A Decentralized Primal-Dual Framework for Non-convex Smooth Consensus Optimization

Gabriel Mancino-Ball, Yangyang Xu, and Jie Chen

Abstract—In this work, we introduce ADAPD, A DecentralAlized Primal-Dual algorithmic framework for solving non-convex and smooth consensus optimization problems over a network of distributed agents. ADAPD makes use of an inexact ADMM-type update. During each iteration, each agent first inexactiy solves a locally strongly convex subproblem and then performs a neighbor communication while updating a set of dual variables. Two variations to ADAPD are presented. The variants allow agents to balance the communication and computation workload while they collaboratively solve the consensus optimization problem. The optimal convergence rate for non-convex and smooth consensus optimization problems is established; namely, ADAPD achieves ε-stationarity in O(ε⁻¹) iterations. Numerical experiments demonstrate the superiority of ADAPD over several existing decentralized methods.

Index Terms—non-convex consensus optimization, decentralized optimization, primal-dual method, decentralized learning.

I. INTRODUCTION

Given a set of $N$ agents connected by an undirected network (graph) $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, N\}$ denotes the set of agents and $\mathcal{E} = \{(i, j) : \text{agent } i \text{ is connected to agent } j\}$ denotes the set of feasible communication paths among agents, consensus optimization methods solve the following problem using only local computation and local communication,

$$\min_{\mathbf{x} \in \mathbb{R}^N} f_{\mathbf{x}}(\mathbf{x}) \triangleq \frac{1}{N} \sum_{i=1}^{N} f_i(\mathbf{x})$$

where each $f_i : \mathbb{R}^p \to \mathbb{R}$ is a differentiable, potentially non-convex, cost function known only to agent $i$.

Problem (1) arises naturally in various scientific and engineering applications such as distributed machine learning/federated learning [1]–[3], decentralized matrix factorization [4], network sensing and localization [5]–[7], and multi-vehicle coordination [8], to name a few. The decision variable $\mathbf{x}$ can represent the weights of a neural network [1], the location of a particular object [8], or the state of a smart grid system [6], for example. Essentially, any scenario in which data is either too large or is protected for privacy reasons, fits problem (1).

A. Problem Formulation

The standard approach to solve problem (1) in a decentralized manner is to incorporate the graph structure, $G$, either as a constraint [6], [9]–[12] or directly into the algorithm iterations [13]–[17]; our formulation relies on the former. Let each agent maintain a local copy of the global decision variable $\mathbf{x}$, and denote it as $x_i$. Further, assume that any variable with a subscript $i$ belongs to agent $i$ and that $\mathcal{N}(i) \equiv \{j : (i, j) \in \mathcal{E}\}$ is the set of neighboring agents to agent $i$. Define

$$\mathbf{x} \triangleq [x_1 \ldots x_N]^\top \in \mathbb{R}^{N \times p}, \quad F(\mathbf{x}) \triangleq \frac{1}{N} \sum_{i=1}^{N} f_i(x_i),$$

and for notational convenience, by abuse of notation,

$$\nabla F(\mathbf{x}) \triangleq \frac{1}{N} [\nabla f_1(x_1) \ldots \nabla f_N(x_N)]^\top \in \mathbb{R}^{N \times p}$$

to be the concatenation of local decision variables, the objective function, and the gradient of the objective function, all written in matrix notation, respectively. The consensus optimization problem (1) can then be reformulated as

$$\min_{\mathbf{x}} F(\mathbf{x}) \text{ subject to } \mathbf{W} \mathbf{x} = \mathbf{X}$$

where $\mathbf{W} \in \mathbb{R}^{N \times N}$ is typically referred to as the mixing matrix [6], [12], [14] or the gossip matrix [10] and represents the underlying connectivity structure of the network $G$. As long as $G$ is connected, the following assumptions on $\mathbf{W}$ guarantee that problem (4) is equivalent to problem (1) and are standard in the literature [6], [10].

Assumption 1: The mixing matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ satisfies:

(i) (Decentralized property) $w_{ij} > 0$ if $(i, j) \in \mathcal{E}$, otherwise $w_{ij} = 0$,

(ii) (Symmetric property) $\mathbf{W} = \mathbf{W}^\top$,

(iii) (Null space property) null $(\mathbf{I} - \mathbf{W}) = \text{span}(\mathbf{e})$, where $\mathbf{e} \in \mathbb{R}^{N}$ is the vector of all ones, and

(iv) (Spectral property) the eigenvalues of $\mathbf{W}$ lie in the range $(-1, 1)$ and can be ordered as

$$-1 < \lambda_1(\mathbf{W}) \leq \cdots \leq \lambda_2(\mathbf{W}) < \lambda_1(\mathbf{W}) = 1.$$  

1) On the Choice of Mixing Matrices: Several common choices for mixing matrices are presented in [6]. For sake of completeness, we include some here.

- Laplacian-based constant edge weight matrix,

$$\mathbf{W} = \mathbf{I} - \frac{1}{\tau} \mathbf{L}$$

where $\mathbf{L}$ is the Laplacian matrix of $G$ and $\tau > \frac{1}{2} \lambda_1(\mathbf{L})$. Here, $\lambda_1(\mathbf{L})$ is the largest positive eigenvalue of $\mathbf{L}$. As per [6], if the eigenvalues of $\mathbf{L}$ are unknown, one can use $\tau = \max_{i \in \mathcal{V}} |\mathcal{N}(i)|^\varepsilon + \varepsilon$, for some $\varepsilon > 0$.

- Metropolis constant edge weight matrix, for some $\varepsilon > 0$,

$$w_{ij} = \begin{cases} \frac{1}{\max\{|\mathcal{N}(i)|, |\mathcal{N}(j)|\} + \varepsilon}, & (i, j) \in \mathcal{E}, \\ 0, & (i, j) \notin \mathcal{E} \text{ and } i \neq j, \\ 1 - \sum_{k \in \mathcal{V}} w_{ik}, & i = j. \end{cases}$$

This work was supported by the Rensselaer-IBM AI Research Collaboration, part of the IBM AI Horizons Network. J. Chen is supported in part by DOE Award DE-OE0000910.
• Symmetric fastest distributed linear averaging matrix, (FDLA), which is a matrix that achieves the fastest information diffusion through $G$ and is obtained by solving a semidefinite program $[18]$.

Several vital quantities for our analysis come from the spectral properties of $G$. We define

$$
\rho_2 \pm 1 - \lambda_2(W) \in (0, 1), \quad \rho_N \pm 1 + \lambda_N(W) \in (0, 2),
$$

and

$$
\rho_+ \triangleq \|W - \frac{1}{p}ee^\top\|_2 = \max \{\lambda_2(W), \lambda_N(W)\}
$$

where $\lambda_2(W)$ denotes the second largest eigenvalue of $W$, $\lambda_N(W)$ denotes the smallest eigenvalue of $W$, and $\rho_+$ is whichever of these quantities with a larger magnitude. The metric (8) is one way to measure the connectivity of $G$, where $\rho_+ \approx 0$ implies good connectivity. Now, under Assumption 1 and $\null(\sqrt{I - W}) = \null(I - W) = \text{span} (e)$, a further equivalent reformulation to (1) is

$$
\min_X F(X) \text{ subject to } \sqrt{I - WX} = 0,
$$

where $0 \in \mathbb{R}^{n \times p}$ is the matrix of all zeros. The framework proposed in this work uses the method of variable splitting $[19], [20]$ to effectively decouple the communication and the computation phases of traditional primal-dual methods used to solve (9). This is done by adding an extra variable (and constraint) to (9) to have the following problem:

$$
\min_{X, X_0} F(X) \text{ subject to } X = X_0, \sqrt{I - WX}X_0 = 0.
$$

This paper focuses on providing a framework for solving (10) by introducing dual variables and formulating an ADMM-type update $[20]$. We state our technical assumptions on the objective function in (10).

**Assumption 2:** The objective function $F$ in (10) satisfies:

(i) $F$ is $L$-smooth, i.e. there is $0 < L < \infty$ such that

$$
\|\nabla F(X) - \nabla F(Y)\|_F \leq L \|X - Y\|_F, \quad \forall X, Y \in \mathbb{R}^{n \times p}.
$$

(ii) $F$ is lower bounded, i.e. there is $\bar{f}$ such that

$$
-\infty < f \leq \bar{f}(X), \quad \forall X \in \mathbb{R}^{n \times p}.
$$

**Remark 1:** The assumptions (11) and (12) are standard in non-convex optimization. If each $f_i$ is $L_i$-smooth then $L \geq \max_i L_i$ and the lower boundedness assumption is equivalent to the existence of a minimizer of $F$.

Before proceeding, we provide a definition for stationary points of the problem (1). It is standard in the literature $[4], [12], [21]$. We then give a brief literature review.

**Definition 1 (\(\epsilon\)-stationary point):** A matrix $X \in \mathbb{R}^{n \times p}$ is called an $\epsilon$-stationary point of (1) if

$$
\|\frac{1}{n} \sum_{i=1}^N \nabla f_i(x)\|_2^2 + \|X - \bar{x}\|_F^2 \leq \epsilon
$$

where $\bar{x} \triangleq \frac{1}{N}ee^\top X$ is the average vector across the $N$ rows of $X$ and $\bar{X} \triangleq \frac{1}{n}ee^\top X$ is a matrix version of this same average.

**B. Related Works**

Distributed computing dates back decades ago to the seminal work $[22]$. As computational capabilities have increased dramatically during the proceeding years $[23]$, decentralized consensus optimization methods have grown in popularity due to their ability to adapt to various distributed architectures. Convex problems, i.e. where each $f_i$ is convex, have been studied extensively. DGD $[14]$ and the distributed subgradient method in $[13]$ have been shown to have sublinear convergence in the convex differentiable and convex non-differentiable settings, respectively. More robust methods like EXTRA $[6]$ and Acc-DNGD-SC $[16]$ rely on coupling a deterministic gradient computation with one (or multiple) communication step(s) at each iteration. These methods have linear convergence under the assumption that each $f_i$ has Lipschitz continuous gradient and either the global function $f$ is restricted strongly convex$^1$ or each $f_i$ is strongly convex, respectively. SSDA $[10]$ was designed for convex problems where the gradients of the Fenchel conjugates$^2$ of the objective functions $\{f_i\}$ are computable. Primal-dual methods such as decentralized ADMM $[9]$ and IDEAL $[11]$ incorporate the constraint $X - WX = 0$ from (4) directly into the local objective function to solve the max-min problem $\max_{\Lambda} \min_X F(X) + \langle \Lambda, X - WX \rangle + \frac{1}{\eta} \|X - WX\|_F^2$ where $\Lambda \in \mathbb{R}^{n \times p}$ and $\eta > 0$ is a penalty term for violating the consensus constraint. Our method draws inspiration from these primal-dual approaches.

Of particular interest to us are algorithms dealing explicitly with non-convex local cost functions, e.g. neural networks. The celebrated DGD $[15]$ has been shown to converge in the non-convex case using diminishing step-sizes. Adapting DGD to stochastic updates yields D-PSGD $[2]$ which is shown to have a convergence rate of $O\left(\frac{1}{K}\right)$ for very large $K$. Recent works such as $D^2$ $[24]$, DSGT $[25]$, and D-GET $[12]$ make use of stochastic gradient updates mixed with a gradient tracking scheme and draw inspiration from their non-stochastic and centralized counterparts $[6], [26], [27]$. Our proposed method is different from these methods in that the agents inexactly solve a local problem before communicating their updates. The advantage of the proposed method is its scalability to network structures with high-latency and low-bandwidth, which is an issue for methods that perform multiple communications during each update $[25], [28]$. Additionally, we note that our method is a framework for solving (1); thus standard solvers such as gradient descent, accelerated gradient descent, and stochastic gradient descent methods can all be employed by each agent to achieve the fastest convergence for the local subproblem. Two methods that are similar to our proposed method (in that a subproblem is solved at every iteration) are the NEXT $[7]$ and the Prox-PDA $[4]$ algorithms. The NEXT algorithm is robust in the sense that local agents solve a generic strongly convex subproblem, however, two communications must be performed during each iteration. The convergence rate of NEXT was shown in $[29]$ by viewing NEXT as a special case of the SONATA method. The Prox-PDA algorithm utilizes an augmented Lagrangian-type update where each agent first

---

$^1$A convex, differentiable function $h: \mathbb{R}^p \rightarrow \mathbb{R}$ is restricted strongly convex about a point $\bar{x}$ with parameter $\mu > 0$ if $\langle \nabla h(x) - \nabla h(\bar{x}), x - \bar{x} \rangle \geq \mu \|x - \bar{x}\|^2_F$ for all $x \in \mathbb{R}^p$.

$^2$The Fenchel conjugate of a convex function $h: \mathbb{R}^p \rightarrow \mathbb{R}$ is $h^*(y) \triangleq \text{sup}_x \langle x, y \rangle - h(x)$.
minimizes a strongly convex, primal local subproblem and then updates a dual variable. Prox-PDA makes use of an additional proximal term on the primal subproblem to ensure the local subproblem is strongly convex [4]. Our algorithm is closely related to Prox-PDA, however, we employ an ADMM-type update [20] that separates the communication and computation phases each agent must complete. Inspired by the results from [30], we anticipated that the ADMM-type updates would improve practical convergence without sacrificing theoretical guarantees; the results in Section IV confirm this.

Additional algorithms to consider are asynchronous algorithms that do not require a synchronous communication step and algorithms that use time-varying mixing matrices and/or mixing matrices that do not satisfy our assumptions in Assumption 1. Some prominent asynchronous algorithms include AD-PSGD [31], ARock [32], the Asynchronous Primal-Dual method in [33], and APPG [34]. Algorithms that handle different network structures from those considered here have also been considered: Push-Pull [17] handles directed graphs and DlGing [35] is a gradient tracking algorithm that works for network structures where \( W \) changes at every iteration. The goal of our future work is to extend our framework to handle network settings.

C. Summary of Contributions

Our main contributions are listed below:

- We propose ADAPD, a Decentralized Primal-Dual algorithmic framework for solving non-convex and smooth consensus optimization problems over a network of \( N \) agents. Our framework is based on performing inexact ADMM-type updates with the augmented Lagrangian function of problem (10). During each ADAPD iteration, each agent in the network inexactly solves a strongly convex local subproblem, followed by updating an additional local primal variable and two local dual variables, during which agents communicate with their neighbors.

- We present two variants to our framework: ADAPD-OG (ADAPD-One Gradient) and ADAPD-MC (ADAPD-Multiple Communications). ADAPD-OG changes the local strongly convex subproblem to involve a single gradient step. ADAPD-MC allows each agent to communicate multiple times with their neighbors. These variants allow for agents to optimize the balance between performing local computation and local communication.

- The optimal convergence rate for smooth, non-convex consensus optimization problems [21] is established for ADAPD and ADAPD-OG. Namely, we prove that ADAPD and ADAPD-OG converge to an \( \epsilon \)-stationary point, see (13), in \( O(\epsilon^{-1}) \) iterations. The convergence of ADAPD-MC follows directly from the convergence of ADAPD. For both ADAPD and ADAPD-OG, our analysis depends on defining a Lyapunov function that decreases with every iteration. We use this to show the convergence of each method.

- We compare ADAPD on several non-convex problems to state-of-the-art methods such as DGD [15], Prox-PDA [4], D-PSGD [2], DSCT [25], and D-GET [12]. Four experiments are conducted in total: two using deterministic gradients and two using stochastic gradients. In all cases, ADAPD demonstrates numerical superiority over the other state-of-the-art methods.

D. Notation

We use bold face letters such as \( \mathbf{X} \) and \( \mathbf{x} \) to denote a matrix and a vector, respectively. Let \( x_{ij} \) denote the element in the \( i^{th} \) row and \( j^{th} \) column of the matrix \( \mathbf{X} \). The Frobenius norm of a matrix is denoted \( ||\mathbf{X}||_F \), while the Euclidean norm of a vector is denoted \( ||\mathbf{x}||_2 \). Define the standard matrix inner product of \( \mathbf{A}, \mathbf{B} \in \mathbb{R}^{N \times p} \) to be \( \langle \mathbf{A}, \mathbf{B} \rangle \triangleq \sum_{i=1}^{N} \sum_{j=1}^{p} a_{ij} b_{ij} \). For a given symmetric matrix \( \mathbf{U} \in \mathbb{R}^{N \times N} \), we denote \( ||\mathbf{A}||_U^2 \triangleq \langle \mathbf{A}, \mathbf{U} \mathbf{A} \rangle \). If \( \mathbf{U} \) is positive definite, then \( ||\mathbf{A}||_U^2 \) defines a norm. For square matrices \( \mathbf{A}, \mathbf{B} \in \mathbb{R}^{N \times N} \), define the matrix inequality \( \mathbf{A} \preceq \mathbf{B} \) to hold if and only if \( \mathbf{B} - \mathbf{A} \) is positive semi-definite.

II. ADAPD Framework

To solve (10), we employ the augmented Lagrangian function with penalty parameter \( 0 < \eta < \frac{1}{2} \), which is

\[
L_\eta(X,Y,Z) = F(X) + \langle Y, X - X_0 \rangle + \frac{1}{2\eta} ||X - X_0||_F^2 + \langle Z, \sqrt{1 - W}X_0 \rangle + \frac{1}{2\eta} \sqrt{1 - W}X_0 ||_F^2
\]

(14)

with dual variables

\[
Y \triangleq [y_1 \ldots y_N]^T, \ Z \triangleq [z_1 \ldots z_N]^T \in \mathbb{R}^{N \times p}.
\]

The classic ADMM [20] approach for solving (10) performs the following updates using (14): \( X^{k+1} \approx \arg\min_X L_\eta(X;X_0^k;Y^k,Z^k) \) (16), \( X_0^{k+1} \approx \arg\min_{X_0} L_\eta(X^{k+1};X_0;Y^k,Z^k) \) (17), \( Y^{k+1} \approx Y^k + \beta_1 (X^{k+1} - X_0^{k+1}) \) (18), \( Z^{k+1} \approx Z^k + \beta_2 \sqrt{1 - W}X_0^{k+1} \) (19) where \( \beta_1, \beta_2 > 0 \) are the step sizes for the dual variables.

Notice that in practice, the exact minimizer of (16) is difficult to find; thus a natural alternative to (16) would be to perform an inexact update to the local decision variable as in [36]. This would lead to a computationally efficient way to solve the local subproblem (16) that fully utilizes local computing power without overburdening the agents. Further, notice that the optimal solution to (17) involves solving the linear system \( \frac{1}{2N}(2I - W)X_0 = \frac{1}{2N}X^{k+1} + Y^k - \sqrt{1 - W}Z^k \); it should be stated that \( (2I - W)^{-1} \) exists (by Assumption 1(iv)), however, it is not implemented in a decentralized fashion since \( W \) is not diagonal, so solving (17) would involve another iterative method, which may require many communications to find the exact minimizer. To remedy this, we apply a single gradient descent step to the \( X_0 \)-subproblem, which results in a communication efficient method to solve (10). Additionally, the \( Z \) update in (19) cannot be implemented in a decentralized manner as \( \sqrt{1 - W} \) in general will not preserve the underlying network topology. To account for these issues, we propose the following modifications to (16) - (19) and choose \( \beta_1 = \beta_2 = \frac{1}{\eta} \):

\[
X^{k+1} \approx \arg\min_X L_\eta(X;X_0^k;Y^k,Z^k) \quad \text{(20)}
\]

\[
X_0^{k+1} = \frac{1}{2} \left( WX_0^k + X^{k+1} + \eta (Y^k - \sqrt{1 - W}Z^k) \right) \quad \text{(21)}
\]

\[
Y^{k+1} = Y^k + \frac{1}{\eta} (X^{k+1} - X_0^{k+1}) \quad \text{(22)}
\]
As mentioned above, \( \sqrt{1 - W} \) cannot be implemented in a decentralized manner. To overcome this issue, we make the following remark.

Remark 2: Notice that if \( Z_0 \in \text{range}(\sqrt{1 - W}) \), e.g. \( Z_0 = 0 \), then \( Z_k \in \text{range}(\sqrt{1 - W}) \), for all \( k \geq 0 \) from (19). In this case, the update in (19) is equivalent to that in (23) and is thus further equivalent to

\[
Z^{k+1} = Z^k + \frac{\eta}{2} (I - W) X_0^{k+1},
\]

where \( \tilde{Z}^k = \sqrt{1 - W} Z^k \) for all \( k \). Update (24) can be implemented in a fully decentralized manner.

Rearranging (24) with \( k \to 1 \) yields

\[
WX_0^1 = \eta \left( \frac{1}{2} X_0^1 + Z^{k-1} - Z^k \right).
\]

Combining (21), (24), and (25) shows that the \( X_0 \) update in (21) can actually be performed with no communication between the agents for all \( k > 1 \). This implies that except for the first iteration, our framework (20) - (23) requires only one multiplication by \( W \) and hence only one communication among agents (for networks where multiple communications are permitted, see Section II-A and the discussion that follows).

To track the error in the approximate solution of (20) at every step, we require that the cumulative error is summable. More specifically, we require

\[
\| R^{k+1} \|^2_F \leq \epsilon_{k+1}, \quad \text{with} \quad R^{k+1} \equiv \nabla F(X^{k+1}) + Y^k + \frac{1}{2\eta} (X^{k+1} - X_0^0), \quad \forall k \geq 0,
\]

and

\[
\epsilon \triangleq \sum_{k=0}^{\infty} \epsilon_{k+1} < \infty.
\]

Remark 3: Similar to the results in [37], we can require,

\[
\mathbb{E} \left[ \| R^{k+1} \|^2_F \right] \leq \epsilon_{k+1} < \infty, \quad \forall k \geq 0
\]

and the theoretical results are not significantly affected. This allows for stochastic solvers to be used by each local agent. The optimality conditions (26) and (28) are achievable for properly chosen \( \eta > 0 \) since the local subproblems

\[
\min_{x_i} f_i(x_i) + \frac{1}{2\eta} \left\| y_i - x_i - x_0, i \right\|_2^2, \quad \forall i = 1, \ldots, N,
\]

are strongly convex when \( \eta < \frac{1}{L} \); thus (26) and (28) are just formalizing the inexact updates. From an agent’s point of view, (20) - (23) can be summarized in Alg. 1 below.

By Remark 2, we can obtain a unique sequence \( \{ Z^k \}_{k=1}^K \) in range(\( \sqrt{1 - W} \)) from the generated \( \tilde{Z} \)-sequence. Therefore, without causing confusion, we can use the corresponding \( Z \)-sequence in our analysis. Notice that our framework is sufficiently flexible to allow each agent to use different local subroutines to solve (29). In networks where the computing power of the agents differs vastly (see, e.g. [1]), this flexible framework allows for agents with higher compute capabilities to fully utilize their compute power whereas agents with lower compute capabilities are not expected to utilize heavy optimization tools to solve their local subproblem. We now describe two variants/modifications to Alg. 1 that can be employed if the computation constraints and/or the communication constraints are relaxed.

A. Framework Variants

1) Computation Variant: In scenarios where agents may face a lack of computational resources to solve (29), it may be inefficient to compute \( \nabla f_i(.) \) many times. To remedy this, we propose ADAPD-OG (One Gradient), which requires each agent to only compute a single gradient during every iteration. The \( X \) update at iteration \( k \) becomes

\[
X^{k+1} = X^k - \eta \left( \nabla F(X^k) + Y^k \right)
\]

which requires only a single local gradient computation. Alg. 2 displays the pseudocode of ADAPD-OG.

Algorithm 2: ADAPD-OG (agent view)

\[
\text{Input: } X^0, X_0^0, Y^0, Z^0 = \sqrt{1 - W} Z^0 \text{ with } \quad Z^0 \in \text{range}(\sqrt{1 - W}), \quad K, \eta > 0.
\]

\[ \begin{align*}
1 & \quad \text{for } k = 0, \ldots, K - 1 \text{ do} \\
2 & \quad \quad \text{for } i = 1, \ldots, N \text{ in parallel do} \\
3 & \quad \quad \quad \text{Update } x_i \text{ until } \| x_i^{k+1} \|_2^2 \leq \epsilon_{k+1} \text{ with} \\
4 & \quad \quad \quad \quad x_i^{k+1} \leftarrow x_i^k - \eta \left( \nabla f_i(x_i^k) + y_i^k \right) \\
5 & \quad \quad \quad \quad \text{if } k = 0 \text{ then} \\
6 & \quad \quad \quad \quad \quad \text{Update } x_i \text{ until } \| x_i^{k+1} \|_2^2 \leq \epsilon_{k+1} \text{ with} \\
7 & \quad \quad \quad \quad \quad \quad x_i^{k+1} \leftarrow x_i^k + \frac{1}{2} \left( \sum_{j \in N_i \cup \{i\}} w_{ij} x_{j, i}^k + y_i^{k+1} + \eta (y_i^{k} - z_i^{k}) \right) \\
8 & \quad \quad \quad \quad \quad \quad \text{else} \\
9 & \quad \quad \quad \quad \quad \quad \text{Update } x_i \text{ until } \| x_i^{k+1} \|_2^2 \leq \epsilon_{k+1} \text{ with} \\
10 & \quad \quad \quad \quad \quad \quad \quad x_i^{k+1} \leftarrow x_i^k + \frac{1}{2} \left( \sum_{j \in N_i \cup \{i\}} w_{ij} x_{j, i}^{k+1} + y_i^{k} - z_i^{k} \right)
\end{align*}
\]

In scenarios where consensus is the main bottleneck for fast convergence, it may be practical to allow agents more than one communication during each ADAPD update. We denote the following multiple communication modification (either to Alg. 1 or Alg. 2) with appending an “-MC” (Multiple Communications) to the algorithm name.
2) Communication Variant: As stated in the introduction, our analysis depends on the values of $\rho_2$ and $\rho_N$ in (7), which jointly affect the value of $\rho_*$ that measures how quickly an average value can be computed in a decentralized manner. In a centralised computational paradigm, where each agent is allowed to communicate with all other agents either directly or via a central server, the mixing matrix $W$ can be replaced by the averaging matrix $\frac{1}{N}ee^T$. In this instance, $\rho_2 = \rho_N = 1$ and $\rho_* = 0$, which can lead to the fastest convergence for our algorithm (see the constant $C > 0$ in Theorem 1 and its corresponding effect on the step-size $\eta > 0$) in both theory and practice. However, by Assumption 1(i), the communication pattern of the agents is limited to performing only neighbor communications.

In scenarios where consensus is the main bottleneck for convergence (i.e., $\rho_*$ is very close to one), it may be beneficial to allow agents more than one communication during each ADAPD update. We denote the multiple communication modification (either to Alg. 1 or Alg. 2) by appending an “-MC” (Multiple Communications) to the algorithm name.

One straightforward modification is to replace $W$ by $W^R$ ($R$ denotes a power, not an iteration number here) for the $Z$ update in (24) and the computation of $X^0_i$, where $R \geq 1$ is an integer number. Notice that $W^R$ satisfies all the three properties in Assumption 1(ii)-(iv). Thus all our theoretical results hold for this MC modification. Since $\rho_*(W^R) := \|W^R - \frac{1}{N}ee^T\|_2 = \|W - \frac{1}{N}ee^T\|_2 = \rho_*(W)$, this MC modification can lead to a smaller $\rho_*$ if $R > 1$. However, if $\rho_*(W)$ is very close to one, $R$ needs to be very large in order to push $\rho_*(W^R)$ to zero. For this case, more efficient methods have been given in the literature for decentralized averaging [38], [39]. We employ the accelerated method in [38] and call it FastMix. The pseudocode is shown in Alg. 3. While FastMix is called at iteration $k$ of ADAPD-MC or ADAPD-OG-MC, the input $A^0$ will be $X^{k+1}$.

**Algorithm 3: FastMix**

**Input:** $A^0 = A^{-1}$, $W$, $R$.
1. Compute the step-size, $\alpha_W = \left(1 - \sqrt{1 - \rho_2^2} \right)/(1 + \sqrt{1 - \rho_2^2})$.
2. for $r = 0, \ldots, R$ do
3. \[ A^{r+1} = (1 + \alpha_W)WA^r - \alpha_W A^{r-1} \]
4. end

**Output:** $A^R$

The following lemma shows that the properties in Assumption 1(ii)-(iv) still hold for the operator used in FastMix and thus our theoretical results hold for ADAPD-MC and ADAPD-OG-MC. The proof of this lemma is located in the Supplementary Material.

**Lemma 1:** The output of Alg. 3 can be represented as $A^R = \mathcal{P}(W, R) A^0$, where $\mathcal{P}(W, R)$ is a polynomial of $W$ and satisfies the properties in Assumption 1(ii)-(iv).

We note that employing Alg. 3 is only feasible if either: (1) the communication pattern is so sparse that consensus error is the main bottleneck for convergence, or (2) communication cost is low relative to the computation cost, meaning that agents can communicate faster than they can compute. In practice, it is suggested that agents find a balance that distributes work evenly between communication and computation.

### III. Theoretical Guarantees

Our theoretical analysis draws from decentralized analytical methods such as [4], [15] and classical non-convex ADMM analyses, as in [30]. We first show the change in the augmented Lagrangian function value after one ADAPD iteration, i.e. (20) - (23), then we define a Lyapunov function and use this to show convergence. A crucial quantity for our analysis is

\[
V_k^0 \pm \left(X_{0}^{k+1} - X_{0}^k \right) - \left(X_{0}^{k} - X_{0}^{k-1} \right)
\]  

(31)

which measures the difference in the difference of successive $X_0$ updates. We define $X_0^{k+1} \equiv X_0^k$, to ensure that $V_0^k$ is defined for all $k \geq 0$. In the convergence analysis of Alg. 1, we will make consistent use of the following fact.

**Fact 1 (Peter-Paul and Young’s Inequality):** For any $\delta > 0$, any integer $m$, and matrices $A, B, \{A_i\}_{i=1}^m$ of appropriate sizes,

\[
\langle A, B \rangle \leq \frac{\delta}{2} \|A\|_F^2 + \frac{1}{2\delta} \|B\|_F^2.
\]

(32)

\[
\| \sum_{i=1}^m A_i \|_F^2 \leq m \| \sum_{i=1}^m A_i \|_F^2.
\]

(33)

#### A. Convergence Results of ADAPD

The first step in the analysis creates an equivalence expression among the dual and primal variables. All relevant supporting Lemmas and proofs are located in Appendix B.

**Lemma 2:** For all $k \geq 0$, the dual variables in (22) and (23) can be expressed as

\[
\sqrt{1 - WZ} = Y_k - \frac{1}{2}W(X_k^0 - X_k^{k-1})
\]

(34)

\[
Y_k = R - \nabla F(X_k) - \frac{1}{\eta}(X_k^0 - X_k^{k-1}).
\]

(35)

Next, we characterize the change of the augmented Lagrangian function value after one ADAPD iteration.

**Lemma 3:** Let $\{(X_k^0, X_k^k, Y_k, Z_k)\}$ be obtained from Alg. 1 or equivalently by updates (20)-(23) such that (26) holds. If $\eta < \frac{1}{L_S}$, then it holds for all $k \geq 0$ that

\[
\begin{align*}
\mathcal{L}_{\eta}(X_k^{k+1}, X_k^{k+1}, Y_k^{k+1}, Z_k^{k+1}) - \mathcal{L}_{\eta}(X_k^0, X_k^0, Y_k, Z_k) \\
\leq \frac{4L_S\eta}{2\eta} \|X_k^{k+1} - X_k^0\|_F^2 + \frac{4\eta}{\rho_2} \|Y_k^{k+1}\|_F^2 \\
+ \eta \|Y_k^{k+1} - Y_k^k\|_F^2 + \eta \|Z_k^{k+1} - Z_k^k\|_F^2.
\end{align*}
\]

(36)

Notice that the inequality in (36) does not imply the non-increasing monotonicity of the augmented Lagrangian function at the generated iterates. Below, we bound the dual variable change by the primal variable change and the $V_0^k$ term. Then we establish another inequality and add it to (36) to build a non-increasing Lyapunov function.

**Lemma 4:** Under the assumptions of Lemma 3, it holds that for all $k \geq 0$,

\[
\begin{align*}
\eta \|Y_k^{k+1} - Y_k^k\|_F^2 &\leq 4L_S\eta \|X_k^{k+1} - X_k^k\|_F^2 + \frac{4\eta}{\rho_2} \|V_0^{k+1}\|_F^2 + 8\eta\epsilon_k, \\
\eta \|Z_k^{k+1} - Z_k^k\|_F^2 &\leq \frac{8L_S\eta}{\rho_2} \|X_k^{k+1} - X_k^k\|_F^2 + \frac{10}{\rho_2\eta} \|V_0^k\|_F^2 + \frac{16\eta}{\rho_2} \epsilon_k.
\end{align*}
\]

(37)

(38)
where $V_h^k$ is defined in (31).

Lemma 5: For all $k \geq 0$, the following relation holds

$$\frac{1}{2\eta} \left( \left\| \nabla f - Wx_{k+1}^0 \right\|_F^2 + \left\| \nabla f - W(x_{k+1}^0 - x_h^k) \right\|_F^2 \right) - \frac{1}{2\eta} \left\| \nabla f - Wx_h^k \right\|_F^2 + \frac{1}{2\eta} \left\| x_h^{k+1} - x_h^k \right\|_W^2 - \frac{1}{2\eta} \left\| x_h^k - x_h^{k-1} \right\|_W^2 \leq (L - \frac{1}{2\eta}) \left\| x_h^{k+1} - x_h^k \right\|_F^2 + \frac{1}{2\eta} \left\| x_h^{k+1} - x_h^{k-1} \right\|_F^2 \leq (L - \frac{1}{2\eta}) \left\| x_h^{k+1} - x_h^k \right\|_F^2 + \frac{1}{2\eta} \left\| x_h^k - x_h^{k-1} \right\|_F^2 \tag{39}$$

where $V_h^k$ is defined in (31).

Using Lemmas 4 and 5, we are ready to build a non-increasing Lyapunov function as follows.

Lemma 6: Let $\{(X^k, X_h^0, Y^k, Z^k)\}$ be obtained from Alg. 1 or equivalently by updates (20)-(23) such that (26) holds. If $\eta < \frac{1}{2L}$, then

$$L_\eta(x_h^k, x_h^{k+1}, Y^{k+1}, Z^{k+1}) + \frac{C_1}{2\eta} \left\| \nabla f - Wx_h^0 \right\|_F^2 + \frac{C_1}{2\eta} \left\| x_h^{k+1} - x_h^k \right\|_F^2 \leq L_\eta(x_h^k, x_h^k, Y^k, Z^k) + \frac{C_1}{2\eta} \left\| \nabla f - Wx_h^0 \right\|_F^2 + \frac{C_1}{2\eta} \left\| x_h^k - x_h^{k-1} \right\|_F^2 + \left( \frac{8L^2(p + \eta)K^2}{2\eta - \eta_2} \right) \left\| X^{k+1} - X^k \right\|_F^2 + \left( \frac{16L_p^2(1 + \eta)K^2}{2\eta - \eta_2} \right) \left\| X^{k+1} - X^k \right\|_W^2 \tag{40}$$

for all $k \geq 0$, where $C \geq \frac{20sF^2}{p + \eta}$ is a fixed constant.

For the same, fixed $C \geq \frac{20sF^2}{p + \eta}$ as used in Lemma 6, define the Lyapunov function:

$$\Phi^k \leq L_\eta(x, x_h^k, Y^k, Z^k) + \frac{C_1}{2\eta} \left\| \nabla f - Wx_h^0 \right\|_F^2 + \frac{C_1}{2\eta} \left\| x_h^k - x_h^{k-1} \right\|_F^2 \tag{41}$$

We show the lower boundedness of this Lyapunov function in the following proposition and use this to obtain the convergence of Alg. 1.

Proposition 1: Under Assumptions 1 and 2, let $\{(X^k, X_h^0, Y^k, Z^k)\}$ be obtained from Alg. 1 or equivalently by updates (20)-(23) such that (26) and (27) hold. Choose $\hat{C}$ and $\eta$ such that

$$C \geq \frac{20sF^2}{p + \eta N} \quad \text{and} \quad \eta < \min \left\{ \frac{1}{2C}, \frac{1}{2CL} \left[ \frac{p \left( \left( (C+2) + \frac{(C+2)^2 + 32s^2}{4L_p^2} \right) \right) - \left( \frac{(C+2)^2 + 32s^2}{4L_p^2} \right)}{16L(p + 2)} \right] \right\} \tag{42}$$

Then the Lyapunov function (41) is uniformly lower bounded. More specifically, for all $k \geq 0$,

$$\Phi^k \geq \Phi \geq f - \frac{p + (32L + 16L_p)}{2L_p^2} \sum_{k=0}^{\infty} \epsilon_k - 1 > -\infty \tag{43}$$

where we take $\epsilon_0 = \epsilon_1$ and $f$ is defined in Assumption 2.

We are now in position to prove the convergence rate results of ADAPD.

Theorem 1: Under the same conditions assumed in Proposition 1, it holds that

$$\frac{1}{2L} \sum_{k=0}^{K-1} \left\{ \left\| X^{k+1} - X^k \right\|_F^2 + \left\| X^k - X_h^0 \right\|_F^2 \right\} \leq \frac{\Delta_x}{K} + \frac{(32L + 16L_p + (4C + 2))}{2L_p^2} \sum_{k=0}^{K-1} \epsilon_k, \tag{44}$$

where $\Delta_x = \delta^0 - \delta$ and

$$C_1 = \min \left\{ \frac{\rho_1 (C+2)(L_p - (8L + 16L_p))}{2L_p^2} \eta^2, \frac{1 - 2CL\eta}{2\eta} \right\}. \tag{45}$$

Theorem 2 (Convergence of ADAPD): Under the same conditions assumed in Proposition 1, it holds

$$\frac{1}{K} \sum_{k=0}^{K-1} \left\{ \left\| \nabla f(x_h^k) \right\|_F^2 + \left\| X^k - X_h^k \right\|_F^2 \right\} \leq \left( \frac{(2L^2 + 12) \Delta_1}{K_1} \right) + \left( \frac{(192L^2 + 96)\rho_2}{K_1 \rho_2} \right) \sum_{k=0}^{K-1} \epsilon_k \tag{46}$$

where $C_1$ is defined in (45), $C_2 \leq \max \left\{ \frac{48L\rho_2^2 \eta_1}{(1 + \rho_2)^2}, \frac{208L\rho_2^2 \eta_1}{(1 + \rho_2)^2} \right\}$, $C_3 \leq \frac{(32L + 16L_p + (4C + 2)) \rho_2}{2L_p^2}$, $C_4 \leq \frac{16s}{\eta}$, $X_h^k \leq \frac{1}{N} \sum_{i=1}^{N} X_i^k$, and $X_h^k \leq 10^{-7} \eta X^k$.

Remark 4: Let $k_0 = \arg\min_{1 \leq k \leq K} \left\{ \left\| \nabla f(x_h^k) \right\|_F^2 + \left\| X_h^k - X_h^0 \right\|_F^2 \right\}$. Then $\left\| \nabla f(x_h^{k_0}) \right\|_F^2 + \left\| X_h^{k_0} - X_h^0 \right\|_F^2 = O \left( \frac{1}{\eta} \right)$. Hence, in order to produce an $\epsilon$-stationary point as defined in Definition 1, we need $K = O \left( \frac{1}{\epsilon} \right)$ iterations. Furthermore, notice that all the problems in (29) are smooth and strongly convex. The steepest gradient method has linear convergence to solve such problems. Hence, to produce $X_h^{k+1}$ as a $\frac{C_2}{K}$-accurate solution of the problem in (29), it needs $O \left( \log \left( \frac{N}{\epsilon_k+1} \right) \right)$ gradient evaluations for each $i = 1, \ldots, N$. Choose $\epsilon_k = \frac{1}{1 + \frac{1}{K\rho_2}}$ for all $k \geq 0$ and for some $p > 1$, then $\{\epsilon_k\}$ is summable, and the total gradient evaluations to produce an $\epsilon$-stationary point of (1) would be $\sum_{k=0}^{K-1} O \left( \log (N(k + 1))^p \right) = O \left( \frac{1}{\epsilon^p} \right)$.

B. Convergence Results of ADAPD-OG

The convergence rate results of the ADAPD-OG follow the same logic as the results for ADAPD, hence all supporting Lemmas and proofs are located in the Supplementary Material. Notice that (35) is no longer a valid relation when Alg. 2 is used. Instead, we have the following from (22) and (30):

$$Y^k = -\nabla F(X^k - 1) - \frac{1}{\eta} \left( X_h^0 - X_h^{k-1} \right), \quad \forall k \geq 0. \tag{46}$$

As in the analysis for ADAPD, we define $X_h^{k-1} \pm X_h^0$ and further define $X_h^0 \pm X_h^0$. We have the following result.

Theorem 3 (Convergence of ADAPD-OG): Under Assumptions 1 and 2, let $\{(X^k, X_h^0, Y^k, Z^k)\}$ be obtained from Alg. 2 or equivalently by (30) and (21)-(23). Choose $\hat{C}$ and $\eta$ such that

$$\hat{C} \geq \frac{12sF^2}{p + \eta N} \quad \text{and} \quad \eta < \min \left\{ \frac{1}{2C}, \frac{1}{2CE} \left[ \frac{p \left( \left( (C+2) + \frac{(C+2)^2 + 32s^2}{4L_p^2} \right) \right) - \left( \frac{(C+2)^2 + 32s^2}{4L_p^2} \right)}{16L(p + 2)} \right] \right\}. \tag{47}$$
Then, it holds
\[
\frac{1}{K} \sum_{k=0}^{K-1} \left( \| \nabla f(x_{k+1}) \|_2^2 + \| x_{k+1} - x_k \|_2^2 \right) \leq \frac{(2L^2 + 1)C_1 + C_2 ) \Delta_k}{C_1 K}
\]
where \( C_1 \leq \min \{ \frac{48L^2 \eta^2 + 112}{(1-\rho)^2}, \frac{1}{\eta^2} \}, \) \( C_2 \leq \frac{\Delta_k}{\eta^2}, \Delta_k = \Phi^0 - f + 1, \) and \( \frac{1}{N} \sum_{i=1}^{N} x_i^k \) and \( x_i^k \) are given by
\[
\frac{1}{N} \sum_{i=1}^{N} x_i^k \quad \text{and} \quad x_i^k = \frac{1}{N} e e^T X^k.
\]

IV. NUMERICAL EXPERIMENTS

We test our proposed methods on several non-convex problems: (1) a binary classification problem using logistic regression with a non-convex regularizer, (2) a multi-target cooperative localization problem, and (3) two image classification problems using convolutional neural networks. The experiments serve to verify both the flexibility of our methods, as well as their numerical superiority over other decentralized optimization methods.

For experiments (1) and (2), we compare our methods to DGD with a diminishing step-size [15] and the single gradient version of Prox-PDA, called Prox-GPDA [4]. Additionally, for Alg. 1 we fix \( \nu \) the \( \nu \)-version of Prox-PDA, called Prox-GPDA [4]. Furthermore, for DGD with a diminishing step-size [15] and the single gradient optimization methods.

For experiment (3), we compare to D-PSGD [2], DSGT [25], and D-GET [12]. For all experiments, we fix a set of penalty parameters/step-sizes and optimize each algorithm over this set, choosing whichever penalty/step-size performs the best. For all methods besides Prox-GPDA, we use the same mixing matrix, which will be described in each subsection below. For Prox-GPDA, we take \( W \) to be the formulation as given in [4] (see equation (23) and the discussion that follows).

A. Non-Convex Regularized Logistic Regression

We consider the non-convex decentralized binary classification problem [4], [37]. Utilizing a logistic regression formulation, the local agent cost functions are given by,
\[
f_i(x_i) = \frac{1}{m_i} \sum_{j=1}^{m_i} \log \left( 1 + \exp(-b_j \langle x_i, a_j \rangle) \right) + \sum_{d=1}^{D} \frac{\alpha_i |x_i[d]|^2}{1 + |x_i[d]|^2},
\]
where \( x_i[d] \) denotes the \( d \)th component of the vector \( x_i \). Given a set of data \( \{ (a_j, b_j) \}_{j=1}^{m_i} \) for all \( i = 1, \ldots, N \), where \( b_j \in \{-1, +1\} \) denotes a particular class label, (48) can be used to perform binary classification and the non-convex regularizer, \( \sum_{d=1}^{D} \frac{\alpha_i |x_i[d]|^2}{1 + |x_i[d]|^2} \) helps to induce sparsity on the solutions. We use the a9a dataset [41], [42] which consists of 32,561 training data points and 16,281 testing data points. Each data point, \( a_j \in \mathbb{R}^{123} \) contains numerical features about adults from the 1994 Census database and \( b_j \) indicates whether or not the adults earn more or less than \( \$50,000 \) per year. We fix \( N = 50 \) for this experiment and simulate agent connectivity in two ways: (1) using a ring-structured graph and (2) using a random Erdős Rényi graph, with connection probability equal to 0.3 (i.e. each agent is connected to roughly 15 other agents).

For the ring-structured graph, we choose \( W \) to be
\[
w_{ij} = \begin{cases} 
\frac{1}{4}, & i = j, \\
\frac{1}{4}, & (i, j) \in E \text{ and } i \neq j, \\
0, & \text{otherwise},
\end{cases}
\]
and for the random Erdős Rényi graph, we use the Laplacian-based constant edge weight matrix from (5). We vary \( \alpha \in \{0.01, 1.0\} \) to study the effect that the non-convex term has on each agent’s local subproblem. For all runs, we fix the communication budget to 500 neighbor communications. Additionally, we compare ADAPD-MC and ADAPD-OG-MC to the other methods. We perform 20 iterations of the FastMix Alg. 3 during every outer iteration of Alg. 1 for ADAPD-MC and 10 iterations for ADAPD-OG-MC. This means we only compute 50 gradients for ADAPD-OG-MC, to keep with the 500 communication budget. We report the \( \epsilon \)-stationarity violation (13) for the following four scenarios: (i) the random Erdős Rényi graph with \( \alpha = 1.0 \), (ii) the random Erdős Rényi graph with \( \alpha = 0.01 \), (iii) the ring-structured graph with \( \alpha = 1.0 \), and (iv) the ring-structured graph with \( \alpha = 0.01 \).

The value of \( \alpha \) directly affects the \( L \)-smooth value for this problem: a larger \( \alpha \) leads to a larger \( L \), thus making the problem more challenging. Scenarios (i) and (ii) compare the performance of these methods when the communication pattern is not a bottleneck in achieving \( \epsilon \)-stationarity. It is clear that when the problem is more challenging (Figure 1, far left), utilizing more gradient computations leads to faster convergence. When the problem is less challenging, the ADAPD-OG variant can perform just as well as ADAPD, but it utilizes almost half of the gradient computations (Figure 1, second from left). This suggests that assessing the level of difficulty of the problem can guide agents to use more, or less, gradients. Scenarios (iii) and (iv) compare the performance of these methods when the communication pattern is a bottleneck in achieving \( \epsilon \)-stationarity. Here, we notice that performing multiple communications during each ADAPD (or ADAPD-OG) iteration leads to a smoother convergence curve (see Figure 1, two rightmost images). Figure 1 (two rightmost images) also demonstrates that performing less gradient computations and more communications does not adversely affect the performance of our methods. Thus ADAPD and its variants provide agents with the flexibility to choose whether more gradient computations or more neighbor communications are necessary to solve the consensus problem. Finally, notice that in all four scenarios, ADAPD and its variants from Section II-A outperform both DGD and Prox-GPDA.

B. Multi-Target Cooperative Localization

Multi-target cooperative localization is a target locating problem [5]: given only a noisy distance metric, can \( N \) agents locate \( N \) common targets? Let \( \{ \omega_i \}_{i=1}^{N} \) be a set of locations of the agents, i.e. \( \omega_i \in \mathbb{R}^{2} \). Then the local objective function for each agent is given by
\[
f_i(x_i) = \frac{1}{2} \sum_{i=1}^{N} \left( \xi_{i,i} - \| x_i[t] - \omega_i \|_2^2 \right)^2
\]
where \( \xi_{i,i} \) is a random variable that represents a noisy distance metric, and \( x_i = [x_i[1]^T \ldots x_i[N_T]^T]^T \in \mathbb{R}^{2N_T} \) is
stacking of the vectors \( \{ x_i[t] \}_{i=1}^{N_T} \). Note that (49) is indeed non-convex, but it is not globally \( L \)-smooth for any \( L \geq 0 \). However, we still find this problem is valuable to test our methods. Denote the true targets as \( x^* \); these are used to generate \( \xi_i[t] \) for all \( i \) and \( t \) by computing

\[
\xi_i[t] = \| x^* - \omega_i \|^2 + \epsilon_i[t]
\]

where \( \epsilon_i[t] \) is drawn from a normal distribution with mean 0 and variance \( \sigma^2 > 0 \). For all of our experiments, \( \sigma^2 = 0.01 \). We simulate agent connectivity by randomly generating \( N = 50 \) agents in \([-1, 1] \times [-1, 1] \) grid and creating an edge between agents if the Euclidean distance between them is less than or equal to 0.3. Each coordinate in the targets \( x^* \) is drawn independently from a normal distribution with mean 0 and variance 0.1. Figure 2 shows the connectivity of the agents, as well as an example of target locations.

For this example, \( W \) is chosen to be the Laplacian-based constant edge weight matrix from (5). We randomly generate \( N_T = 5 \) targets and limit the communications to be 500 for all algorithm runs, with all methods starting from the same initial value. Since the data is random, we perform 10 independent trials and plot the mean results, as well as a 95% confidence interval around the mean.

Figure 2 shows that in terms of stationarity violation, ADAPD is superior to DGD and Prox-GPDA. Using less than 50% gradient computations on each agent, ADAPD is able to both solve the localization problem and find the true targets with fewer communications than the other methods. Additionally, ADAPD-OG slightly outperforms Prox-GPDA in both metrics, thus even with a 1:1 gradient computation to neighbor communication, we have better numerical performance than competitors.

### C. Convolutional Neural Networks

For these experiments, we fix \( N = 8 \) agents and use a ring-structured graph with

\[
W_{ij} = \begin{cases} 
\frac{1}{N}, & (i, j) \in E \text{ or } i = j, \\
0, & \text{otherwise}.
\end{cases}
\]

We train the models on a cluster of 8 NVIDIA Tesla V100 GPUs, where each GPU represents an agent. PyTorch is used in the training of the models and OpenMPI is used to perform the neighbor communication of the neural network weights.

1) MNIST: The first Convolutional Neural Network (CNN) experiment we perform is training LeNet [43] on the MNIST dataset. We make the activation function for each layer the hyperbolic tangent function to ensure smoothness of the local objective functions. Since methods like DSGT [25] and D-GET [12] require multiple neighbor communications during each update, we instead fix the number of epochs for this experiment to 20 and fix the mini-batch size to 64 for all methods. For ADAPD and ADAPD-OG, we simply replace the full gradient, \( \nabla f_i(x_i) \), by a stochastic gradient, \( \nabla f_i(x_i) \), during each local agent update. As stated above, if (28) holds, then ADAPD converges in expectation at the same rate as presented in Theorem 2. However, since we fix the epoch budget for this problem, we limit each agent to performing two local stochastic gradient steps during each ADAPD iteration. A convergence analysis for a stochastic version of ADAPD-OG is non-trivial and is a future goal of ours. Similar to [2], we report the stationarity violation for all methods, as well as the training loss and testing accuracy using the average of the local agent’s weights

3In practice, this is not feasible due to the decentralized communication pattern, however, an average model can be obtained after all local training has been done by performing many neighbor communication rounds [18]. Note that the training loss reported here is not scaled by \( \frac{1}{N} \) to facilitate a fair comparison with standard CNN training methods (i.e. centralized training).

Additionally, we report the wall-clock time taken to reach and exceed 97% testing accuracy for the MNIST image classification problem in Table I. This value comes from selecting the highest whole number testing accuracy that most methods exceed. D-GET does not achieve this accuracy in the allotted amount of epochs. The “Samples” column indicates the amount of data visited by each agent to achieve the 97% testing accuracy and the “Communications” column indicates the corresponding number of communications performed by each agent (for D-GET, these values are simply the total numbers used during training). For sake of completeness, we include each method’s highest testing accuracy in the last column.

While D-GET is able to achieve a slightly lower stationarity violation, the training loss and testing accuracy indicate it does not converge to a solution that solves the classification problem well. Both D-PSGD and DSGT perform similarly to ADAPD in terms of stationarity violation and training loss, however, Figure 3 and Table I show that both ADAPD and
ADAPD-OG outperform these methods in terms of testing accuracy, suggesting that ADAPD is able to find a solution that generalizes better than the primal-only methods. Additionally, Table I shows that ADAPD and ADAPD-OG require far less communications to achieve a high testing accuracy. In a network setting where communication time dominates the computation time, ADAPD and its variants can far outperform the competitors.

2) CIFAR-10: The second CNN experiment we perform is training the ALL-CNN model [44] on the CIFAR-10 dataset [45]. We add batch normalization after every ReLU activation function and perform no data augmentation prior to training. For these experiments, we fix the mini-batch size to 128 for all methods and limit the number of updates so that each method runs for 500 epochs. We only use the ADAPD algorithm for these experiments and test with two different scenarios: (i) each agent performs 4 local stochastic gradient steps before communicating and (ii) each agent computes only 1 local stochastic gradient step, but we tune the dual step-size in (22) and (23). We report the same metrics as in the MNIST experiment in Figure 4 using the average of the local agent’s weights.

Similar to the MNIST image classification problem, we report the wall-clock time taken to reach and exceed 89% testing accuracy in Table II. We note that the ADAPD method performs worse than the other methods in terms of stationarity violation (see Figure 4). This is largely due to the violation of the consensus term, $\|X_k - \bar{X}_k\|^2_F$. However, ADAPD performs better than the competitors in terms of testing accuracy (see Figure 4 and Table II). Similar to the MNIST results, this suggest that ADAPD is able to find a solution to the image classification problem that generalizes better than the competitors. Additionally, ADAPD greatly saves on the number of data samples and communications necessary to achieve a high testing accuracy.

V. CONCLUSION

In this work, we presented ADAPD: A Decentralized Primal-Dual framework for solving non-convex and smooth consensus optimization problems over a network of agents. Two variants to ADAPD are presented, the ADAPD-OG (One Gradient) and the ADAPD-MC (Multiple Communications). We demonstrated that ADAPD and its variants achieve the optimal complexity result for the class of smooth, non-convex, decentralized consensus problems considered in this work. Finally, we presented four numerical experiments that
validate our claim that ADAPD outperforms other state-of-the-art decentralized methods. Future research topics would be extending the theoretical guarantees of ADAPD-Og to the stochastic case and demonstrating convergence in a time-varying/asynchronous setting of ADAPD and its variants.

**REFERENCES**

[1] B. McMahan and D. Ramage, “Federated learning: Collaborative machine learning without centralized training data,” *Google AI Blog*, 2017.

[2] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, “Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc., 2017, pp. 5330–5340.

[3] X. Liang, A. M. Javid, M. Skoglund, and S. Chatterjee, “Asynchronous primal-dual algorithm for fast distributed nonconvex optimization and learning over networks,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 3027–3036.

[4] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, “On the linear convergence of the admm in decentralized consensus optimization,” *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[5] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié, “Optimal algorithms for smooth and strongly convex distributed optimization in networks,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 3027–3036.

[6] Y. Arjevani, J. Bruna, B. Can, M. Gurbuzbalaban, S. Jegelka, and H. Lin, “Ideal: Inexactly decentralized accelerated augmented lagrangian method,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 20648–20659.

[7] K. Pan, S. Lu, and M. Hong, “Improving the sample and communication complexity for decentralized non-convex optimization: Joint gradient estimation and tracking,” in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. Virtual: PMLR, 13–18 July 2020, pp. 9217–9228.

[8] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié, “Optimal algorithms for smooth and strongly convex distributed optimization in networks,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 3027–3036.

**TABLE II**

*Time to reach 89% testing accuracy on the CIFAR-10 image classification problem. Final column represents highest overall testing accuracy. Bold items indicate the best value.*

| Method             | To reach 89% testing accuracy | Highest accuracy (%) |
|--------------------|-------------------------------|----------------------|
| D-PSGD             | 2,089.29                      | 89.16                |
| DSGT               | 2,423.85                      | 89.22                |
| DGET               | X                             | 86.14                |
| ADAPD (4 SGD steps)| 2,097.8                       | 89.3                 |
| ADAPD (1 SGD step) | 1,473.55                      | 89.96                |

**Fig. 4.** In order from left to right: stationarity violation, training loss, testing accuracy, and a zoomed version of the testing accuracy over the last few hundred epochs for the CIFAR-10 image classification problem.
APPENDIX A

SUPPORTING LEMMATA AND PROOFS FOR THE FASTMIX ALGORITHM

Proof [of Lemma 1] Notice that the iterations of Alg. 3 define a polynomial in $W$, denote this as $\mathcal{P}(W,r)$ for any $r \geq 0$. Let $A^\mathcal{P} = \mathcal{P}(W,R)A^0$ be the output of Alg. 3 after $R \geq 1$ iterations, then

$$\mathcal{P}(W,R) = \alpha_0 R(\alpha W) l + \sum_{r=1}^{R} \alpha_r R(\alpha W) r$$

(50)

for some constants $\alpha_r, R(\alpha W) \in \mathbb{R}$ that depend on $\alpha W$ and the iteration $R$, for all $r = 0, \ldots, R$. Additionally, by line 3 in Alg. 3, we have the following relation among the polynomials $\mathcal{P}(W,R)$,

$$\mathcal{P}(W,R) = (1 + \alpha W) \mathcal{P}(W,R - 1) - \alpha W \mathcal{P}(W,R - 2)$$

(51)

for all $R \geq 2$, where $\mathcal{P}(W,1) = (1 + \alpha W) W - \alpha W I$ and $\mathcal{P}(W,0) = I$. Clearly, (50) satisfies Assumption 1(ii), since $W = W^T$. Proceeding by induction, when $R = 1$ we have

$$\mathcal{P}(W,1) = e = ((1 + \alpha W) W - \alpha W I) e = (1 + \alpha W) e - \alpha W e = e$$

where the second equality comes from Assumption 1(iii) of $W$. Thus, $e - \mathcal{P}(W,1) e = 0$, as required. Assume that $\mathcal{P}(W,r) e = e$ for all $r = 0, 1, \ldots, R - 1$, then by (51) we have

$$\mathcal{P}(W,R) e = (1 + \alpha W) \mathcal{P}(W,R - 1) e - \alpha W \mathcal{P}(W,R - 2) e = (1 + \alpha W) W e - \alpha W I e = e$$

where we have used the inductive hypothesis and Assumption 1(iii) of $W$. Thus Alg. 3 satisfies Assumption 1(iii). Finally, by Lemma 2 in [38], Alg. 3 satisfies the spectral property in Assumption 1(iv), and thus we complete the proof.

APPENDIX B

SUPPORTING LEMMATA AND PROOFS FOR ADAPD

The results below follow from Lemma 6 and Equation (76) in [37]. For sake of completeness, we include a proof here.

Lemma 7: If (26) is satisfied and $\eta < \frac{1}{2\gamma}$, then

$$\mathcal{L}_\gamma(X^{k+1}, X_0^k, Y^k, Z^k) - \mathcal{L}_\gamma(X^k, X_0^k, Y^k, Z^k) \leq \frac{2\gamma}{\gamma - \eta} \|X^{k+1} - X^k\|^2_F + \frac{\eta}{\gamma - \eta} I, \forall k \geq 0.$$  

(52)

Proof (By 11), we have

$$-F(X^k) \leq -F(X^{k+1}) + \langle -\nabla F(X^{k+1}), X^k - X^{k+1} \rangle + \frac{\eta}{2} \|X^{k+1} - X^k\|^2_F.$$  

Hence,

$$\mathcal{L}_\gamma(X^{k+1}, X_0^k, Y^k, Z^k) - \mathcal{L}_\gamma(X^k, X_0^k, Y^k, Z^k) \leq \langle -\nabla F(X^{k+1}), X^{k+1} - X^k \rangle - \frac{1}{\gamma} \|X^k - X_0^k\|^2_F + \frac{\eta}{2} \|X^{k+1} - X^k\|^2_F
= \langle \nabla F(X^{k+1}) + Y^k + \frac{\eta}{2} (X^{k+1} - X^k), X^k - X^{k+1} \rangle
+ \frac{\eta}{2} \|X^{k+1} - X^k\|^2_F - \frac{1}{\gamma} \|X^k - X_0^k\|^2_F.$$  

However,
where we have used the compatibility of the 2-norm and the Frobenius norm and Assumption 1(iv). Hence, it holds
\[
\begin{align*}
\mathcal{L}_\eta(X^{k+1}, X_0^{k+1}, Y^{k+1}, Z^{k+1}) & \leq \mathcal{L}_\eta(X^{k+1}, X_0^{k+1}, Y^{k+1}, Z^{k}) + \frac{2}{\eta} \left\| X^{k+1} - X_0^k \right\|_F^2 \\
& \quad + \left( V_{X_0} \mathcal{L}_\eta(Y^{k+1}, X_0^{k+1}, Y^{k}, Z^{k}), X_0^{k+1} - X_0^{k} \right) \\
& \leq \frac{2}{\eta} \left\| X^{k+1} - X_0^k \right\|_F^2 \\
& \quad + \left( V_{X_0} \mathcal{L}_\eta(Y^{k+1}, X_0^{k+1}, Y^{k}, Z^{k}), X_0^{k+1} - X_0^{k} \right)
\end{align*}
\]
(53)

Rearranging terms, we get the desired result.

Lemma 9: For all $k \geq 0$, the followings hold:
\[
\begin{align*}
\mathcal{L}_\eta(X^{k+1}, X_0^{k+1}; Y^{k+1}, Z^{k+1}) & = \eta \left\| Y^{k+1} - Y^k \right\|^2_F \\
\mathcal{L}_\eta(X^{k+1}, X_0^{k+1}; Y^{k+1}, Z^{k+1}) & = \eta \left\| Z^{k+1} - Z^k \right\|^2_F
\end{align*}
\]
(54)
(55)

Proof By the $Y$ update (22), we have
\[
\begin{align*}
\mathcal{L}_\eta(X^{k+1}, X_0^{k+1}; Y^{k+1}, Z^{k+1}) & = \left\langle Y^{k+1} - Y^k, X^{k+1} - X_0^{k+1} \right\rangle \\
& = \eta \left\| Y^{k+1} - Y^k \right\|^2_F
\end{align*}
\]
(32)

for all $k \geq 0$. Hence, (54) holds. Similarly, by the $Z$ update (23), we have
\[
\begin{align*}
\mathcal{L}_\eta(X^{k+1}, X_0^{k+1}; Y^{k+1}, Z^{k+1}) & = \left\langle Z^{k+1} - Z^k, \sqrt{I - W} X_0^{k+1} \right\rangle \\
& = \eta \left\| Z^{k+1} - Z^k \right\|^2_F
\end{align*}
\]
(33)

for all $k \geq 0$. Thus (55) holds, and we complete the proof.

Proof [of Lemma 2] Recall $\tilde{Z}^k \doteq \sqrt{I - W} Z^k$ for all $k \geq 0$. Thus by (21), we have
\[
\begin{align*}
X_0^k - X_0^{k-1} & = -\frac{1}{\eta} \left( -Y^{k-1} - \frac{1}{\eta} (X^k - X_0^{k-1}) + Z^{k-1} + \frac{1}{\eta} (I - W) X_0^{k-1} \right) \\
& \doteq \frac{1}{2} (Y^k - \frac{1}{2} X_0^k - X_0^{k-1}) + \frac{1}{2} (I - W) (X_0^k - X_0^{k-1})
\end{align*}
\]
(22)

(23)

Combining like terms, rearranging, and multiplying both sides by $\frac{1}{2}$ gives (34). To prove (35), we have from (22) that $Y^{k+1} = Y^k - \frac{1}{2} (X^k + X_0^k)$; plugging this into (26) with $k \leftarrow k - 1$ yields the desired result.

Proof [of Lemma 3] The inequality follows from rewriting $\mathcal{L}_\eta(X^{k+1}, X_0^{k+1}; Y^{k+1}, Z^{k+1}) - \mathcal{L}_\eta(X^k, X_0^k, Y^k, Z^k)$ as the summation of the left-hand sides of (32), (53), (54), and (55) and using those four inequalities.

Proof [of Lemma 4] To prove (37), by (35), we have
\[
\begin{align*}
\eta \left\| Y^{k+1} - Y^k \right\|^2_F & = \eta \left\| R^{k+1} - R^k - \nabla F(X^{k+1}) + \nabla F(X^k) - \frac{1}{\eta} Y_0^k \right\|^2_F \\
& \leq 4 \eta \left( \left\| R^{k+1} \right\|^2_F + \left\| R^k \right\|^2_F + \left\| \nabla F(X^{k+1}) - \nabla F(X^k) \right\|^2_F + \frac{1}{\eta} \left\| V_0^{k} \right\|^2_F \right) \\
& \leq 4 \eta \left( \left\| R^{k+1} \right\|^2_F + \frac{1}{\eta} \left\| V_0^{k} \right\|^2_F + 8 \eta \epsilon_k \right)
\end{align*}
\]
(33)

where in the last inequality we have further used $\epsilon_{k+1} \leq \epsilon_k$ for all $k \geq 0$. To prove (38), notice that if $Z^0 \in \text{range}(\sqrt{I - W})$, then by (19), $Z^k \in \text{range}(\sqrt{I - W})$ for all $k \geq 0$. Thus we have
\[
\begin{align*}
\eta \left\| Z^{k+1} - Z^k \right\|^2_F & \leq \frac{1}{\eta} \left\| \sqrt{I - W} (Z^{k+1} - Z^k) \right\|^2_F. \\
& \leq \frac{1}{\eta} \left\| Y^{k+1} - Y^k \right\|^2_F + \frac{2}{\eta} \left\| W V_0^k \right\|^2_F \\
& \leq 8L^2 \eta \left\| X^{k+1} - X^k \right\|^2_F + \frac{8}{\eta} \left\| V_0^k \right\|^2_F + 16 \eta \epsilon_k + \frac{2}{\eta} \left\| W V_0^k \right\|^2_F \\
& \leq 8L^2 \eta \left\| X^{k+1} - X^k \right\|^2_F + \frac{10}{\eta} \left\| V_0^k \right\|^2_F + 16 \eta \epsilon_k
\end{align*}
\]
(37)

where the last inequality uses Assumption 1(iv). Dividing both sides of (59) by $\rho_2$ and using (56), we complete the proof.

Proof [of Lemma 5] By (35), we have
\[
\begin{align*}
\left\langle Y^{k+1} - Y^k, X_0^k - X_0^0 \right\rangle & \left. = \right. \left\langle R^{k+1} - R^k - \nabla F(X^{k+1}) + \nabla F(X^k) - \frac{1}{\eta} Y_0^k, X_0^k - X_0^0 \right\rangle \\
& \left. = \right. \left\langle R^{k+1} - R^k - \nabla F(X^{k+1}) + \nabla F(X^k), X_0^k - X_0^0 \right\rangle
\end{align*}
\]
(60)

We handle the two sides of (60) separately. First, we have
\[
\begin{align*}
\left\langle Y^{k+1} - Y^k, X_0^k - X_0^0 \right\rangle & \left. \doteq \right. \left\langle \sqrt{I - W} Z^{k+1} - \sqrt{I - W} Z^k + \frac{1}{\eta} W V_0^k, X_0^k - X_0^0 \right\rangle \\
& \left. \doteq \right. \left\langle \sqrt{I - W} Z^{k+1} - \sqrt{I - W} Z^k + \frac{1}{\eta} W V_0^k, X_0^k - X_0^0 \right\rangle \\
& \doteq \left\langle \sqrt{I - W} Z^{k+1} - \sqrt{I - W} Z^k + \frac{1}{\eta} W V_0^k, X_0^k - X_0^0 \right\rangle
\end{align*}
\]
(61)
Since the minimum eigenvalue of \( \matr{A} \) is \( \rho_{\matr{A}} > 0 \) in (7), it holds \( \frac{\rho_{\matr{A}}}{\rho_{\matr{B}}} \matr{A} \leq \frac{\rho_{\matr{A}}}{\rho_{\matr{B}}} \matr{W} \) when \( \rho_{\matr{B}} \geq \frac{\rho_{\matr{A}}}{\rho_{\matr{B}}} \matr{W} \). Hence, we have 

\[
0 \leq \frac{C_{\rho_{\matr{A}}-20\rho_{\matr{A}}}}{2\rho_{\matr{B}}} \|
\text{\textbf{V}}_0\|_F^2 + \frac{C_{\rho_{\matr{A}}-20\rho_{\matr{A}}}}{2\rho_{\matr{B}}} \|
\text{\textbf{V}}_0\|_W^2
\]

by noticing 

\[
\frac{C_{\rho_{\matr{A}}-20\rho_{\matr{A}}}}{2\rho_{\matr{B}}} \|
\text{\textbf{V}}_0\|_F^2 + \frac{C_{\rho_{\matr{A}}-20\rho_{\matr{A}}}}{2\rho_{\matr{B}}} \|
\text{\textbf{V}}_0\|_W^2 \geq 0.
\]

In addition, it holds 

\[
\frac{C_{\rho_{\matr{A}}-20\rho_{\matr{A}}}}{2\rho_{\matr{B}}} \|
\text{\textbf{V}}_0\|_F^2 + \frac{C_{\rho_{\matr{A}}-20\rho_{\matr{A}}}}{2\rho_{\matr{B}}} \|
\text{\textbf{V}}_0\|_W^2 = 0.
\]

Thus, by the definition of \( f \) in (12), we have that for any integer number \( K \geq 1 \), 

\[
\begin{align*}
\sum_{k=0}^{K-1} (\Phi^{k+1} - f) \\
\geq \sum_{k=0}^{K-1} \left( \mathcal{L}_{\mathcal{K}}(\mathbf{X}^{k+1}, \mathbf{X}^{k+1}, \mathbf{Y}^{k+1}, \mathbf{Z}^{k+1}) - \mathcal{L}_{\mathcal{K}}(\mathbf{X}^{k}, \mathbf{X}^{k}, \mathbf{Y}^{k}, \mathbf{Z}^{k}) \right) \\
\geq \sum_{k=0}^{K-1} \left( \frac{2(8L^2\rho_{\matr{A}}+16L^2\rho_{\matr{B}})^2+2L^2\rho_{\matr{B}}\rho_{\matr{C}}}{4\rho_{\matr{D}}^2} \|
\text{\textbf{X}}^{k+1} - \mathbf{X}^2\|_F^2 \\
- \frac{1}{2L} \|
\text{\textbf{X}}^{k+1} - \mathbf{X}^2\|_F^2 + \frac{1}{2L} \|
\text{\textbf{X}}^{k+1} - \mathbf{X}^2\|_W^2 \right) \\
\geq \frac{1}{2L} \|
\text{\textbf{X}}^{k+1} - \mathbf{X}^2\|_F^2 + \frac{1}{2L} \|
\text{\textbf{X}}^{k+1} - \mathbf{X}^2\|_W^2 \geq -M.
\end{align*}
\]

Thirdly, by (40) and the definition of \( \Phi^k \) in (41), we have 

\[
\begin{align*}
\mathbf{X}^{k+1} - \mathbf{X}^2 & \leq \Phi^k + \frac{\rho_{\matr{A}}+(32L^2+16L^2)^2+4\rho_{\matr{D}}}{4\rho_{\matr{D}}^2} \epsilon_k. \\
\begin{cases}
\rho_{\matr{A}} & \geq \rho_{\matr{D}}.
\end{cases}
\end{align*}
\]

Hence, it holds from the choice of \( C \) and \( \eta \) that 

\[
\Phi^{k+1} \leq \Phi^k + \frac{\rho_{\matr{A}}+(32L^2+16L^2)^2+4\rho_{\matr{D}}}{4\rho_{\matr{D}}^2} \epsilon_k.
\]

Now assume that there exists \( \epsilon_0 \geq 0 \) such that \( \Phi^{\epsilon_0} - f < -\frac{\rho_{\matr{A}}+(32L^2+16L^2)^2+4\rho_{\matr{D}}}{4\rho_{\matr{D}}^2} \epsilon_k \). Then summing up (68) gives \( \Phi^{\epsilon_k} \leq \Phi^{\epsilon_0} - f + \frac{\rho_{\matr{A}}+(32L^2+16L^2)^2+4\rho_{\matr{D}}}{4\rho_{\matr{D}}^2} \epsilon_k \) for all \( k \geq \epsilon_0 \). Hence, \( \epsilon_k < -\infty \), which contradicts (66). Therefore, we conclude that \( \Phi^{\epsilon_k} \geq f - \frac{\rho_{\matr{A}}+(32L^2+16L^2)^2+4\rho_{\matr{D}}}{4\rho_{\matr{D}}^2} \epsilon_k \) and complete the proof. 

\[
\begin{align*}
\mathcal{L}_{\mathcal{K}}(\mathbf{X}^{k+1}, \mathbf{X}^{k+1}, \mathbf{Y}^{k+1}, \mathbf{Z}^{k+1}) \\
= \mathcal{L}_{\mathcal{K}}(\mathbf{X}^{k+1}) + \frac{1}{2\rho_{\matr{D}}} \|
\text{\textbf{X}}^{k+1} - \mathbf{X}^2\|_F^2 \\
+ \frac{1}{2\rho_{\matr{D}}} \|
\text{\textbf{Z}}^{k+1} - \mathbf{Z}^2\|_F^2 \\
+ \frac{1}{2\rho_{\matr{D}}} \|
\text{\textbf{Y}}^{k+1} - \mathbf{Y}^2\|_F^2 \\
+ \frac{1}{2\rho_{\matr{D}}} \|
\text{\textbf{X}}^{k+1} - \mathbf{X}^2\|_W^2 \\
+ \frac{1}{2\rho_{\matr{D}}} \|
\text{\textbf{Z}}^{k+1} - \mathbf{Z}^2\|_W^2 \\
+ \frac{1}{2\rho_{\matr{D}}} \|
\text{\textbf{Y}}^{k+1} - \mathbf{Y}^2\|_W^2 \\
= \frac{C_{\rho_{\matr{A}}-20\rho_{\matr{A}}}}{2\rho_{\matr{B}}} \|
\text{\textbf{V}}_0\|_F^2 + \frac{C_{\rho_{\matr{A}}-20\rho_{\matr{A}}}}{2\rho_{\matr{B}}} \|
\text{\textbf{V}}_0\|_W^2 + \frac{C_{\rho_{\matr{A}}-20\rho_{\matr{A}}}}{2\rho_{\matr{B}}} \|
\text{\textbf{V}}_0\|_W^2 \\
+ \frac{C_{\rho_{\matr{A}}-20\rho_{\matr{A}}}}{2\rho_{\matr{B}}} \|
\text{\textbf{V}}_0\|_W^2 = 0.
\end{align*}
\]
Now, by (23) and (59), we have
\[ \sum_{k=0}^{\infty} \frac{\|X_k - X_k\|^2}{F} \leq \frac{1}{1 - \rho^*} \frac{\|X_0 - X_k\|^2}{F} \]
and thus the inequality in (69) implies the desired result. □

Proof [of Theorem 2] First, we have for all \( k \geq 0 \) that
\[ \|X_k - X_k\|^2 \leq \frac{1}{1 - \rho^*} \|X_0 - X_k\|^2 \]
and simplifying the result, we obtain
\[ \|X_k - X_k\|^2 \leq \frac{1}{1 - \rho^*} \|X_0 - X_k\|^2 \].

Now, by (23) and (59), we have
\[ \eta \|X_k - X_k\|^2 = \eta \|\nabla F(X_k)\|^2 \]
and by (22),
\[ \|X_k - X_k\|^2 \leq \frac{4L^2\eta^2}{F} \|X_k - X_k\|^2 + \frac{4}{F} \|V_k\|^2 + 8\eta^2 e_k \].

Thus,
\[ \frac{1}{F} \sum_{k=0}^{K-1} \|X_k - X_k\|^2 \]
where we have used the fact that \( \|I - W\| \leq 2 \), and defined \( C_2 \doteq \max \left\{ \frac{48L^2\eta^2}{F}, \frac{208}{1 - \rho^*} \right\} \) and \( C_3 \doteq \frac{32L + 16L_2p_1 + 4(1-p_1)^2}{2Lp_2} \).

Furthermore, we use (34) and (35) to have
\[ \nabla F(X_k) + \sqrt{\nabla F(X_k)} \leq X_k - \frac{1}{\eta} (I + W)[X_0 - X_k]. \]

Now, by Assumption 1(iii), we have \( e^T \sqrt{\nabla F(X_k)} = 0 \). Hence
\[ \frac{1}{F} \sum_{k=0}^{K-1} \|\nabla F(X_k)\|^2 \]

and
\[ \sum_{k=0}^{K-1} \|\nabla F(X_k)\|^2 \]

We complete the proof. □

APPENDIX C
SUPPORTING LEMMAS AND PROOFS FOR ADAPD-OG

We begin the convergence analysis with showing the change in the augmented Lagrangian function value between two consecutive iterations.

Lemma 10: Provided that \( \eta < \frac{1}{L} \), we have
\[ L_\eta \langle X_k, X_k - Y_k, Z_k \rangle + L_\eta \langle X_k, X_k - Y_k, Z_k \rangle \]

for all \( k \geq 0 \).

Proof By (11), we have
\[ F(X_k) \]

and
\[ F(X_k) \leq F(X_k) + \langle \nabla F(X_k), X_k - X_k \rangle + \frac{\eta}{2} \|X_k - X_k\|^2 . \]
Thus,
\[
\begin{align*}
\mathcal{L}_\eta(X^{k+1},X^0,\mathbf{Z}^k) - \mathcal{L}_\eta(X^k,\mathbf{X}^0,\mathbf{Y}^k,\mathbf{Z}^k) & \leq \langle \nabla F(X^k),X^{k+1} - X^k \rangle + \frac{\eta}{2} \|X^{k+1} - X^k\|_F^2 + \langle Y^k,X^{k+1} - X^k \rangle \\
& + \frac{1}{2\eta} \|X^{k+1} - X^k\|_F^2 - \frac{1}{2\eta} \|X^k - X^k\|_F^2 \\
& = \langle \nabla F(X^k) + Y^k + \frac{1}{2\eta}(X^{k+1} - X^k),X^{k+1} - X^k \rangle \\
& + \frac{L\eta-1}{2\eta} \|X^{k+1} - X^k\|_F^2 \\
& \leq \frac{C\eta}{2\eta} \|X^{k+1} - X^k\|_F^2.
\end{align*}
\]

\[
\square
\]

**Lemma 11:** Let \((X^k,\mathbf{X}^0,\mathbf{Y}^k,\mathbf{Z}^k)\) be obtained from Alg. 2 or equivalently by updates (30) and (21)-(23). If \(\eta < \frac{1}{C}\), then it holds for all \(k \geq 0\),
\[
\begin{align*}
\mathcal{L}_\eta(X^{k+1},X^0;\mathbf{Y}^k,\mathbf{Z}^k) & \leq \frac{L\eta}{2\eta} \|X^{k+1} - X^k\|_F^2 + \frac{2}{\eta} \|V^k\|_F^2, \\
& + \eta \|Y^{k+1} - Y^k\|_F^2 + \eta \|Z^{k+1} - Z^k\|_F^2.
\end{align*}
\]

\[
\text{Lemma 12: Under the assumptions of Lemma 11, it holds that for all } k \geq 0,
\begin{align*}
\eta \|Y^{k+1} - Y^k\|_F^2 & \leq \frac{L^2}{2\eta^2} \|\mathbf{X}^k - \mathbf{X}^{k-1}\|_F^2 + \frac{2}{\eta} \|V^k\|_F^2, \\
\eta \|Z^{k+1} - Z^k\|_F^2 & \leq \frac{L^2}{2\eta^2} \|\mathbf{X}^k - \mathbf{X}^{k-1}\|_F^2 + \frac{2}{\eta} \|V^k\|_F^2.
\end{align*}
\]

We now define a new Lyapunov function based on the results from Lemma 14. For the same given, fixed, \(\hat{C} \geq \frac{12\eta p_2}{\rho_2 pN}\), used in (83), define
\[
\Phi^k \triangleq \mathcal{L}_\eta(X^k,\mathbf{X}^0,\mathbf{Y}^k,\mathbf{Z}^k) + \frac{\eta}{2\eta} \|\mathbf{Y} - \mathbf{W}\mathbf{X}^k\|_F^2 + \frac{\eta}{2\eta} \|\mathbf{X}_0^k - \mathbf{X}_0^{k-1}\|_F^2 \\
+ \frac{2L^2\rho_2\eta + 8L^2\eta + C\eta \rho_2}{2\eta^2} \|\mathbf{X}^k - \mathbf{X}^{k-1}\|_F^2.
\]

Before showing that this Lyapunov function is lower bounded, we first demonstrate the relation between the Lyapunov function at two consecutive iterations. Using Lemma 14, for all \(k \geq 0\), we have
\[
\hat{\Phi}^{k+1} + \left(\frac{\rho_2(\hat{C} + 1) - \rho_2(\rho_2 + 1) + 2L^2\eta^2 - \rho_2\eta}{2\eta}\right) \|\mathbf{X}^{k+1} - \mathbf{X}^k\|_F^2 \\
+ \left(\frac{1}{2\eta} - \hat{C}\eta\right) \|\mathbf{X}_0^{k+1} - \mathbf{X}_0^k\|_F^2 \leq \Phi^k
\]
which comes directly from adding and subtracting \(L^2\rho_2\eta + 8L^2\eta + C\eta \rho_2\) to the left hand side of (83), combining like terms, and using (84). Now we show that \(\Phi^k\)
has a finite lower bound via the following proposition.

**Proposition 2:** Under Assumptions 1 and 2, let \((X^k,\mathbf{X}^0,\mathbf{Y}^k,\mathbf{Z}^k)\) be obtained from Alg. 2 or equivalently by (30) and (21)-(23). Choose \(\hat{C}\) and \(\eta\) such that
\[
\hat{C} \geq \frac{12\eta p_2}{\rho_2 pN} \quad \text{and} \quad \eta \leq \min\left\{\frac{1}{\hat{C}L}, \frac{\rho_2(\hat{C} + 1) - \rho_2(\rho_2 + 1) + 2L^2\eta^2 - \rho_2\eta}{8L^2\eta + C\eta \rho_2}\right\}.
\]
Then the Lyapunov function (84) is uniformly lower bounded. More specifically, for all \(k \geq 0\),
\[
\Phi^k \geq f - 1 > -\infty,
\]
where \(f\) is defined in Assumption 2.

We are now in position to prove the convergence rate results of Alg. 2.

**Theorem 4:** Under the same conditions assumed in Proposition 2, it holds that
\[
\hat{C} \geq \frac{\rho_2}{\rho_2 pN} \sum_{k=0}^{K-1} \min\left\{\frac{\rho_2(\hat{C} + 1) - \rho_2(\rho_2 + 1) + 2L^2\eta^2 - \rho_2\eta}{8L^2\eta + C\eta \rho_2}, \frac{1}{2\eta}\hat{C}\right\}.
\]

\[
\text{Proof [of Theorem 3]} \quad \text{By (23), we have}
\begin{align*}
\frac{1}{\eta} \|\mathbf{X}^{k+1} - \mathbf{X}^k\|_F^2 & = \eta \|\mathbf{Y}^{k+1} - \mathbf{Z}^k\|_F^2 \\
& \leq 4L^2\eta \|\mathbf{X}^k - \mathbf{X}^{k-1}\|_F^2 + \frac{\eta}{2\eta} \|V^k\|_F^2
\end{align*}
\]
and by (22),
\[
\|\mathbf{X}^{k+1} - \mathbf{X}^k\|_F^2 = \eta \|\mathbf{X}^{k+1} - \mathbf{X}^k\|_F^2 \\
\leq \frac{L^2\eta^2}{2\eta} \|\mathbf{X}^k - \mathbf{X}^{k-1}\|_F^2 + 2 \|V^k\|_F^2.
\]
Thus,
\[
\frac{1}{K} \sum_{k=0}^{K-1} \|X^{k+1} - \hat{X}^{k+1}\|_F^2 \\
\leq \frac{1}{(1-\rho_2)^2 K} \sum_{k=0}^{K-1} \left\| (I - W)X^{k+1} \right\|_F^2 \\
\leq \frac{2}{(1-\rho_2)^2 K} \sum_{k=0}^{K-1} \left( \left\| (I - W)(X^{k+1} - X_0) \right\|_F^2 + \sum_{k=0}^{K-1} \left\| (I - W)X_0 \right\|_F^2 \right) \\
\leq \frac{24L^2\rho^2}{(1-\rho_2)^2 K} \sum_{k=0}^{K-1} \left\| X^{k+1} - X_0 \right\|_F^2 + \frac{28}{(1-\rho_2)^2 K} \sum_{k=0}^{K-1} \left\| X_0 \right\|_F^2 \\
\leq \frac{112}{(1-\rho_2)^2 K} \sum_{k=0}^{K-1} \left\| X^{k+1} - X_0 \right\|_F^2.
\]
(92)

We complete the proof.

(93)

where we have used the fact that \(\|I - W\|_2 \leq 2\) and defined 
\(\bar{C}_2 \doteq \max \left\{ \frac{48L^2\rho^2}{(1-\rho_2)^2}, \frac{112}{(1-\rho_2)^2} \right\} \). Furthermore, we use (34) and (46) to have
\[
\nabla F(X^k) + \sqrt{\mathbf{I} - \mathbf{W}}Z^{k+1} = -\frac{1}{\eta}(I + W)[X_0^{k+1} - X_0^k].
\]
(94)

Now, using Assumption 1(ii), we have \(\mathbf{e}^T \sqrt{\mathbf{I} - \mathbf{W}} = 0\). Hence,
\[
\frac{1}{K} \sum_{k=0}^{K-1} \left\| \nabla f(\hat{X}^{k+1}) \right\|_F^2 \\
= \frac{1}{K} \sum_{k=0}^{K-1} \left\| \frac{1}{\sqrt{2}} \mathbf{e}^T (\nabla F(\hat{X}^{k+1}) + \sqrt{\mathbf{I} - \mathbf{W}}Z^{k+2}) \right\|_F^2 \\
\leq \left\| \frac{1}{\sqrt{2}} \mathbf{e}^T \right\|_2^2 \frac{1}{K} \sum_{k=0}^{K-1} \left\| F(\hat{X}^{k+1}) + \sqrt{\mathbf{I} - \mathbf{W}}Z^{k+2} \right\|_F^2 \\
\leq \frac{2}{K} \sum_{k=0}^{K-1} \left\| F(\hat{X}^{k+1}) + \sqrt{\mathbf{I} - \mathbf{W}}Z^{k+2} \right\|_F^2 \\
\leq \frac{2}{K} \sum_{k=0}^{K-1} \left\| F(X^{k+1}) - \nabla F(X^{k+1}) \right\|_F^2 \\
\leq \frac{2}{K} \sum_{k=0}^{K-1} \left\| X^{k+1} - X_0^{k+1} \right\|_F^2 \\
\leq \frac{2(\bar{C}_2 L^2 + \bar{C}_3)\Lambda_{\phi}}{C_1 K}.
\]
(95)

where we have used \(\|I + W\|_2 \leq 2\) in last inequality and defined \(\bar{C}_3 \doteq \frac{2}{\eta}\). Finally, we have that
\[
\min_{1 \leq k' \leq K} \left( \left\| \nabla f(X^{k'}) \right\|_2^2 + \left\| X^{k'} - \hat{X}^k \right\|_F^2 \right) \\
\leq \frac{1}{K} \sum_{k=0}^{K-1} \left( \left\| \nabla f(X^{k+1}) \right\|_2^2 + \left\| X^{k+1} - \hat{X}^k \right\|_F^2 \right) \\
\leq \frac{(2L^2 + 1)\bar{C}_2 \bar{C}_3 \Lambda_{\phi}}{C_1 K}.
\]
(96)

We complete the proof.