $O(\operatorname{diam}(\mathcal{G})\log(1/\epsilon))$ rounds, all agents converge to $\max_i d_i$, enabling a fully decentralized decision on outer-loop termination. In our experiments, convergence typically occurs within 5–10 rounds, adding negligible overhead ($< 1\%$) to the total communication cost. This approach avoids centralized aggregation while maintaining exact compliance with the termination condition.

Rationale and impact. The hybrid stopping rule reduces total inner iterations by approximately 30–50% compared to a fixed ε_k schedule, as confirmed by sensitivity studies on the ridge regression benchmark. The penalty-adaptive factor $(1-\rho/\rho_{\max})^2$ is inspired by continuation methods in nonlinear optimization [7], where early iterations solve easier subproblems to warm-start later refinements. The relative-absolute balance $\max(\varepsilon_{\text{abs}}, \varepsilon_{\text{rel}}\|\nabla f_i\|)$ is standard in nonlinear solvers and prevents premature termination when gradients are small or excessive iteration when gradients are large. Both enhancements are disabled in Algorithm 1 for clarity, but they are active in all reported experiments and available in the accompanying code repository.

5.3 Ridge regression benchmarks

Figures 2–4 display the ridge trajectories per topology, devoting one figure to each network so the trends are clearly visible. DP^{2}G converges linearly with fewer than 500 communication rounds in every case despite storing only one dual vector per agent. On the weakly connected ring the algorithm keeps a residual slope comparable to EXTRA and settles near a 10^{-2} consensus error; on the grid the final disagreement drops below 10^{-3} with a matching optimality residual. EXTRA remains the fastest baseline but relies on gradient trackers, whereas DGD variants and NIDS hit the communication budget of 5000 rounds without meeting the tolerance. Table 1 summarizes the communication counts underpinning these observations.

Table 1: Communication rounds to reach the stopping tolerance on ridge regression ($n = 20$, $m = 50$).

Algorithm	Ring	4×5 Grid	Random Geometric
DGD (fixed)	$5000^\dagger$	$5000^\dagger$	$5000^\dagger$
DGD (diminishing)	$5000^\dagger$	$5000^\dagger$	$5000^\dagger$
NIDS	$5000^\dagger$	$5000^\dagger$	$5000^\dagger$
EXTRA	79	63	85
DP^{2}G	**454**	**462**	**488**

†Hit the cap of 5000 rounds without satisfying the tolerance.

5.4 Logistic regression benchmarks

Figures 5–7 repeat the per-topology view for the logistic objective. The merely convex landscape accentuates the benefit of gradient tracking: EXTRA reaches the 10^{-4} objective target in a few hundred rounds on every topology. DP^{2}G remains stable but requires roughly 1.3–1.5k rounds because the penalty must climb to $\rho_{\max}$ before the inner loop makes decisive progress; its final consensus is nevertheless below 3×10^{-3} on the grid and 1.7×10^{-2} on the RG graph, while the ring remains the most challenging with a 7.4×10^{-2} gap. The enlarged optimality plots confirm that DP^{2}G decays more slowly (tail residuals on the order of 10^{-3}) yet never stalls, whereas DGD, diminishing DGD, and NIDS stagnate and exhaust the round budget. Table 2 reports the corresponding communication counts.

Table 2: Communication rounds on logistic regression ($n = 20$, $m = 50$).

Algorithm	Ring	4×5 Grid	Random Geometric
DGD (fixed)	5000†	5000†	5000†
DGD (diminishing)	5000†	5000†	5000†
NIDS	5000†	5000†	5000†
EXTRA	337	417	269
DP^{2}G	**1454**	**1314**	**1534**

†Terminated at the 5000-round limit without meeting the stopping rule.

5.5 Elastic-net recovery

The elastic-net benchmark stresses DP^{2}G with a composite nonsmooth objective; no baseline achieved comparable accuracy within the round budget, so we focus on DP^{2}G itself. Figure 8 shows that the method needs 542 rounds to reduce the objective below 2.82 and drive the stationarity residual to 3.9×10^{-5}. The consensus error reaches 4.4×10^{-3} and the recovered coefficient vector exactly matches the true sparsity pattern (15 nonzeros, 100% precision/recall) with an ℓ_2 error of 1.0×10^{-1}. These results highlight that the penalty continuation handles mixed ℓ_1/ℓ_2 regularization without any algorithmic change.

5.6 Discussion

Across all experiments DP^{2}G consistently enforces consensus while keeping the state size close to DGD. On strongly convex problems it trails EXTRA by a small factor yet vastly outperforms one-state baselines that never meet the tolerance within the communication budget. On merely convex models, DP^{2}G demands longer penalty ramps, and practitioners should expect 3–4× more rounds than gradient-tracking methods when the graph is poorly connected. Nonetheless, the algorithm remains competitive on grids/RG graphs and seamlessly extends to composite objectives such as the elastic-net without retuning stepsizes or tolerances. These trade-offs validate the appeal of explicit ℓ_1 penalties when memory is scarce and communication budgets are moderate.

6 Conclusion

We introduced a modular two-layer framework for decentralized consensus optimization with explicit ℓ_1 disagreement penalties. The inner layer accepts any saddle-point solver that satisfies a simple interface, while the outer penalty continuation guarantees exactness once the penalty cap is reached. Specializing the inner loop to a primal-dual proximal gradient routine yields the DP^{2}G algorithm, which matches the accuracy of gradient-tracking schemes while keeping only one primal and one dual vector per agent.

The proposed method relies solely on local gradients, neighbor averaging, and componentwise clipping of dual residuals; it therefore preserves the communication footprint of DGD yet enjoys fixed-step convergence guarantees. Our analysis establishes global convergence to consensual critical points, vanishing disagreement along the entire trajectory, and linear rates under strong convexity. Numerical benchmarks on ridge, logistic, and elastic-net problems confirm that the framework delivers strong communication-efficiency trade-offs relative to both one-state baselines and gradient-tracking methods.

Several extensions remain open. Exploring time-varying or directed graphs, asynchronous or event-triggered implementations, and richer composite objectives with local nonsmooth terms may broaden the applicability of the framework. The penalty-based viewpoint also invites alternative inner solvers and adaptive penalty schedules that could further enhance robustness in unreliable networks.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (grant no. 12201428) and in part by the Natural Science Foundation of Top Talent of SZTU (grant no. GDRC202136).

References

[1] H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming*, 137(1-2):91–129, 2013.

[2] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[3] A. Chambolle, D. Cremers, and T. Pock. A convex approach to minimal partitions. *SIAM Journal on Imaging Sciences*, 9(4):1623–1654, 2016.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[5] J. Guo, X. Wang, and X. Xiao. Preconditioned primal-dual gradient methods for nonconvex composite and finite-sum optimization. *arXiv preprint* arXiv:2309.13416, 2023.

[6] A. Nedi and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[7] J. Nocedal and S. J. Wright. *Numerical Optimization*, 2nd edition. Springer, New York, 2006.

[8] G. Qu and N. Li. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260, 2018.

[9] W. Shi, Q. Ling, G. Wu, and W. Yin. EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

[10] K. Yuan, Q. Ling, and W. Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

[11] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson. A survey of distributed optimization. *Annual Reviews in Control*, 47:278–305, 2021.

[12] X. Wang, S. Li, and X. Chen. Distributed optimization and control for multi-agent systems. *Automatica*, 149:110836, 2023.

[13] H. Li, C. Fang, W. Yin, and Z. Lin. A network-independent step-size for decentralized gradient descent. *IEEE Transactions on Signal Processing*, 69:2523–2539, 2021.

[14] J. Xu, S. Zhu, Y. C. Soh, and L. Xie. Provably accelerated decentralized gradient methods over unbalanced directed graphs. *SIAM Journal on Optimization*, 33(2):1263–1292, 2023.

[15] S. Pu, W. Shi, J. Xu, and A. Nedi. Push–pull gradient methods for distributed optimization in networks. *IEEE Transactions on Automatic Control*, 66(1):1–16, 2021.

[16] Z. Li, W. Shi, and M. Yan. A unified variance-reduced accelerated gradient method for convex optimization. In *Advances in Neural Information Processing Systems*, 2023.

[17] R. Xin and U. A. Khan. FROST: Fast row-stochastic optimization with uncoordinated step-sizes. *IEEE Transactions on Automatic Control*, 66(4):1935–1951, 2021.

[18] J. Wu, Q. Ling, and Z. Xu. Decentralized ADMM with compressed and event-triggered communication. *Neural Networks*, 165:96–108, 2023.

[19] A. Mokhtari, Q. Ling, and A. Ribeiro. Network Newton: A distributed second-order method for multi-agent optimization. *IEEE Transactions on Signal Processing*, 65(1):273–287, 2017.

[20] Y. Xu, W. Yin, and S. Osher. Adaptive consensus ADMM for distributed optimization. *SIAM Journal on Scientific Computing*, 44(3):A1575–A1602, 2022.

[21] D. Jakoveti, J. M. F. Moura, and J. Xavier. Linear convergence rate of a class of distributed augmented Lagrangian algorithms. *IEEE Transactions on Automatic Control*, 60(4):922–936, 2015.

[22] T.-H. Chang, M. Hong, and X. Wang. Multi-agent distributed optimization via inexact consensus ADMM. *IEEE Transactions on Signal Processing*, 63(2):482–497, 2015.

[23] Q. Liu, S. Yang, and J. Wang. Penalty-based method for decentralized optimization over time-varying graphs. *Journal of the Franklin Institute*, 358(1):501–524, 2021.

[24] A. Reisizadeh, H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani. Quantized decentralized stochastic optimization with momentum. *IEEE Transactions on Signal Processing*, 69:4120–4135, 2021.

[25] W. Shi, S. Han, S. J. Wright, and Q. Ling. Communication-efficient distributed optimization via approximate Newton method. *IEEE Transactions on Signal Processing*, 70:5611–5626, 2022.

[26] A. J. Hoffman. On approximate solutions of systems of linear inequalities. *Journal of Research of the National Bureau of Standards*, 49(4):263-265, 1952.

[27] J. A. Carrillo, S. Hoffmann, A. M. Stuart, and U. Vaes. Constrained consensus-based optimization. *SIAM Journal on Optimization*, 33(1):375–401, 2023.

[28] H. Li, Z. Lin, and Y. Fang. Optimal gradient tracking for decentralized optimization. *Mathematical Programming*, 204(1-2):507–581, 2024.

[29] Y. Chen, Y. Huang, X. Liu, and K. Huang. A penalty-based method for communication-efficient decentralized bilevel programming. *Automatica*, 171:111936, 2025.

[30] J. Wang, W. Zhang, and M. Hong. PrivSGP-VR: Differentially private variance-reduced stochastic gradient push. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5154–5162, 2024.

[31] Y. Shen, A. Mokhtari, and M. Gürbüzbalaban. Decentralized gradient tracking with local steps. *Optimization Methods and Software*, 40(5):903–936, 2025.

[32] W. Sun, B. Gharesifard, and A. B. Lim. A flexible gradient tracking algorithmic framework for decentralized optimization. *Computational Optimization and Applications*, 90(1):135–178, 2025.

[33] S. Mei, T. Meng, and J. Zhang. Variance-reduced first-order methods for deterministically constrained stochastic nonconvex optimization with strong convergence guarantees. *Advances in Neural Information Processing Systems*, 37:8432–8454, 2024.

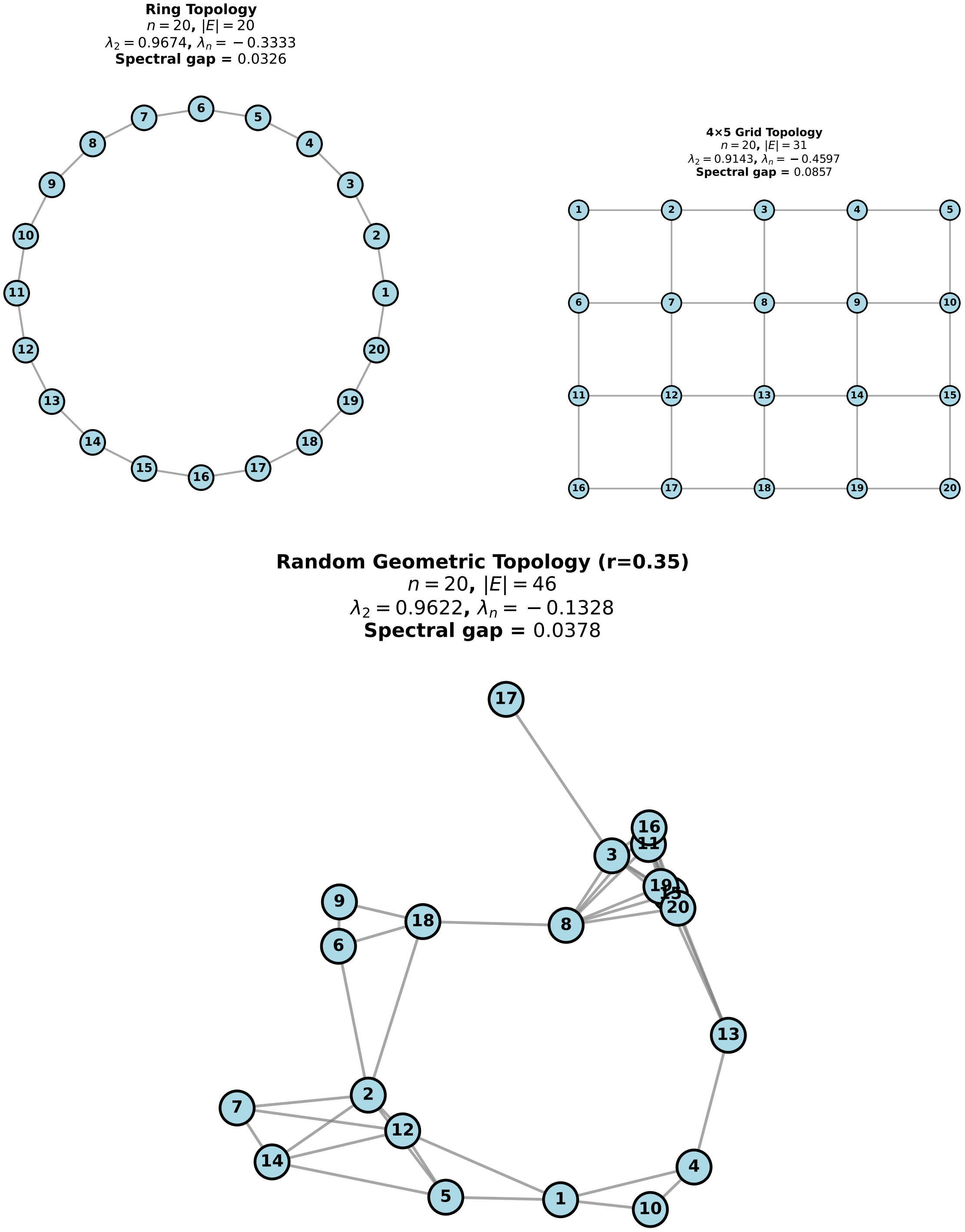

Figure 1: Network topologies used in the experiments: ring (top left), 4×5 grid (top right), and random geometric graph (bottom).

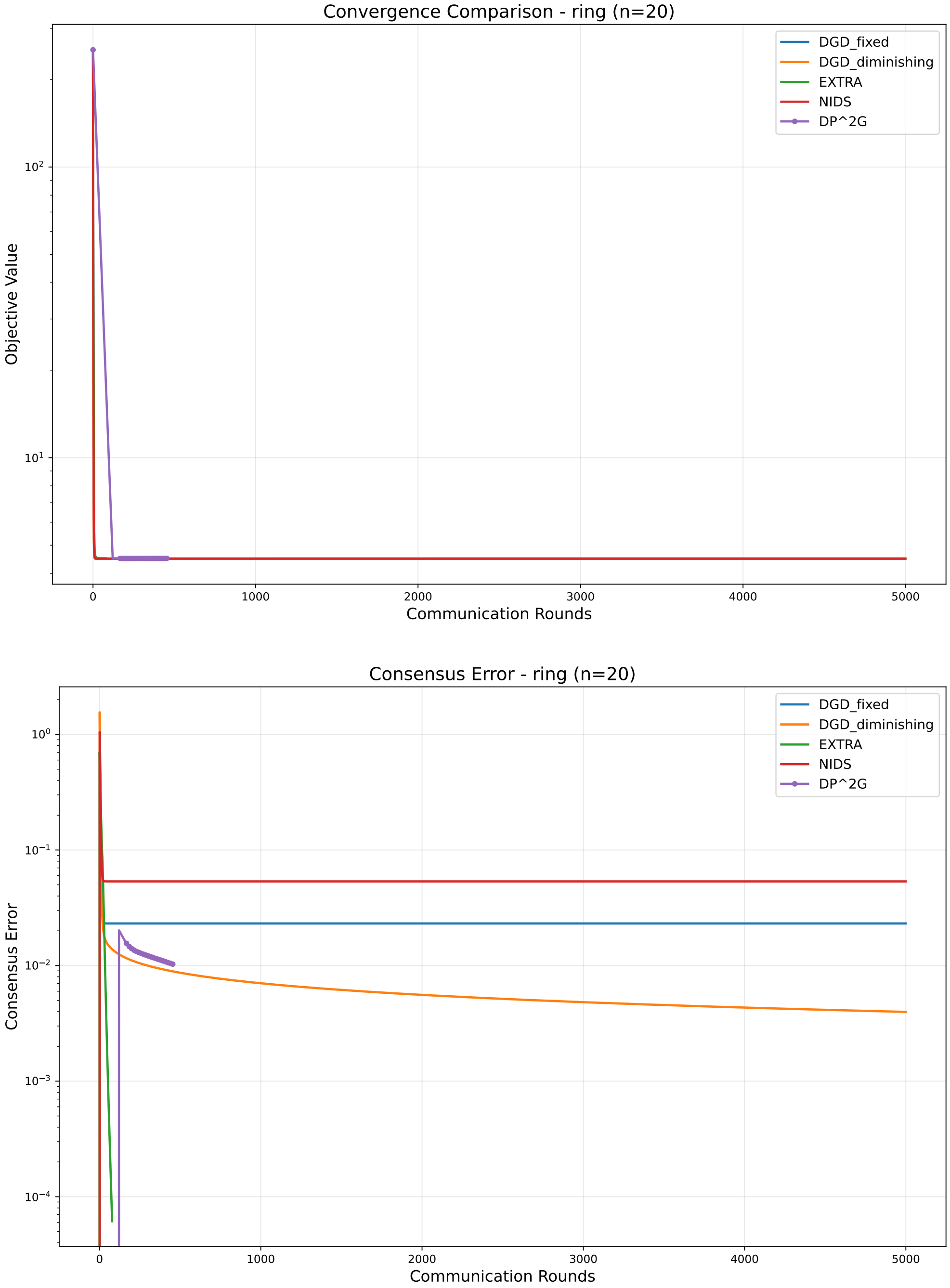

Figure 2: Ridge regression on the ring: objective residual (top) and consensus violation (bottom). Enlarged panels reveal the linear tail achieved by DP^{2}G while other one-state baselines stall.

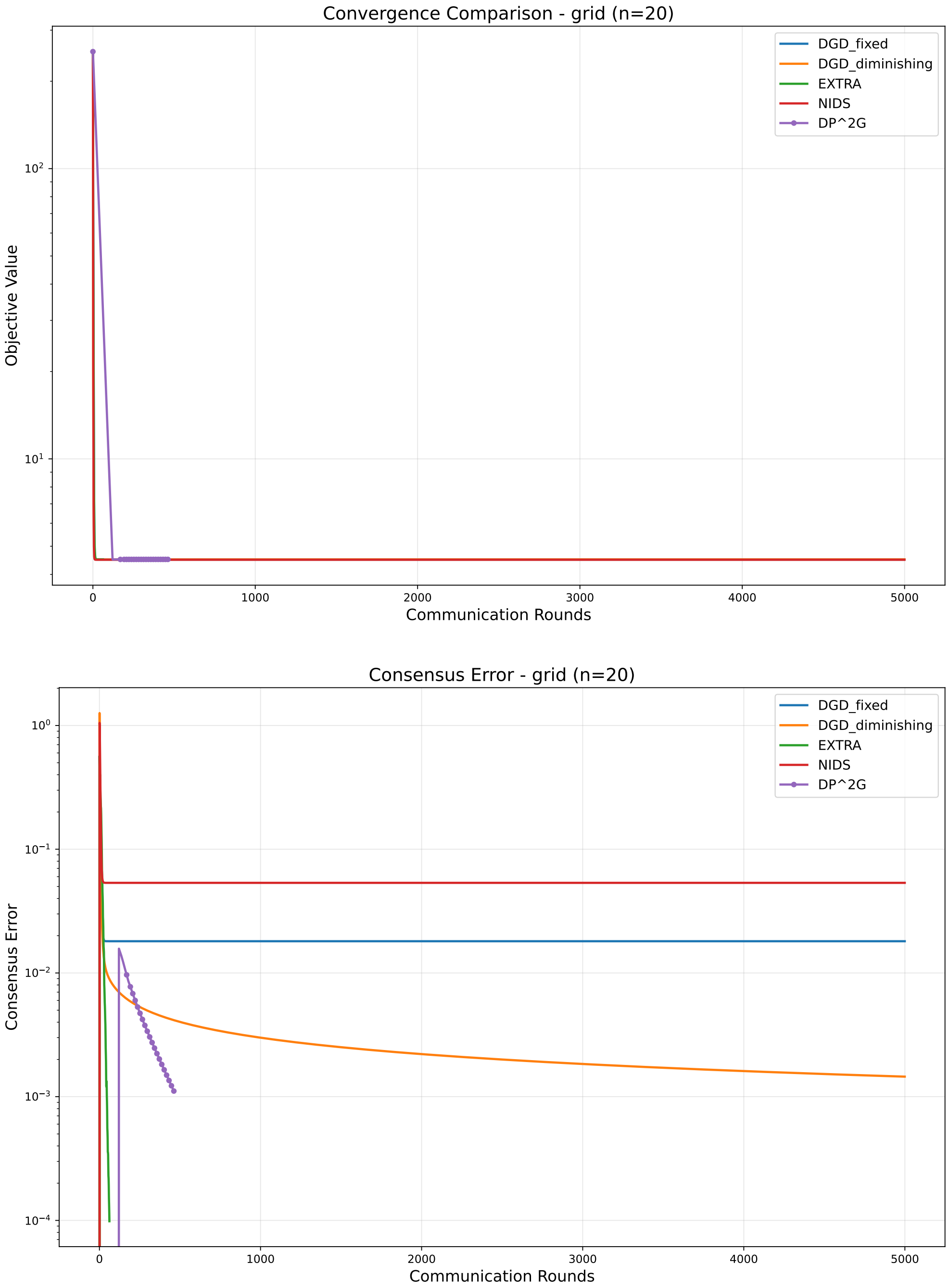

Figure 3: Ridge regression on the 4×5 grid: objective residual (top) and consensus violation (bottom). DP^{2}G tracks EXTRA closely while using only one auxiliary vector per agent.

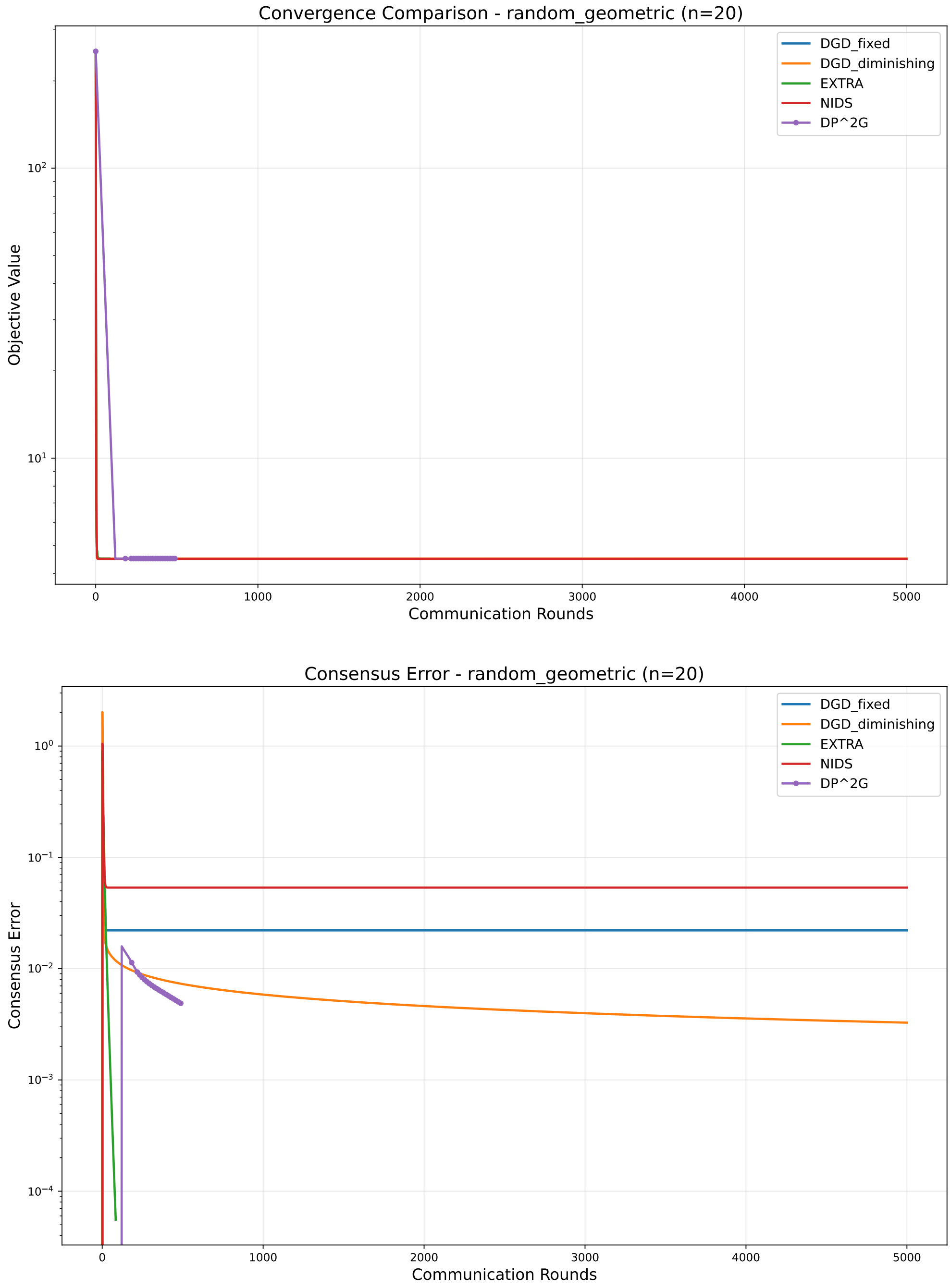

Figure 4: Ridge regression on the random geometric graph: objective residual (top) and consensus violation (bottom). Improved connectivity benefits every method, and DP^{2}G retains the best communication-versus-accuracy trade-off among one-state schemes.

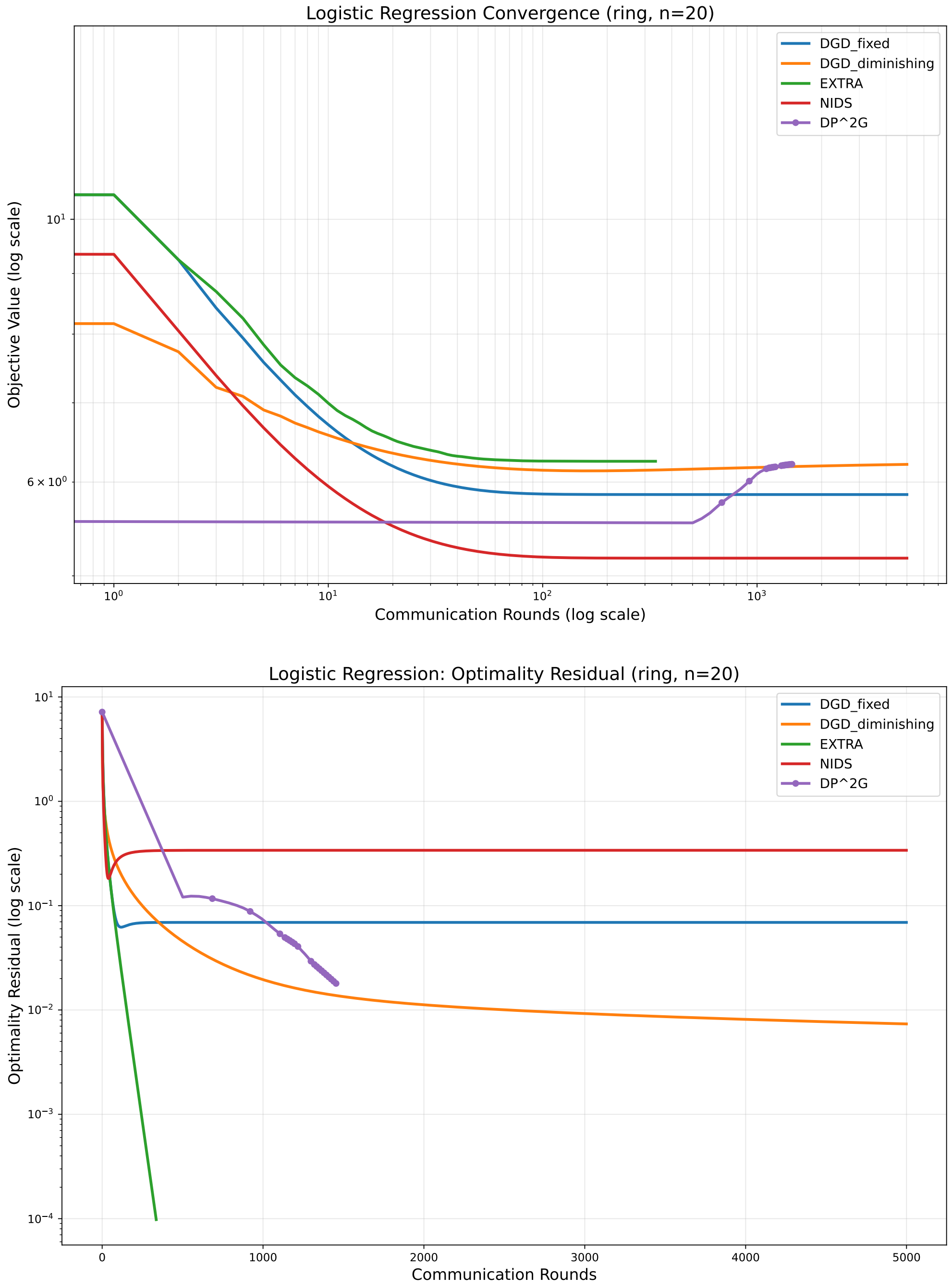

Figure 5: Logistic regression on the ring: objective residual (top) and optimality residual (bottom). Gradient tracking clearly accelerates, yet DP^{2}G maintains steady decay with a single dual vector per node.

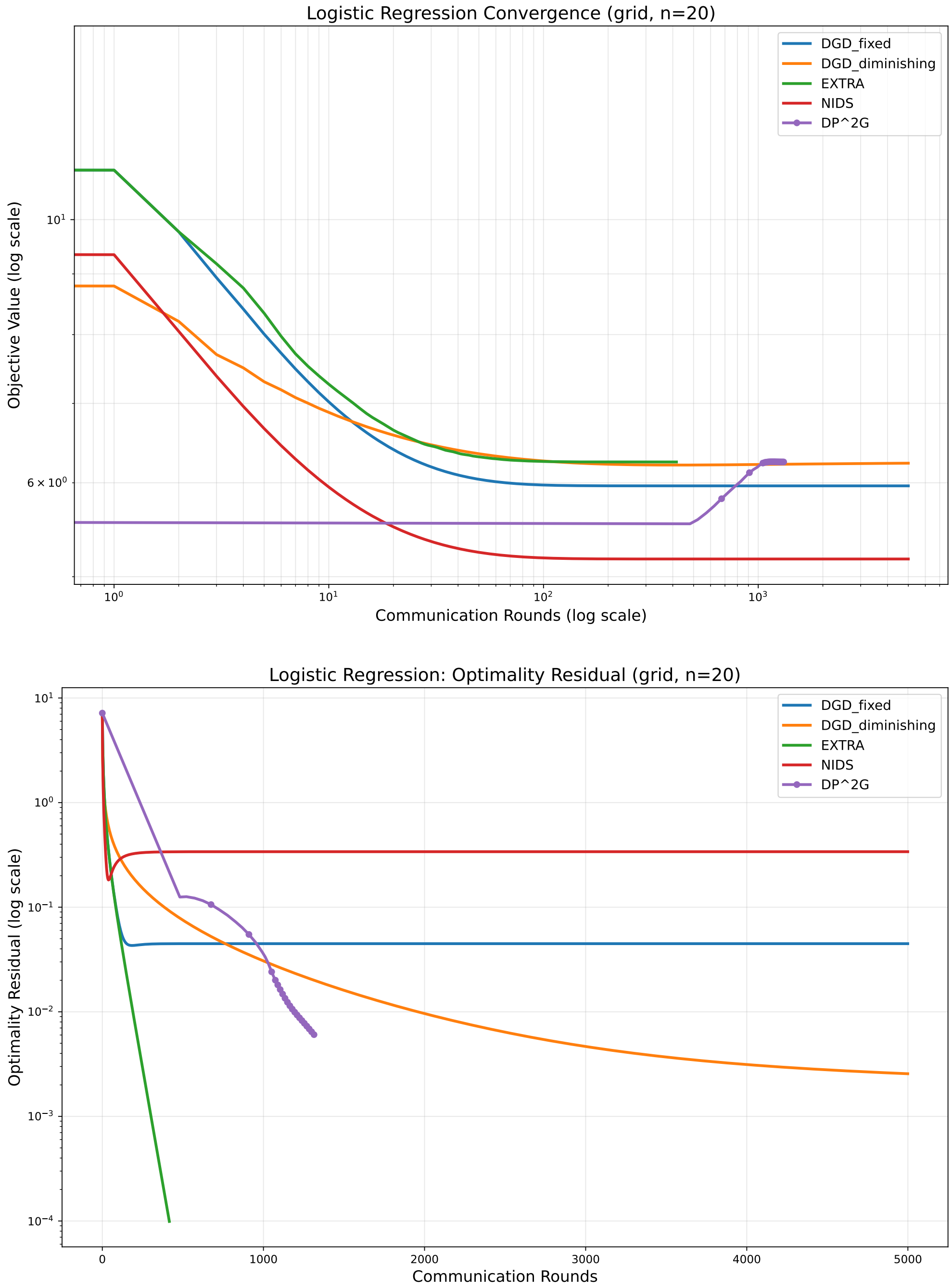

Figure 6: Logistic regression on the 4 × 5 grid: objective residual (top) and optimality residual (bottom). The milder connectivity narrows the gap between DP^{2}G and EXTRA while preserving the memory advantage.

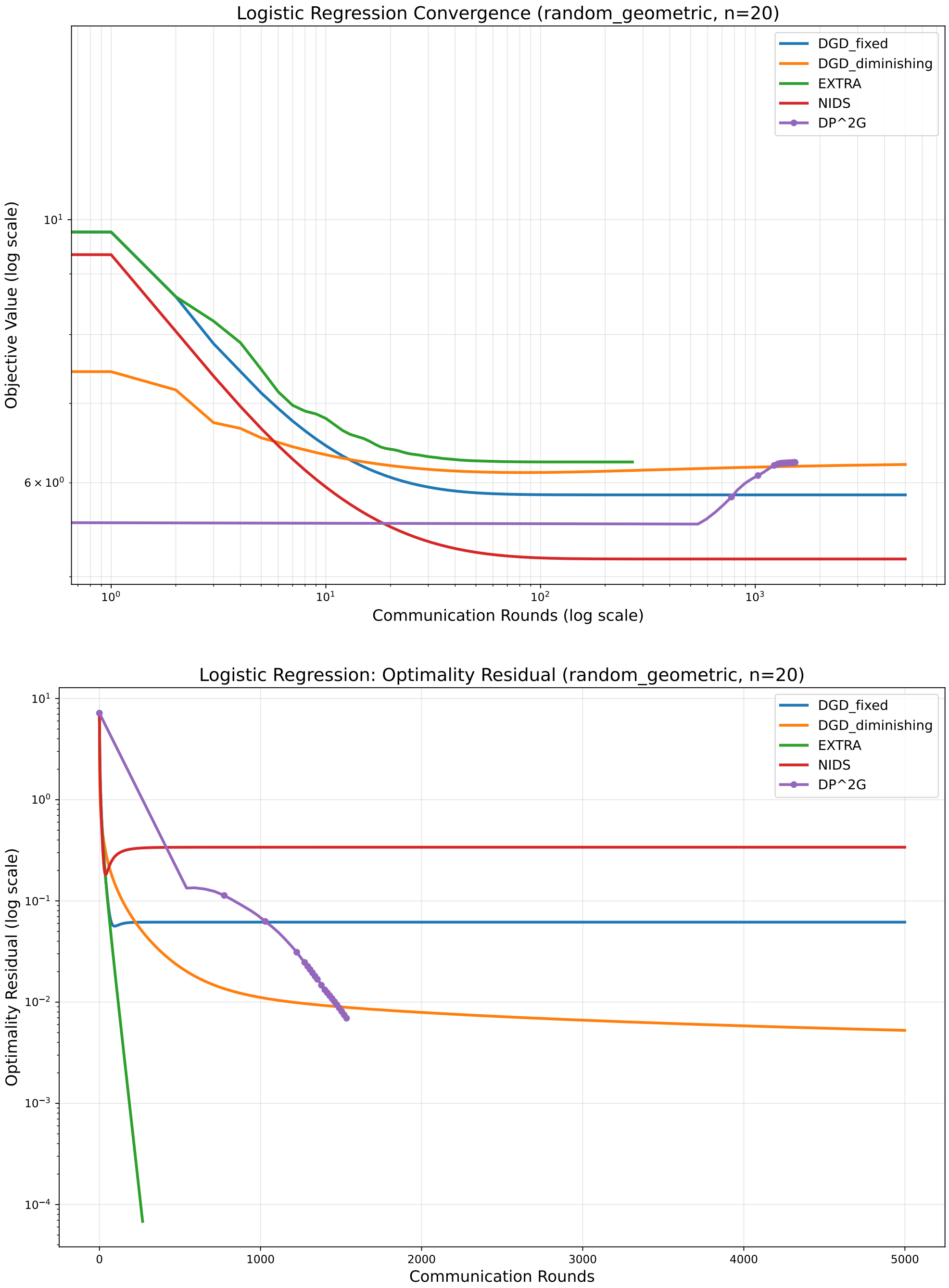

Figure 7: Logistic regression on the random geometric graph: objective residual (top) and optimality residual (bottom). Improved connectivity yields the fastest DP^{2}G decay among the logistic benchmarks.

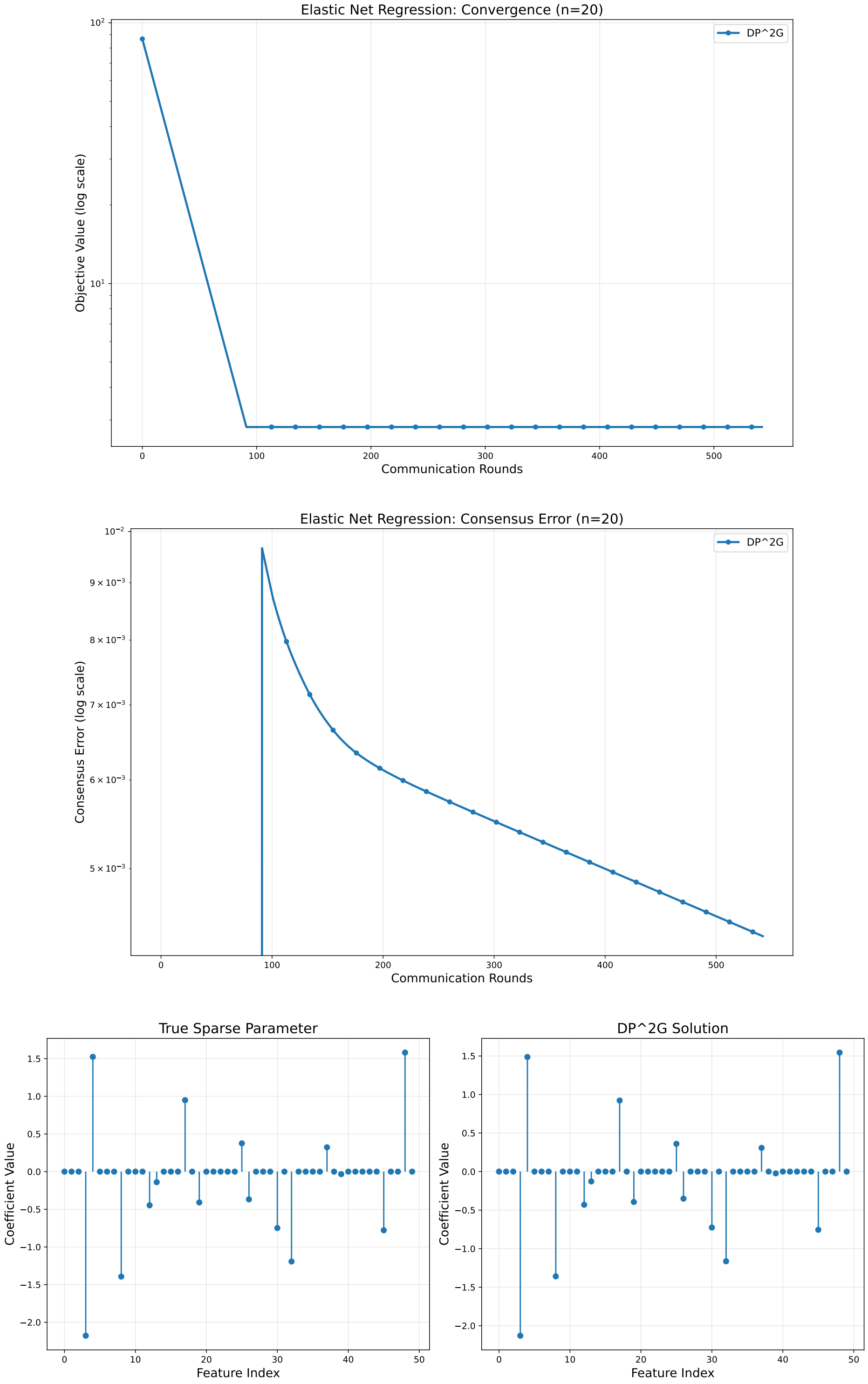

Figure 8: Elastic-net benchmark (random geometric graph, $n = 20$): objective residual (top), consensus error (middle), and recovered sparsity pattern (bottom).