FedNew: A Communication-Efficient and Privacy-Preserving Newton-Type Method for Federated Learning

Anis Elgabli [1] Chaouki B. Issaid [1] Amrit S. Bedi [2] Ketan Rajawat [3] Mehdi Bennis [1] Vaneet Aggarwal [4]

Abstract

Newton-type methods are popular in federated learning due to their fast convergence. Still, they suffer from two main issues, namely: *low communication efficiency* and *low privacy* due to the requirement of sending Hessian information from clients to parameter server (PS). In this work, we introduced a novel framework called FedNew in which there is no need to transmit Hessian information from clients to PS, hence resolving the bottleneck to improve communication efficiency. In addition, FedNew hides the gradient information and results in a privacy-preserving approach compared to the existing state-of-the-art. The core novel idea in FedNew is to introduce a two level framework, and alternate between updating the inverse Hessian-gradient product using only one alternating direction method of multipliers (ADMM) step and then performing the global model update using Newton's method. Though only one ADMM pass is used to approximate the inverse Hessian-gradient product at each iteration, we develop a novel theoretical approach to show the converging behavior of FedNew for convex problems. Additionally, a significant reduction in communication overhead is achieved by utilizing stochastic quantization. Numerical results using real datasets show the superiority of FedNew compared to existing methods in terms of communication costs.

[1]University of Oulu [2]University of Marylad, College Park, MD, USA [3]Indian Institute of Technology Kanpur, India [4]Purdue University, IN, USA. Correspondence to: Anis Elgabli <anis.elgabli@oulu.fi>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

1. Introduction

In this paper, we consider the following *federated learning* (FL) problem

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x), \tag{1}$$

where d denotes the dimension of the model $x \in \mathbb{R}^d$ we wish to train, n is the total number of clients/clients in the system, $f_i(x)$ is a convex loss/risk associated with data stored on client $i \in [n] := \{1, 2, \ldots, n\}$, and $f(x)$ is the empirical loss/risk. While first-order methods for FL problems are extensively studied in the literature, recently a handful of second-order methods have been proposed for FL problems. The main advantage of second-order methods is their faster convergence rate, although they suffer from high communication costs. Besides that, sharing the first and second-order information (which contains client's data) may create a privacy issue. This is because the Hessian matrix contains valuable information about the properties of the local function/data, and when shared with the parameter server (PS), privacy may be violated by eavesdroppers or an honest-but-curious PS. For instance, authors in (Yin et al., 2014) show that the eigenvalues of the Hessian matrix can be used to extract important information of the input images.

In this work, we are interested in developing communication efficient second-order methods while still preserving the privacy of the individual clients. To this end, we propose a novel framework that hides the gradients as well as the Hessians of the local functions, yet uses second-order information to solve the FL problem. In particular, we divide the standard Newton step into *outer* and *inner* levels. The objective of the inner level is to learn what we call the *Hessian inverse-gradient product* or *Zeroth Hessian inverse-gradient product* for the computation-efficient version (i.e., $(\nabla^2 f(x^k))^{-1}\nabla f(x^k)$ or $(\nabla^2 f(x^0))^{-1}\nabla f(x^k)$) without sharing individual gradients and/or Hessians. We use the ADMM method to solve the inner level problem due to their faster convergence as compared to gradient descent (GD) methods. Specifically, one ADMM step is used to approximate the solution of the inner level problem at each iteration. The solution of the inner level is used to perform the outer level update, which is similar to Newton's method.

The proposed approach ensures privacy and communication efficiency. Next, we summarize the state-of-the-art of first and second-order FL methods.

1.1. First-order FL Methods

The *de-facto* solution to problem (1) is to use distributed (stochastic) gradient descent method (DGD/DSGD). In the k-th iteration of DGD, the PS shares x^k with all clients, each client computes its local gradient with respect to x^k and transmits it to the PS which aggregates them to form the global gradient and perform one GD step as follows.

$$x^{k+1} = x^k - \alpha \nabla f(x^k) = x^k - \frac{\alpha}{n} \sum_{i=1}^{n} \nabla f_i(x^k). \tag{2}$$

Throughout this paper, we will refer to FL with DGD as FedGD. To enable communication-efficient FL, several techniques were proposed for enhancing communication efficiency, which is a key bottleneck in the FL. These techniques include *model compression/quantization* (Konečnỳ et al., 2016; Alistarh et al., 2017; Khirirat et al., 2018; Elgabli et al., 2020), *local computation* utilizing schemes such as local SGD (McMahan et al., 2017a; Stich, 2020; Khaled et al., 2020; Mishchenko et al., 2021a), and *censoring or partial participation* (McMahan et al., 2017a; Gower et al., 2019; Chen et al., 2018). Moreover, solutions that utilize a combination of the above techniques (e.g., censoring and quantization) were also investigated (Sun et al., 2019; Issaid et al., 2021). Other relevant techniques for further reducing the communication cost of FL methods include the use of *momentum* (Mishchenko et al., 2019; Zhize Li & Richtárik, 2020), and *adaptive learning rates* (Malitsky & Mishchenko, 2019; Xie et al., 2019; Reddi et al., 2020; Mishchenko et al., 2021b).

1.2. Towards Second-order FL Methods

First-order FL methods suffer from slow convergence speed in terms of the number of iterations/communication rounds. Moreover, their convergence speed is a function of the condition number, which depends on: (i) the structure of the model being trained, (ii) the choice of loss function, and (iii) the distribution of training data. On the other hand, second-order methods are known to be much faster owing to the fact that they make an extra computational effort to estimate the local curvature of the loss landscape, which is useful in generating faster and more adaptive update directions. Therefore, second-order methods perform more computations per communication round to achieve less number of communication rounds. Since in FL, it is often communication and not computation that is the key bottleneck, second-order methods are becoming attractive and have recently gained attention. The Newton's method, which forms the basis for most efficient second-order methods, enjoys a fast *condition-number-independent* (local) convergence rate (Beck, 2014), is given by

$$x^{k+1} = x^k - \Big(\sum_{i=1}^{n} \nabla^2 f_i(x^k)\Big)^{-1} \sum_{i=1}^{n} \nabla f_i(x^k). \tag{3}$$

However, fewer number of communication rounds does not necessarily induce a lower communication cost. Communication overhead is also affected by the number of transmitted bits between the clients and PS, which depends on the size of the transmitted vector/matrix per communication round. Hence, a direct application of Newton's method does not induce an efficient distributed implementation as it requires repeated communication of the local Hessian matrices $\nabla^2 f_i \in \mathbb{R}^{d\times d}$ to the server. This is prohibitive and constitutes a massive number of transmitted bits, which requires high communication energy and bandwidth. Another important concern when implementing Newton's method is privacy, as it requires sharing both the gradient and the Hessian at each iteration, which makes it vulnerable to inversion attacks (Fredrikson et al., 2015; Hitaj et al., 2017). For instance, in a linear regression task, the Hessian matrix is nothing but $D^T D$, where D is the data matrix which results in privacy issues for each client.

Recently, (Safaryan et al., 2021) proposed a Newton-based framework that avoids communicating the full Hessian matrix at each iteration. The idea is to share a compressed version of the Hessian matrix utilizing compression techniques such as top-K, and Rank-R approximation. While this approach can reduce the communication cost, it has a number of shortcomings: (i) it does not solve the privacy issue since every client still shares the gradient and a compressed version of the Hessian matrix, (ii) it introduces further computations to perform compression, and (iii) it may not lead to high communication savings when the rank of the Hessian matrix is high. It is also worth mentioning that this approach, as well as the standard Newton's method, require matrix inversion at the PS at each iteration. On the other hand, even though the Newton Zero algorithm (proposed in (Safaryan et al., 2021)) requires matrix inversion only at the first iteration, it still shares the full Hessian matrix at the initial iteration, which necessitates $\mathcal{O}(d^2)$ communication cost at the first iteration, besides overlooking privacy. Moreover, in contrast to first-order FL methods which rely on simple aggregation at the PS, existing Newton-based methods require matrix inversion and multiplication (at least for the first iteration) which restricts the use of quantization to Hessian and gradient at the same time.

To obviate the above-mentioned limitations, we propose FedNew, a novel and efficient framework that ensures privacy by hiding the Hessian and the gradient. i.e., neither gradient nor Hessian is sent directly. In addition, FedNew reduces the communication cost compared to standard second-order methods by transmitting only $\mathcal{O}(d)$ information at each iteration, similar to first-order methods. Furthermore, FedNew reduces the computation cost when utilizing the zeroth Hessian at each iteration (similar to Newton Zero), since it performs matrix inversion only once (at the first iteration). Finally, to further reduce communication overhead, we propose a variant, coined Q-FedNew, which quantizes the transmitted variable. We summarize our contributions as follows.

(1) We propose a novel framework to solve the FL problem using second-order information. In particular, we decompose the objective function of the problem of learning the *Hessian inverse-gradient product* into a sum of separable functions and solve it in a distributed way. The framework alternates between updating the *Hessian inverse-gradient product* and the global model. To the best of our knowledge, this is the first work that utilizes one step of the ADMM method to estimate Newton directions with convergence guarantees.

(2) The proposed FedNew algorithm is rooted in a communication-efficient and privacy-preserving way to estimate the *Hessian inverse-gradient product* at each iteration. In contrast to existing approaches, FedNew does not require clients to share their gradient and the Hessian matrix (or compressed Hessian matrix) at any iteration, including the first one; hence $\mathcal{O}(d)$ communication cost is guaranteed at each iteration including the first one.

(3) We prove that the proposed FedNew algorithm asymptotically converges to the optimal direction of Newton method. We develop a novel proof technique to show that the proposed ADMM based inner and outer level iterates interacts with each other and shows converging behavior under some assumptions (cf. Sec. 3). Moreover, we provide a privacy analysis of FedNew and show that the reconstruction of gradients/Hessians is not possible (cf. Sec. 4).

(4) To further reduce the communication cost per iteration, we leverage stochastic quantization and propose quantized version called Q-FedNew. It was possible due to the unique feature of FedNew where clients share only a vector with PS that is not involved in any further multiplication/inversion at the PS (cf. Sec. 5).

We conduct extensive simulations on real datasets and show the performance gain of the proposed FedNew and Q-FedNew algorithms in comparison to Newton Zero and FedGD in terms of the number of transmitted bits and number of communication rounds.

2. Proposed Framework: FedNew

We start by writing the Newton update step for the global model x at iteration k as

$$x^{k+1} = x^k - (\nabla^2 f(x^k))^{-1}\nabla f(x^k), \tag{4}$$

where f (the empirical loss function) is assumed to be a continuously differentiable function over $x \in \mathbb{R}^d$. A key observation is that the direction $z(x^k){:=}(\nabla^2 f(x^k))^{-1}\nabla f(x^k)$ is the solution of the following problem

$$z(x^k) = \arg\min_{y\in\mathbb{R}^d} \frac{1}{2} y^T \nabla^2 f(x^k) y - y^T \nabla f(x^k). \tag{5}$$

Note that the problem in (5) is an unconstrained convex optimization problem that can be solved at iteration k to obtain the optimal direction $z(x^k)$. To solve this problem, one could utilize any existing iterative solver and obtain a direction that is close to the optimal $z(x^k)$ for a given x^k. Utilizing the solution of (5), we can then update x as, $x^{k+1} = x^k - z(x^k)$ which is a one Newton step from x^k. However, the main issue here is that solving (5) iteratively introduces a double loop, which can be communication and computation expensive. To address this issue, we propose a two-level framework detailed next.

2.1. Inner Level: One Pass ADMM

We note that solving (5) completely at each iteration k exhibits three challenges: (i) it requires sharing local Hessians and gradients with the PS which is communication expensive, (ii) it creates privacy issues since clients share their gradients and Hessians, and (iii) it requires matrix inversion at the PS at each iteration. We will explain how to overcome the limitation (iii) later in this section. Let us first start by addressing the limitations in (i) and (ii), by reformulating the problem as follows

$$\begin{aligned}\min_{y_i, y} \quad & \frac{1}{n}\sum_{i=1}^{n}\Big(\frac{1}{2} y_i^T(\nabla^2 f_i(x^k) + \alpha I) y_i - y_i^T \nabla f_i(x^k)\Big) \\ \text{s.t.} \quad & y_i = y, \quad \forall i \in [n],\end{aligned} \tag{6}$$

where we introduce a tuning parameter α. The problem in (6) is a consensus reformulation of the problem in (5) where y_i denotes the local directions, and y denotes the global direction. Note that the direction $-[\nabla^2 f(x^k) + \alpha I]^{-1}\nabla f(x^k)$, the inexact Newton direction, is also a descent direction (Marteau-Ferey et al., 2019; Karimireddy et al., 2018; Mishchenko, 2021; Zhang et al., 2021) which boils down to Levenberg-Marquardt algorithm for least square problems which exhibits global convergence (Levenberg, 1944; Marquardt, 1963; Bergou et al., 2020). By reformulating (5) as (6), the objective function becomes separable across clients, which allows the solution to be distributed and avoid clients sharing their Hessians/gradients at each iteration. We leverage ADMM algorithm to solve the problem in (6) in a distributed manner.

To simplify the notation, from now on, we will use H_i^k for Hessian $\nabla^2 f_i(x^k)$ and g_i^k for gradient $\nabla f_i(x^k)$ of client i at iteration k. With that, the augmented Lagrangian of the optimization problem (6) can be written as,

$$\mathcal{L}_\rho\left(\{y_i\}_{i=1}^n, y, \lambda\right) = \sum_{i=1}^n \left(\frac{1}{2} y_i^T (H_i^k + \alpha I) y_i - y_i^T g_i^k\right) \quad (7)$$
$$+ \sum_{i=1}^n \langle \lambda_i, y_i - y\rangle + \frac{\rho}{2}\sum_{i=1}^n \|y_i - y\|_2^2,$$

where $\lambda = \{\lambda_i\}_{i=1}^n$ is the collection of dual variables and $\rho > 0$ is a constant penalty parameter. One ADMM pass is performed at each iteration k. Hence, the primal and the dual variables are updated as follows.

(1) Each client i updates its primal variable y_i^k by solving the following problem,

$$y_i^k = \arg\min_{y_i} \left\{ \frac{1}{2} {y_i}^T (H_i^k + \alpha I) y_i + \langle \lambda_i^{k-1}, y_i - y^{k-1}\rangle - {y_i}^T g_i^k + \frac{\rho}{2}\|y_i - y^{k-1}\|_2^2 \right\}, \quad (8)$$

which gives the solution,

$$y_i^k = (H_i^k + \alpha I + \rho I)^{-1}(g_i^k - \lambda_i^{k-1} + \rho y^{k-1}). \quad (9)$$

Then, every client transmits its updated local variable y_i^k to the PS.

(2) The primal variable of the PS is updated by solving the following problem

$$y^k = \arg\min_{y} \left\{ \sum_{i=1}^n \langle \lambda_i^{k-1}, y_i^k - y\rangle + \frac{\rho}{2}\sum_{i=1}^n \|y_i^k - y\|_2^2 \right\}, \quad (10)$$

Which gives the solution,

$$y^k = \frac{1}{n}\sum_{i=1}^n (y_i^k + \lambda_i^{k-1}/\rho). \quad (11)$$

Once the global variable y^k is updated at the PS, it will be shared with all clients.

(3) The dual variables are updated locally for each client

$$\lambda_i^k = \lambda_i^{k-1} + \rho(y_i^k - y^k). \quad (12)$$

From (11) and (12), we know that $\sum_{i=1}^n \lambda_i^k = 0, \forall k$, hence (11) can be written as

$$y^k = \frac{1}{n}\sum_{i=1}^n y_i^k. \quad (13)$$

Therefore, the global variable y^k is just the average of the local variables y_i^k, for all $i \in [n]$.

Algorithm 1 FedNew (**Fed**erated **New**ton)

```
1: Parameters: $K$; $\rho$, $\alpha$
2: Initialization: $x^0, y^0, \{y_i^0\}_{i=1}^n, \{\lambda_i^0\}_{i=1}^n \in \mathbb{R}^d$
3: $k \leftarrow 0$
4: while $k < K$ do
5:     on all clients: compute local gradient $g_i^k$ and local Hessian $H_i^k$.
6:     on all clients: update $y_i^k$ using (9) and send to the PS
7:     on the PS: update $\boldsymbol{y}^k$ using (13) and $x^k$ using (14) and transmit them to all clients
8:     on all clients: update $\lambda_i^k$ using (12)
9: end while
```

2.2. Outer Level: Approximate Newton Step

At the outer level, after calculating the global direction y^k from (13), we perform the following update

$$x^{k+1} = x^k - y^k, \quad (14)$$

where y^k is an approximation to the optimal direction $y^*(x^k)$. We note that the closed-form expression in (9) still involves an inversion of the local Hessian H_i^k, which can be avoided by using H_i^0 instead of H_i^k at each iteration. To summarize, at iteration k, given g_i^k and H_i^k (H_i^0 if we avoid inversion), each client updates and transmits y_i^k. Then the PS updates y^k, and performs a one Newton step before sharing the updated model and y^k with all the clients. Finally, each client updates the dual variables via (12). The detailed steps of the algorithm are summarized in Algorithm 1. Using ADMM to solve the inner level optimization problem introduces a new set of variables (local primal and dual variables), and by performing only one ADMM pass each time, the subproblems are not solved optimally, which may affect the convergence of the outer iterate x^k in our framework. Therefore, it becomes essential to study the convergence behavior of the inner iterates so that outer iterates eventually follows the newton directions only. Proving convergence for such a combination between a primal based method (Newton method) for the outer problem and a primal-dual based method with one pass only each iteration (one-pass ADMM method) for the inner problem is very challenging. However, in the next section, we provide convergence analysis of the proposed framework and show that convergence holds under some assumptions and conditions.

3. Convergence Analysis

In this section, we study the convergence of the proposed FedNew algorithm. Note that we have introduced a two level framework to approximate the update in (4). The solution of (5) would result in the optimal direction, which we then use to update the model parameter at the PS. Since

(5) is strictly convex, there exists a unique optimal solution which we can write in closed form, but it requires that all clients transmit their hessian matrices to the PS, where the PS performs summation of the local Hessians, and then matrix inversion, which is costly, and we want to avoid. The path we took is to use an iterative distributed ADMM algorithm to solve the inner level problem in (6). There are two possibilities, the first is that we solve the inner problem iteratively till convergence, and then utilize the solution for the Newton update. In this case, x^k would trivially converge to x^* for $\alpha = 0$ following the existing analysis in the literature (Karimireddy et al., 2018; Marteau-Ferey et al., 2019), but it would add a lot of computational burden on the clients and is not very practical solution to aim for. In contrast, we propose to solve the inner problem via one step ADMM (where we just take one step of standard ADMM), and then perform the outer Newton update. Note that this approach is really effective for practical use, but introduces errors in the Newton directions to be used for the outer update. This makes the convergence proof challenging and does not hold straightforwardly from the existing proof of the ADMM or Newton based methods. To address this challenge, we develop a proof technique next to show that FedNew converges asymptotically.

To analyze the convergence behavior of Algorithm 1, we assume that each function f_i is twice differentiable, convex, has L-Lipschitz continuous gradient, and L_*-Lipschitz continuous Hessian. Let $Q_i(x,y)$ be equal to $\frac{1}{2}y_i^T(\nabla^2 f_i(x^k)+\alpha I)y_i - y_i^T\nabla f_i(x^k)$, with this definition, we assume that ∇Q is Lipschitz continuous in y with constant L_q, i.e., for any given $x^1, x^2 \in X$, we have

$$\|\nabla Q_i(x^1,y_i^1) - \nabla Q_i(x^2,y_i^2)\|^2 \le L_q\|y_i^1 - y_i^2\|^2. \quad (15)$$

We start with the necessary and sufficient optimality conditions of the inner problem in (6) at the k-th iteration (the k-th Newton step), which are the primal and dual feasibility conditions (Boyd et al., 2011) defined as

$$y_i^\star(x^k) = y^\star(x^k), \qquad \text{(primal feas.)} \quad (16)$$
$$(H_i^k + \alpha I)y_i^\star(x^k) - g_i^k + \lambda_i^\star(x^k) = 0, \qquad \text{(dual feas.)} \quad (17)$$

for all $i \in [n]$. In (16)-(17), $y_i^\star(x^k)$ and $\lambda_i^\star(x^k)$ denote the optimal values of y_i^k and λ_i^k, respectively, at the k^{th} iteration, i.e. when running the ADMM steps to the end. Note that if $y_i^\star(x^k)$ is computed then $x^{k+1} = x^k - \frac{1}{n}\sum_{i=1}^n y_i^{\star k}$. Next, we write the optimality conditions at the optimal model x^* as

$$y_i^\star = y^\star = 0, \forall i \in [n] \qquad \text{(primal feas.)} \quad (18)$$
$$g_i^\star = \lambda_i^\star, \forall i \in [n] \qquad \text{(dual feas.)} \quad (19)$$

where g_i^k becomes $g_i^\star$ in (19). The equality in (18) and (19) follows from the fact that $y^\star = \nabla^2[f(x^\star) + \alpha I]^{-1}\nabla f(x^\star) = 0$ because $\nabla f(x^\star) = 0$. Note that we drop the k notation in (18)-(19) because we write them for x^* which is independent of k. Next, to simplify the analysis, we introduce a local copy x_i of the model x to write

$$x^{k+1} = \frac{1}{n}\sum_{i=1}^n x_i^{k+1} = \frac{1}{n}\sum_{i=1}^n (x_i^k - y_i^k). \quad (20)$$

Re-writing (14) as (20) does not change anything in the algorithm but simplifies the analysis. With this, we first introduce two intermediate lemmas (Lemma 1 and Lemma 2) which would lead us to Theorem 1.

Lemma 1. *In Algorithm 1 (FedNew), for each iteration k, it holds that*

$$\sum_{i=1}^n \langle \lambda_i^k + s^k - \lambda_i^{\star k}, y_i^k - y^{\star k}\rangle \le -\alpha \sum_{i=1}^n \|y_i^k - y^{\star k}\|^2, \quad (21)$$

where $s^k = \rho(y^k - y^{k-1})$ is the dual residual of the inner problem.

The proof of Lemma 1 is provided in Appendix A. The result in Lemma 1 will be used in Lemma 2 to impose an upper bound on the difference between the optimality gap of the inner problem at iteration k and $k-1$.

Lemma 2. *In Algorithm 1 (FedNew), for each iteration k, it holds that*

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^k - \lambda_i^{\star k}\|^2 + 2\beta_1 \sum_{i=1}^n \|y_i^k - y^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 \\
&+ 2\rho n\|y^k - y^{k-1}\|^2 \\
&\le \frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^{k-1} - \lambda_i^{\star k-1}\|^2 + \frac{2L_q^2}{\rho}\sum_{i=1}^n \|y_i^{k-1} - y^{\star k-1}\|^2 \\
&\quad + \frac{4L_q^2 n}{\rho}\|y^{k-1} - y^{\star k-1}\|^2 + 2\rho n\|y^{k-1} - y^{k-2}\|^2 \\
&\quad - 2\beta_2 \sum_{i=1}^n \|y_i^k - y^{\star k}\|^2, \qquad (22)
\end{aligned}$$

where $\beta_1 > 0, \beta_2 > 0$, and

$$\beta_1 + \beta_2 \le \alpha - 2.5\rho - \frac{8L_q^2 n}{\rho}. \quad (23)$$

The proof of Lemma 2 is provided in Appendix B. Now, we are ready to state the main theoretical result of this work. To proceed towards the main theorem, we define Lyapunov function V^k as

$$\begin{aligned}
V^k := &\frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^k - \lambda_i^{\star k}\|^2 + 2\beta_1 \sum_{i=1}^n \|y_i^k - y^{\star k}\|^2 \\
&+ \rho n\|y^k - y^{\star k}\|^2 + 2\rho n\|y^k - y^{k-1}\|^2, \qquad (24)
\end{aligned}$$

which quantifies the distance to the optimal for the dual variable λ_i^k, and the primal variable y^k. We present the main result in Theorem 1.

Theorem 1. *With α that satisfies (23) for $\beta_1 \geq \frac{L_q^2}{\rho}$, $\rho \geq 2L_q$, and $\beta_2 > 0$, the sequence of iterates of FedNew (cf. Algorithm 1) are such that $\lambda_i^k \rightarrow \lambda_i^{\star k}$, $y_i^k \rightarrow y^{\star k}$, and $y^k \rightarrow y^{\star k}$ as $k \rightarrow \infty$.*

Proof. The detailed proof of Theorem 1 is provided in Appendix C. To prove the asymptotic convergence of the proposed algorithm, we utilize the results of Lemma 1 and Lemma 2, and show that the Lyapunov function V^k is monotonically decreasing for each k. Subsequently, we utilize the Monotone Convergence Theorem to claim that V^k converges to zero as $k \rightarrow \infty$. This further implies that $\lambda_i^k \rightarrow \lambda_i^{\star k}$, $y_i^k \rightarrow y^{\star k}$, and $y^k \rightarrow y^{\star k}$ as $k \rightarrow \infty$. □

We remark that even though we obtain the approximate inexact Newton directions via approximating the solution of the distributed optimization problem formulated in (6) at each iteration, we will asymptotically move in the inexact Newton direction, which is a descent direction. One can utilize the global convergence analysis of inexact Newton methods in (Karimireddy et al., 2018; Marteau-Ferey et al., 2019) to further establish the global convergence of FedNew and we leave that to future scope of this work. Finally, it is worth mentioning that replacing H_i^k with H_i^0 in the formulation (6) and Algorithm 1 does not change the results of Lemmas 1 and Lemma 2 since they both require positive definiteness of the Hessian, which already is a property of H_i^0. With this, it is easy to show that Theorem 1 result also holds for the computation-efficient implementation of FedNew.

4. Privacy Analysis

Revealing the local gradient is vulnerable to model inversion, and reconstruction attacks (Fredrikson et al., 2015; Hitaj et al., 2017). In addition, revealing the Hessian increases vulnerability, since more information about the local function/data is released. These attacks infer the statistical profiles of training samples and violate data privacy. Against such an adversarial inverse problem, we aim at preserving privacy defined as follows.

Definition 1 (Zhang et al., 2018) A mechanism M : $M(X) \rightarrow Y$ is defined to be privacy preserving if the input X cannot be uniquely derived from the output Y.

We treat X as local gradient/Hessian to be protected, and consider Y as the known information at an eavesdropper such as the PS or another client. At iteration k, under the standard Newton's method, the PS receives the gradient g_i^k and the Hessian H_i^k from each client i, thereby violating the privacy defined in Definition 1. In sharp contrast, the PS in FedNew receives y_i^k, which is neither the gradient nor the Hessian. Consequently, we ensure that privacy is preserved against curious PS and against any eavesdropper with the knowledge of the updating trajectory of the variable y. The following theorem formally states that FedNew preserves the privacy of the local gradients/Hessians.

Theorem 2. *At each iteration $k \geq 0$, FedNew preserves the privacy of each local gradient update g_i^k and local zero-Hessian H_i^k.*

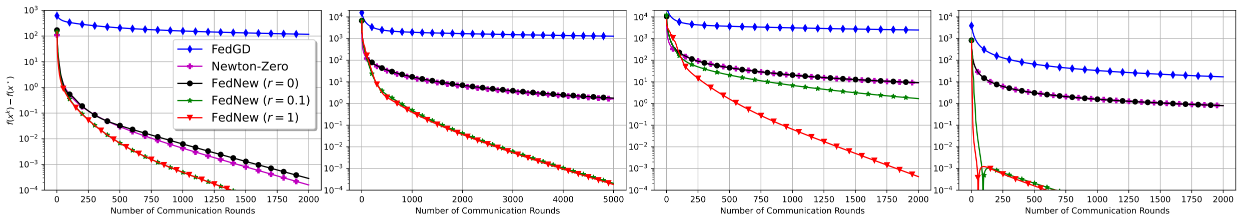
Proof. The eavesdropper needs to solve (9) with respect to H_i^k and g_i^k. However, this single equation has three unkowns at the reciever which are H_i^k, g_i^k, and λ_i^{k-1}. Hence, the receiver, the eavesdropper cannot have a unique solution for H_i^k, g_i^k since the number of variables $V = 3$ is greater than the number of equations $E = 1$. This finalizes the proof. □

The key point of the proof is to show that the inverse problem of an eavesdropper boils down to solving a set of equations at every iteration, in which the number of unknowns is larger than the number of equations. Therefore, each client's local gradient/Hessian cannot be uniquely derived.

5. Quantized FedNew

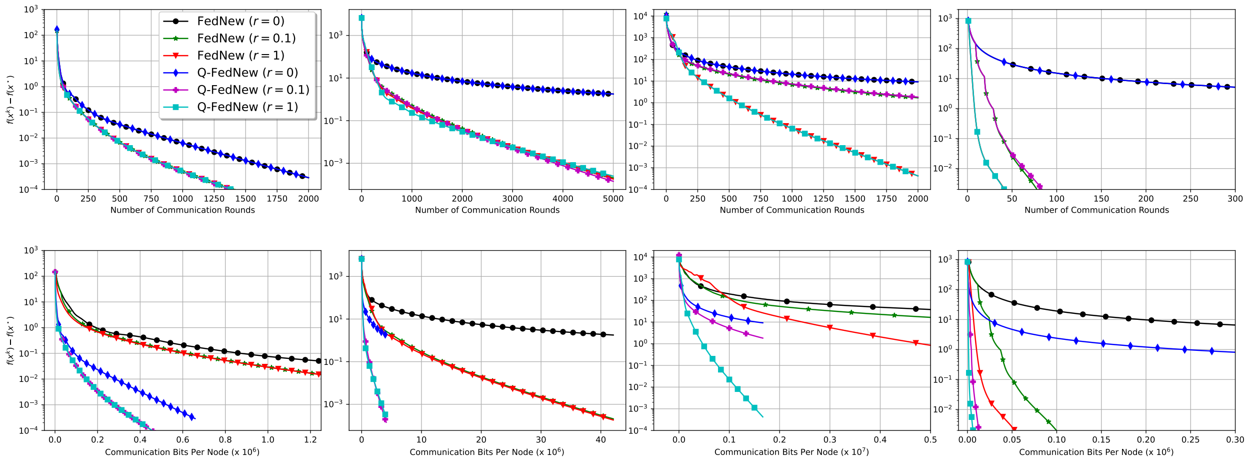
Similar to first-order methods, FedNew allows each client to transmit a vector whose size is equivalent to the model size. At the PS, all received vectors need only to be aggregated and averaged. Therefore, in contrast to other second-order methods, the received information from all clients at the PS is not involved in any further multiplication and/or inversion. Note that such multiplication and/or inversion may make existing quantization schemes used in first-order methods no longer applicable. Hence, by utilizing this feature of FedNew, we are able to further quantize the transmitted variable (y_i^k) using quantization schemes used in first-order methods in the literature (Elgabli et al., 2020), and numerically show the convergence of the proposed approach. With quantization, we can significantly reduce the communication overhead per iteration. We refer to the quantized version of FedNew by Q-FedNew.

Next, we describe the quantization procedure of Q-FedNew. At iteration k, each client i quantizes the difference between y_i^k and the previously quantized vector $\hat{y}_i^{k-1}$ as $y_i^k - \hat{y}_i^{k-1} = Q_i(y_i^k, \hat{y}_i^{k-1})$. The function $Q_i(\cdot)$ is a stochastic quantization operator that depends on the quantization probability $p_{i,j}^k$ for each element $j \in \{1, 2, \cdots, d\}$, and on b_i^k the bits used to representing each element. We choose $p_{i,j}^k$ and b_n^k as follows. The j-th element $[\hat{y}_i^{k-1}]_j$ of the previously quantized model vector is centered at the quantization range $2R_i^k$ that is equally divided into



(a) Dataset: a1a (b) Dataset: w7a (c) Dataset: w8a (d) Dataset: phishing

Figure 1. Optimality gap of FedNew compared to FedGD and Newton Zero in terms of the number of communication rounds per client for different datasets. FedNew($r = 0$) require close to Newton Zero's number of communication rounds for ϵ optimality gap, yet preserves the privacy. FedNew($r = 0.1$) and FedNew($r = 1$) achieve faster convergence.



(a) Dataset: a1a (b) Dataset: w7a (c) Dataset: w8a (d) Dataset: phishing

Figure 2. Optimality gap of Q-FedNew compared to FedNew in terms of number of communication rounds and number of communication bits per client for different datasets.Q-FedNew converges as fast as FedNew, but at significantly less number of transmitted bits.

$2^{b_i^k} - 1$ quantization levels, yielding the quantization step size $\Delta_i^k = 2R_i^k/(2^{b_i^k} - 1)$. In this coordinate, the difference between $[y_i^k]_j$ and $[\hat{y}_i^{k-1}]_j$ is

$$[c_i(y_i^k)]_j = \frac{1}{\Delta_i^k}\left([y_i^k]_j - [\hat{y}_i^{k-1}]_j + R_i^k\right), \tag{25}$$

where adding R_i^k ensures the non-negativity of the quantized value. Then, $[c_i(y_i^k)]_j$ is mapped to $[q_n(y_i^k)]_j$ as

$$[q_n(y_i^k)]_j = \begin{cases} \lceil [c_i(y_i^k)]_j \rceil & \text{with prob. } p_{i,j}^k \\ \lfloor [c_i(y_i^k)]_j \rfloor & \text{with prob. } 1 - p_{i,j}^k \end{cases} \tag{26}$$

where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ are ceiling and floor functions, respectively. Next, we select the probability $p_{i,j}^k$ in (26) such that the expected quantization error is $\mathbb{E}\left[\epsilon_{i,j}^k\right]$ is zero which implies that

$$\begin{aligned} &p_{i,j}^k \left([c_i(y_i^k)]_j - \lceil [c_i(y_i^k)]_j \rceil\right) \\ &+ (1 - p_{i,j}^k)\left([c_i(y_i^k)]_j - \lfloor [c_i(y_i^k)]_j \rfloor\right) = 0. \end{aligned} \tag{27}$$

Solving (27) for $p_{i,j}^k$, we obtain

$$p_{i,j}^k = \left([c_i(y_i^k)]_j - \lfloor [c_i(y_i^k)]_j \rfloor\right). \tag{28}$$

The $p_{i,j}^k$ selection in (28) ensures that the quantization error is unbiased, yielding the quantization error variance $\mathbb{E}\left[\left(\epsilon_{i,j}^k\right)^2\right] \leq (\Delta_i^k)^2/4$ (Reisizadeh et al., 2019). This implies that $\mathbb{E}\left[{\epsilon_n^k}^2\right] \leq d(\Delta_i^k)^2/4$. With the aforementioned stochastic quantization procedure, b_i^k, R_i^k, and $q_i(y_i^k)$ suffice to represent $\hat{y}_i^k$, where

$$q_i(y_i^k) = ([q_i(y_i^k)]_1, [y_i^k]_2, \cdots, [y_i^k)]_d)^{\prime} \tag{29}$$

are transmitted to the PS. After receiving these values at the PS, $\hat{y}_i^k$ can be reconstructed as follows:

$$\hat{y}_i^k = \hat{y}_i^{k-1} + \Delta_i^k q_i(y_i^k) - R_i^k \mathbf{1}. \tag{30}$$

In contrast to full arithmetic precision based communication which uses $32d$ bits to represent the transmitted vector, every transmission payload size of Q-FedNew is $b_i^k d + b_R$ bits, where $b_R \leq 32$ is the required bits to represent R_i^k.

6. Experiments

In this section, we empirically investigate the performance of the proposed algorithms FedNew and Q-FedNew against FedGD (McMahan et al., 2017b), and Newton Zero (Safaryan et al., 2021) for a binary classification problem using regularized logistic regression[1]. We consider 3 variants of FedNew. To explain these variants, we let r be the update rate of the hessian matrix. We consider $r \in \{0, 0.1, 1\}$. i.e., $r = 0$, reflects the case when the hessian matrix is not updated at all. Therefore, H_i^0 is used at each iteration similar to Newton zero. $r = 0.1$ reflects the case when the hessian is updated every 10th iteration. Finally, $r = 1$ reflects the case when the hessian is updated at each iteration, so H_i^k is used at iteration k. As we will show later in the section, updating the hessian matrix at each iteration improves the convergence speed, but at the cost of more computations. We first describe the experimental setup and then discuss the numerical results.

6.1. Experimental Setup

In our experiments, we consider the regularized logistic regression problem

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) + \frac{\mu}{2} \|x\|^2 \right\}, \qquad (31)$$

where the local loss functions are defined as

$$f_i(x) = \frac{1}{m} \sum_{j=1}^{m} \log\left(1 + \exp(-b_{ij} a_{ij}^\top x)\right), \qquad (32)$$

$\{a_{ij}, b_{ij}\}_{j \in [m]}$ forms the data samples of the ith client, and $\mu \geq 0$ is a regularization parameter chosen to be equal to 10^{-3} in all experiments conducted in this section. We use four standard datasets taken from the LibSVM library (Chang & Lin, 2011): a1a, w7a, w8a, and phishing. More details on each dataset, including the number of independent features and the number of clients considered, are summarized in Table 1, where $N = m \times n$ is the total number of samples. In our experiments, each dataset is evenly split between the clients. Note that we have chosen different numbers of clients for each of the datasets to show the performance of the proposed approach under various network sizes. For Q-FedNew, the quantization resolution is 3 bits in all experiments. In the experiments, we plot the optimality gap $f(x^k) - f(x^\star)$ as a function of the number of communication rounds or the number of communicated bits per client, where $f(x^\star)$ is the function value at the 30th iterate of the standard Newton's method. Finally, for each variant of our algorithm, we choose α and ρ that give the fastest convergence in the tested range. We would like to mention that though we theoretically prove convergence for α that satisfies (23), we observe that empirically FedNew converges for any choice of $\alpha \geq 0$.

[1]The code is avaialble at `https://github.com/aelgabli/FedNew`.

6.2. Comparison to Baselines

In Fig. 1, we plot the optimality gap as a function of the number of communication rounds for FedNew and the two baselines: FedGD and Newton Zero. Fig. 1 shows that FedNew-($r = 1$) is the fastest to converge compared to the other algorithms in terms of the number of communication rounds, followed by FedNew-($r = 0.1$), then both Newton Zero and FedNew-($r = 0$), and finally FedGD. In conclusion, when using the updated hessian at each iteration, FedNew can achieve significant reduction in terms of the number of communication rounds, but at the cost of more computations. On the other hand, FedNew-($r = 0$) which avoids updating the hessian can match the convergence speed of Newton Zero while preserving privacy. Periodic update of the hessian ($r = 0.1$) achieves a middle point. In fact, as shown in Fig. 1, in some of the datasets, it converges as fast as the exact hessian based FedNwe ($r = 1$) while reducing the computation cost 10 times since the hessian is computed once every 10 iterations.

Fig. 2 compares Q-FedNew to FedNew. As shown in the figure, for a fixed number of communication rounds, Q-FedNew achieves the same optimality gap as FedNew. However, when the optimality gap is plotted against the number of communicated bits per client, we can clearly see the significant savings in terms of the number of communicated bits per client that Q-FedNew achieves compared to FedNew. For example, for the dataset w8a, Q-FedNew($r = 1$) requires almost $10\times$ less number of transmitted bits compared to FedNew($r = 1$) to achieve the optimality gap of 10^{-3}.

Table 1. Description of the datasets

Dataset	N	m	d	n
a1a	1600	160	99	10
w7a	24640	308	263	80
w8a	49700	829	267	60
phishing	11040	276	40	40

Finally, we observe, from Figs. 1 and 2, that Newton Zero starts at a large number of communicated bits since it requires every client to send the whole Hessian matrix at the first iteration, which consumes a large number of transmitted bits.

7. Conclusion and Future Work

We proposed a novel communication-efficient, and privacy-preserving federated learning framework based on Newton and ADMM methods. Unlike existing approaches, the proposed approach (FedNew) does not require clients to transmit their Hessian or its compressed version at any iteration. Moreover, the proposed approach ensures privacy by hiding the gradient and the Hessian information. FedNew achieves the same communication-efficiency of first-order methods per iteration, while enjoying faster convergence and preserving privacy. In particular, the *inverse Hessian-gradient product* alternates between updating the *inverse Hessian-gradient product* using only one ADMM step at each Newton's iteration, and updating the global model using Newton's method. FedNew is proved to follow the inexact Newton directions asymptotically. The non-asymptotic version of the proof of optimality is left to the future scope of this work. Furthermore, a significant reduction in communication overhead is achieved by utilizing stochastic quantization. Numerical results show the superiority of FedNew compared to existing methods in terms of communication costs while ensuring privacy. Future works will explore the convergence analysis of the quantized version of FedNew (Q-FedNew), and extend the current framework to fully decentralized topology.

8. Acknowledgement

This work is supported by Academy of Finland SMARTER, EU-ICT IntellIoT, EU-CHISTERA LeadingEdge, and CONNECT, Infotech-NOOR and NEGEIN.

References

Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. QSGD : Communication-efficient SGD via gradient quantization and encoding. *Advances in Neural Information Processing Systems*, 30:1709–1720, 2017.

Beck, A. *Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB*. SIAM, 2014.

Bergou, E. H., Diouane, Y., and Kungurtsev, V. Convergence and complexity analysis of a levenberg–marquardt algorithm for inverse problems. *Journal of Optimization Theory and Applications*, 185(3):927–944, 2020.

Boyd, S., Parikh, N., and Chu, E. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

Chang, C.-C. and Lin, C.-J. LIBSVM: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Chen, T., Giannakis, G., Sun, T., and Yin, W. Lag: Lazily aggregated gradient for communication-efficient distributed learning. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Elgabli, A., Park, J., Bedi, A. S., Ben Issaid, C., Bennis, M., and Aggarwal, V. Q-GADMM: Quantized group ADMM for communication efficient decentralized machine learning. *IEEE Transactions on Communications*, 69(1):164–181, 2020.

Fredrikson, M., Jha, S., and Ristenpart, T. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1322–1333, 2015.

Gower, R. M., Loizou, N., Qian, X., Sailanbayev, A., Shulgin, E., and Richtárik, P. SGD: General analysis and improved rates. In *International Conference on Machine Learning (ICML)*, pp. 5200–5209, 2019.

Hitaj, B., Ateniese, G., and Perez-Cruz, F. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 603–618, 2017.

Issaid, C. B., Elgabli, A., Park, J., Bennis, M., and Debbah, M. Communication efficient decentralized learning over bipartite graphs. *IEEE Transactions on Wireless Communications*, pp. 1–1, 2021.

Karimireddy, S. P., Stich, S. U., and Jaggi, M. Global linear convergence of newton's method without strong-convexity or lipschitz gradients. *arXiv preprint arXiv:1806.00413*, 2018.

Khaled, A., Mishchenko, K., and Richtárik, P. Tighter theory for local SGD on identical and heterogeneous data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 4519–4529, 2020.

Khirirat, S., Feyzmahdavian, H. R., and Johansson, M. Distributed learning with compressed gradients. *arXiv preprint arXiv:1806.06573*, 2018.

Konečnỳ, J., McMahan, H. B., Yu, F. X., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. In *NIPS Private Multi-Party Machine Learning Workshop*, 2016.

Levenberg, K. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.

Malitsky, Y. and Mishchenko, K. Adaptive gradient descent without descent. In *International Conference on Machine Learning (ICML)*, pp. 6702–6712, 2019.

Marquardt, D. W. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

Marteau-Ferey, U., Bach, F., and Rudi, A. Globally convergent newton methods for ill-conditioned generalized self-concordant losses. *Advances in Neural Information Processing Systems*, 32, 2019.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1273–1282, 2017a.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017b.

Mishchenko, K. Regularized newton method with global $o(1/k2)$ convergence. *arXiv preprint arXiv:2112.02089*, 2021.

Mishchenko, K., Gorbunov, E., Takáč, M., and Richtárik, P. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019.

Mishchenko, K., Khaled, A., and Richtárik, P. Proximal and federated random reshuffling. *arXiv preprint arXiv:2102.06704*, 2021a.

Mishchenko, K., Wang, B., Kovalev, D., and Richtárik, P. IntSGD: Floatless compression of stochastic gradients. *arXiv preprint arXiv:2102.08374*, 2021b.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Reisizadeh, A., Mokhtari, A., Hassani, H., and Pedarsani, R. An exact quantized decentralized gradient descent algorithm. *IEEE Transactions on Signal Processing*, 67 (19):4934–4947, 2019.

Safaryan, M., Islamov, R., Qian, X., and Richtárik, P. FedNL: Making Newton-type methods applicable to federated learning. *arXiv preprint arXiv:2106.02969*, 2021.

Stich, S. U. Local SGD converges fast and communicates little. In *International Conference on Learning Representations (ICLR)*, 2020.

Sun, J., Chen, T., Giannakis, G., and Yang, Z. Communication-efficient distributed learning via lazily aggregated quantized gradients. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Xie, C., Koyejo, O., Gupta, I., and Lin, H. Local AdaAlter: Communication-efficient stochastic gradient descent with adaptive learning rates. *arXiv preprint arXiv:1911.09030*, 2019.

Yin, X., Ng, B. W.-H., He, J., Zhang, Y., and Abbott, D. Accurate image analysis of the retina using hessian matrix and binarisation of thresholded entropy with application of texture mapping. *PLOS ONE*, 9(4):1–17, 04 2014.

Zhang, C., Ahmad, M., and Wang, Y. Admm based privacy-preserving decentralized optimization. *IEEE Transactions on Information Forensics and Security*, 14(3):565–580, 2018.

Zhang, J., Ling, Q., and So, A. M.-C. A newton tracking algorithm with exact linear convergence for decentralized consensus optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 7:346–358, 2021.

Zhize Li, Dmitry Kovalev, X. Q. and Richtárik, P. Acceleration for compressed gradient descent in distributed and federated optimization. In *International Conference on Machine Learning (ICML)*, pp. 5895–5904, 2020.

Appendices

A. Proof of Lemma 1

At iteration k, and given x^k, we run the one step ADMM to evaluate the direction y_i^k at each client i. From the first order optimality condition of the problem in (8), we can write

$$(H_i^k + \alpha I)y_i^k - g_i^k + \lambda_i^{k-1} + \rho(y_i^k - y^{k-1}) = 0. \tag{33}$$

We add and subtract y^k in the left hand side in the above expression and utilize the dual udpate $\lambda_i^k = \lambda_i^{k-1} + \rho(y_i^k - y^k)$ (cf. (12)) in (33) to get

$$(H_i^k + \alpha I)y_i^k - g_i^k + \lambda_i^k + \rho(y^k - y^{k-1}) = 0. \tag{34}$$

Let s^k be the dual residual at iteration k defined as

$$s^k = \rho(y^k - y^{k-1}) \tag{35}$$

Therefore, (34) can be re-written as

$$(H_i^k + \alpha I)y_i^k - g_i^k + \lambda_i^k + s^k = 0. \tag{36}$$

Re-arranging the terms, we get

$$\lambda_i^k + s^k = g_i^k - (H_i^k + \alpha I)y_i^k. \tag{37}$$

Subtracting $\lambda_i^{\star k}$ from both sides and using (17), we obtain

$$\lambda_i^k + s^k - \lambda_i^{\star k} = (H_i^k + \alpha I)(y_i^{\star k} - y_i^k) \tag{38}$$

Multiplying both sides with $y_i^k - y_i^{\star k}$, we obtain,

$$\langle \lambda_i^k + s^k - \lambda_i^{\star k}, y_i^k - y_i^{\star k} \rangle = \langle (H_i^k + \alpha I)y_i^{\star k} - y_i^k, y_i^k - y_i^{\star k} \rangle \tag{39}$$

Since H_i^k is a positive semidefinite. i.e., $\langle H_i^k(y_i^{\star k} - y_i^k), y_i^k - y_i^{\star k} \rangle \leq 0$, we can re-write (39) as

$$\langle \lambda_i^k + s^k - \lambda_i^{\star k}, y_i^k - y^{\star k} \rangle \leq -\alpha \|y_i^k - y_i^{\star k}\|^2 \tag{40}$$

Using $y_i^{\star k} = y^{\star k}$, and summing over all clients, and we obtain

$$\sum_{i=1}^{n} \langle \lambda_i^k + s^k - \lambda_i^{\star k}, y_i^k - y^{\star k} \rangle \leq -\alpha \sum_{i=1}^{n} \|y_i^k - y^{\star k}\|^2 \tag{41}$$

That concludes the proof.

B. Proof of Lemma 2

We start with the statement of Lemma 1 and multiply both sides of (41) by 2 to obtain

$$2A + 2B \leq -2\alpha \sum_{i=1}^{n} \|y_i^k - y^{\star k}\|^2, \tag{42}$$

where $A := \sum_{i=1}^n \langle \lambda_i^k - \lambda_i^{\star k}, y_i^k - y^{\star k} \rangle$, and $B := \sum_{i=1}^n \langle s^k, y_i^k - y^{\star k} \rangle$. First, we focus on the first term on the left side of (42). Note that since $\sum_{i=1}^n \lambda_i^k = \sum_{i=1}^n \lambda_i^{\star k} = 0$, term A can be written as $A = \sum_{i=1}^n \langle \lambda_i^k - \lambda_i^{\star k}, y_i^k \rangle$. Hence, we can write

$$\begin{aligned}
2A &= 2\sum_{i=1}^n \langle \lambda_i^k - \lambda_i^{\star k}, y_i^k \rangle \\
&\overset{(a)}{=} 2\sum_{i=1}^n \langle \lambda_i^{k-1} + \rho y_i^k - \rho y^k - \lambda_i^{\star k}, y_i^k \rangle \\
&= 2\sum_{i=1}^n \langle \lambda_i^{k-1} - \rho y^k - \lambda_i^{\star k}, y_i^k \rangle + 2\rho \sum_{i=1}^n \|y_i^k\|^2 \\
&\overset{(b)}{=} \frac{2}{\rho}\sum_{i=1}^n \langle \lambda_i^{k-1} - \rho y^k - \lambda_i^{\star k}, \lambda_i^k - \lambda_i^{k-1} + \rho y^k \rangle + \rho \sum_{i=1}^n \|y_i^k\|^2 + \rho \sum_{i=1}^n \|y_i^k\|^2 \\
&= \frac{2}{\rho}\sum_{i=1}^n \langle \lambda_i^{k-1} - \rho y^k - \lambda_i^{\star k}, \lambda_i^k - \lambda_i^{\star k} + \lambda_i^{\star k} - \lambda_i^{k-1} + \rho y^k \rangle + \frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^k - \lambda_i^{k-1} + \rho y^k\|^2 + \rho \sum_{i=1}^n \|y_i^k\|^2 \\
&= -\frac{2}{\rho}\sum_{i=1}^n \|\lambda_i^{k-1} - \rho y^k - \lambda_i^{\star k}\|^2 + \frac{2}{\rho}\sum_{i=1}^n \langle \lambda_i^{k-1} - \rho y^k - \lambda_i^{\star k}, \lambda_i^k - \lambda_i^{\star k} \rangle + \frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^k - \lambda_i^{k-1} + \rho y^k\|^2 + \rho \sum_{i=1}^n \|y_i^k\|^2,
\end{aligned} \tag{43}$$

where we used the update of the dual variables, i.e. $\lambda_i^k = \lambda_i^{k-1} + \rho y_i^k - \rho y^k$ in (a), and replaced y_i^k by $(\lambda_i^k - \lambda_i^{k-1} + \rho y^k)/\rho$ in the first two terms of (b). Now, we use $\lambda_i^k - \lambda_i^{k-1} = (\lambda_i^k - \lambda_i^{\star k}) - (\lambda_i^{k-1} - \lambda_i^{\star k})$ in the third term of (43) to write

$$\begin{aligned}
\frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^k - \lambda_i^{k-1} + \rho y^k\|^2 &= \frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^k - \lambda_i^{\star k} - \left(\lambda_i^{k-1} - \lambda_i^{\star k} - \rho y^k\right)\|^2 \\
&= \frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^k - \lambda_i^{\star k}\|^2 + \frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^{k-1} - \lambda_i^{\star k} - \rho y^k\|^2 - \frac{2}{\rho}\sum_{i=1}^n \langle \lambda_i^{k-1} - \lambda_i^{\star k} - \rho y^k, \lambda_i^k - \lambda_i^{\star k} \rangle.
\end{aligned} \tag{44}$$

Utilizing the expression in (44) into (43), we get

$$\begin{aligned}
2A &= \frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^k - \lambda_i^{\star k}\|^2 - \frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^{k-1} - \lambda_i^{\star k} - \rho y^k\|^2 + \rho \sum_{i=1}^n \|y_i^k\|^2 \\
&= \frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^k - \lambda_i^{\star k}\|^2 - \frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^{k-1} - \lambda_i^{\star k}\|^2 - \rho n \|y^k\|^2 + \rho \sum_{i=1}^n \|y_i^k\|^2
\end{aligned} \tag{45}$$

Next, we tackle the term B. Replacing the dual residual s^{k+1} by its definition, we get

$$\begin{aligned}
2B &= 2\rho \sum_{i=1}^n \langle y^k - y^{k-1}, y_i^k - y^{\star k} \rangle = 2\rho \sum_{i=1}^n \langle y^k - y^{k-1}, y_i^k - y^k + y^k - y^{\star k} \rangle \\
&= 2\rho \sum_{i=1}^n \langle y^k - y^{k-1}, r_i^k \rangle + 2\rho n \langle y^k - y^{k-1}, y^k - y^{\star k} \rangle,
\end{aligned} \tag{46}$$

where we utilized the definition $r_i^k = y_i^k - y^k$. Using $y^k - y^{\star k} = y^k - y^{k-1} + y^{k-1} - y^{\star k}$, we can write

$$2B = 2\rho \sum_{i=1}^n \langle y^k - y^{k-1}, r_i^k \rangle + 2\rho n \|y^k - y^{k-1}\|^2 + 2\rho n \langle y^k - y^{k-1}, y^{k-1} - y^{\star k} \rangle. \tag{47}$$

Next, using $y^k - y^{k-1} = y^k - y^{\star k} - (y^{k-1} - y^{\star k})$, we can write

$$\begin{aligned}2B &= 2\rho\sum_{i=1}^{n}\langle y^k - y^{k-1}, r_i^k\rangle + 2\rho n\|y^k - y^{k-1}\|^2 - 2\rho\|y^{k-1} - y^{\star k}\|^2 + 2\rho n\langle y^k - y^{\star k}, y^{k-1} - y^{\star k}\rangle \\ &= 2\rho\sum_{i=1}^{n}\langle y^k - y^{k-1}, r_i^k\rangle + \rho n\|(y^k - y^{\star k}) - (y^{k-1} - y^{\star k})\|^2 + \rho n\|y^k - y^{k-1}\|^2 - 2\rho\|y^{k-1} - y^{\star k}\|^2 \\ &\quad + 2\rho n\langle y^k - y^{\star k}, y^{k-1} - y^{\star k}\rangle \qquad (48) \\ &= \rho n\|y^k - y^{k-1}\|^2 + \rho n\|y^k - y^{\star k}\|^2 - \rho n\|y^{k-1} - y^{\star k}\|^2 + 2\rho\sum_{i=1}^{n}\langle y^k - y^{k-1}, r_i^k\rangle.\end{aligned}$$

Using the definition of r_i^k and the equality in (13), we have

$$\sum_{i=1}^{n}\langle y^k - y^{k-1}, r_i^k\rangle = \langle y^k - y^{k-1}, \sum_{i=1}^{n} r_i^k\rangle = \langle y^k - y^{k-1}, \sum_{i=1}^{n} y_i^k - ny^k\rangle = 0. \qquad (49)$$

Hence, from (48), we obtain

$$2B = \rho n\|y^k - y^{k-1}\|^2 + \rho n\|y^k - y^{\star k}\|^2 - \rho n\|y^{k-1} - y^{\star k}\|^2. \qquad (50)$$

Substituting (45) and (48) into (42) and rearrange terms, we obtain,

$$\begin{aligned}&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k - \lambda_i^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 \qquad (51) \\ &\le \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1} - \lambda_i^{\star k}\|^2 + \rho n\|y^{k-1} - y^{\star k}\|^2 - \rho n\|y^k - y^{k-1}\|^2 + \rho n\|y^k\|^2 - \rho\sum_{i=1}^{n}\|y_i^k\|^2 - 2\alpha\sum_{i=1}^{n}\|y_i^k - y^{\star k}\|^2\end{aligned}$$

using $n\sum_{i=1}^n \|y_i^k\|^2 \ge \|\sum_{i=1}^n y_i^k\|^2$, we can easily show that $\rho\sum_{i=1}^n \|y_i^k\|^2 \ge \rho n\|y^k\|^2$, so both terms cancel from the RHS of (51), and we obtain

$$\begin{aligned}&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k - \lambda_i^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 \\ &\le \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1} - \lambda_i^{\star k}\|^2 + \rho n\|y^{k-1} - y^{\star k}\|^2 - \rho n\|y^k - y^{k-1}\|^2 - 2\alpha\sum_{i=1}^{n}\|y_i^k - y^{\star k}\|^2. \qquad (52)\end{aligned}$$

Adding and subtracting λ_i^{k-1}, we get

$$\begin{aligned}&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k - \lambda_i^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 \\ &\le \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1} - \lambda_i^{\star k-1} + \lambda_i^{\star k-1} - \lambda_i^{\star k}\|^2 + \rho n\|y^{k-1} - y^{\star k}\|^2 - \rho n\|y^k - y^{k-1}\|^2 - 2\alpha\sum_{=1}^{n}\|y_i^k - y^{\star k}\|^2 \qquad (53)\end{aligned}$$

Using $\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2$, the following holds

$$\begin{aligned}&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k - \lambda_i^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 \qquad (54) \\ &\le \frac{2}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1} - \lambda_i^{\star k-1}\|^2 + \frac{2}{\rho}\sum_{i=1}^{n}\|\lambda_i^{\star k-1} - \lambda_i^{\star k}\|^2 + \rho n\|y^{k-1} - y^{\star k}\|^2 - \rho n\|y^k - y^{k-1}\|^2 - 2\alpha\sum_{=1}^{n}\|y_i^k - y^{\star k}\|^2.\end{aligned}$$

Using $\lambda_i^{\star k} = g_i^k - (H_i^k + \alpha^k I)y_i^{\star k}$, we get

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^k - \lambda_i^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 \\
&\leq \frac{2}{\rho}\sum_{i=1}^n \|\lambda_i^{k-1} - \lambda_i^{\star k-1}\|^2 + \frac{2}{\rho}\sum_{i=1}^n \|g_i^{k-1} - (H_i^{k-1} + \alpha^{k-1}I)y_i^{\star k-1} - g_i^k + (H_i^k + \alpha^k I)y_i^{\star k}\|^2 \\
&\quad + \rho n\|y^{k-1} - y^{\star k}\|^2 - \rho n\|y^k - y^{k-1}\|^2 - 2\alpha\sum_{=1}^n \|y_i^k - y^{\star k}\|^2. \qquad (55)
\end{aligned}$$

Using the definition of $Q_i(\cdot,\cdot)$, we write (55) as

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^k - \lambda_i^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 \\
&\leq \frac{2}{\rho}\sum_{i=1}^n \|\lambda_i^{k-1} - \lambda_i^{\star k-1}\|^2 + \frac{2}{\rho}\sum_{i=1}^n \|\nabla Q_i(x^{k-1}, y_i^{\star k-1}) - \nabla Q_i(x^k, y_i^{\star k})\|^2 \\
&\quad + \rho n\|y^{k-1} - y^{\star k}\|^2 - \rho n\|y^k - y^{k-1}\|^2 - 2\alpha\sum_{=1}^n \|y_i^k - y^{\star k}\|^2. \qquad (56)
\end{aligned}$$

Using the lipschitz continuity of ∇Q, we obtain

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^k - \lambda_i^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 \\
&\leq \frac{2}{\rho}\sum_{i=1}^n \|\lambda_i^{k-1} - \lambda_i^{\star k-1}\|^2 + \frac{2L_q^2}{\rho}\sum_{i=1}^n \|y_i^{\star k-1} - y_i^{\star k}\|^2 + \rho n\|y^{k-1} - y^{\star k}\|^2 - \rho n\|y^k - y^{k-1}\|^2 - 2\alpha\sum_{=1}^n \|y_i^k - y^{\star k}\|^2. \qquad (57)
\end{aligned}$$

Equivalently, we can write (57) as

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^k - \lambda_i^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 \\
&\leq \frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^{k-1} - \lambda_i^{\star k-1}\|^2 + \frac{1}{\rho}\sum_{i=1}^n \|g_i^{k-1} - s^{k-1} - (H_i^{k-1} + \alpha^{k-1}I)y_i^{k-1} - g_i^{k-1} + (H_i^{k-1} + \alpha^{k-1}I)y_i^{\star k-1}\|^2 \\
&\quad + \frac{2L_q^2}{\rho}\sum_{i=1}^n \|y_i^{\star k-1} - y_i^{\star k}\|^2 + \rho n\|y^{k-1} - y^{\star k}\|^2 - \rho n\|y^k - y^{k-1}\|^2 - 2\alpha\sum_{=1}^n \|y_i^k - y^{\star k}\|^2. \qquad (58)
\end{aligned}$$

Using the inequality $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ and the expression of ∇Q, we obtain

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^k - \lambda_i^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 \\
&\leq \frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^{k-1} - \lambda_i^{\star k-1}\|^2 + \frac{2}{\rho}\sum_{i=1}^n \|\nabla Q_i(x^{k-1}, y_i^{k-1}) - \nabla Q_i(x^{k-1}, y_i^{\star k-1})\|^2 + \frac{2n}{\rho}\|s^{k-1}\|^2 + \frac{2L_q^2}{\rho}\sum_{i=1}^n \|y_i^{\star k-1} - y_i^{\star k}\|^2 \\
&\quad + \rho n\|y^{k-1} - y^{\star k}\|^2 - \rho n\|y^k - y^{k-1}\|^2 - 2\alpha\sum_{=1}^n \|y_i^k - y^{\star k}\|^2 \\
&\leq \frac{1}{\rho}\sum_{i=1}^n \|\lambda_i^{k-1} - \lambda_i^{\star k-1}\|^2 + \frac{2L_q^2}{\rho}\sum_{i=1}^n \|y_i^{k-1} - y^{\star k-1}\|^2 + \frac{2n}{\rho}\|s^{k-1}\|^2 + \frac{2L_q^2}{\rho}\sum_{i=1}^n \|y^{\star k-1} - y^{\star k}\|^2 \\
&\quad + \rho n\|y^{k-1} - y^{\star k}\|^2 - \rho n\|y^k - y^{k-1}\|^2 - 2\alpha\sum_{=1}^n \|y_i^k - y^{\star k}\|^2. \qquad (59)
\end{aligned}$$

Replacing s^{k-1} by its definition, we get

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k-\lambda_i^{\star k}\|^2+\rho n\|y^k-y^{\star k}\|^2\\
&\le \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1}-\lambda_i^{\star k-1}\|^2+\frac{2L_q^2}{\rho}\sum_{i=1}^{n}\|y_i^{k-1}-y^{\star k-1}\|^2+2n\rho\|y^{k-1}-y^{k-2}\|^2+\frac{2L_q^2 n}{\rho}\|y^{\star k-1}-y^{\star k}\|^2\\
&\quad+\rho n\|y^{k-1}-y^{\star k}\|^2-\rho n\|y^k-y^{k-1}\|^2-2\alpha\sum_{i=1}^{n}\|y_i^k-y^{\star k}\|^2. \qquad (60)
\end{aligned}$$

Let $\alpha=\alpha_1+\alpha_2$. Adding and subtracting y^k in $\|y^{k-1}-y^{\star k}\|^2$ and using $\|a+b\|^2\le 2\|a\|^2+2\|b\|^2$, we obtain

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k-\lambda_i^{\star k}\|^2+\rho n\|y^k-y^{\star k}\|^2\\
&\le \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1}-\lambda_i^{\star k-1}\|^2+\frac{2L_q^2}{\rho}\sum_{i=1}^{n}\|y_i^{k-1}-y^{\star k-1}\|^2+2\rho n\|y^{k-1}-y^{k-2}\|^2+\frac{2L_q^2 n}{\rho}\|y^{\star k-1}-y^{\star k}\|^2\\
&\quad+2\rho n\|y^k-y^{k-1}\|^2+2\rho n\|y^k-y^{\star k}\|^2-\rho n\|y^k-y^{k-1}\|^2-2\alpha_1\sum_{i=1}^{n}\|y_i^k-y^{\star k}\|^2-2\alpha_2\sum_{i=1}^{n}\|y_i^k-y^{\star k}\|^2. \qquad (61)
\end{aligned}$$

Using $\sum_{i=1}^{n}\|y_i^k-y^{\star k}\|^2\ge\|\sum_{i=1}^{n}(y_i^k-y^{\star k})\|^2=n\|y^k-y^{\star k}\|^2$, we can rewrite (61) as

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k-\lambda_i^{\star k}\|^2+\rho n\|y^k-y^{\star k}\|^2\\
&\le \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1}-\lambda_i^{\star k-1}\|^2+\frac{2L_q^2}{\rho}\sum_{i=1}^{n}\|y_i^{k-1}-y^{\star k-1}\|^2+2\rho n\|y^{k-1}-y^{k-2}\|^2+\frac{2L_q^2 n}{\rho}\|y^{\star k-1}-y^{\star k}\|^2\\
&\quad+2\rho n\|y^k-y^{k-1}\|^2+2\rho n\|y^k-y^{\star k}\|^2-\rho n\|y^k-y^{k-1}\|^2-2\alpha_1 n\|y^k-y^{\star k}\|^2-2\alpha_2\sum_{i=1}^{n}\|y_i^k-y^{\star k}\|^2. \qquad (62)
\end{aligned}$$

Assuming $\|y^{k-1}-y^k\|^2\le\|y^k-y^{\star k}\|^2$ which holds for sufficiently large ρ, and choosing $\alpha_1\ge 2.5\rho$, we get

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k-\lambda_i^{\star k}\|^2+\rho n\|y^k-y^{\star k}\|^2\\
&\le \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1}-\lambda_i^{\star k-1}\|^2+\frac{2L_q^2}{\rho}\sum_{i=1}^{n}\|y_i^{k-1}-y^{\star k-1}\|^2+2\rho n\|y^{k-1}-y^{k-2}\|^2+\frac{2L_q^2 n}{\rho}\|y^{\star k-1}-y^{\star k}\|^2\\
&\quad-2\rho n\|y^k-y^{k-1}\|^2-2\alpha_2\sum_{i=1}^{n}\|y^k-y^{\star k}\|^2. \qquad (63)
\end{aligned}$$

Re-arranging the terms, we get

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k-\lambda_i^{\star k}\|^2+\rho n\|y^k-y^{\star k}\|^2+2\rho n\|y^k-y^{k-1}\|^2\\
&\le \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1}-\lambda_i^{\star k-1}\|^2+\frac{2L_q^2}{\rho}\sum_{i=1}^{n}\|y_i^{k-1}-y^{\star k-1}\|^2+2\rho n\|y^{k-1}-y^{k-2}\|^2\\
&\quad+\frac{2L_q^2 n}{\rho}\|y^{\star k-1}-y^{\star k}\|^2-2\alpha_2\sum_{i=1}^{n}\|y_i^k-y^{\star k}\|^2. \qquad (64)
\end{aligned}$$

Adding and subtracting y^{k-1} in $\|y^{\star k-1} - y^{\star k}\|^2$, we get

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k - \lambda_i^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 + 2\rho n\|y^k - y^{k-1}\|^2 \\
&\leq \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1} - \lambda_i^{\star k-1}\|^2 + \frac{2L_q^2}{\rho}\sum_{i=1}^{n}\|y_i^{k-1} - y^{\star k-1}\|^2 + 2\rho n\|y^{k-1} - y^{k-2}\|^2 + \frac{4L_q^2 n}{\rho}\|y^{\star k-1} - y^{k-1}\|^2 \\
&\quad + \frac{4L_q^2 n}{\rho}\|y^{k-1} - y^{\star k}\|^2 - 2\alpha_2\sum_{i=1}^{n}\|y_i^k - y^{\star k}\|^2. \qquad (65)
\end{aligned}$$

Next, let $\alpha_2 = \alpha_3 + \alpha_4$. Adding and subtracting y^k in $\|y^{k-1} - y^{\star k}\|^2$, we get

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k - \lambda_i^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 + 2\rho n\|y^k - y^{k-1}\|^2 \\
&\leq \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1} - \lambda_i^{\star k-1}\|^2 + \frac{2L_q^2}{\rho}\sum_{i=1}^{n}\|y_i^{k-1} - y^{\star k-1}\|^2 + 2\rho n\|y^{k-1} - y^{k-2}\|^2 + \frac{4L_q^2 n}{\rho}\|y^{\star k-1} - y^{k-1}\|^2 \\
&\quad + \frac{8L_q^2 n}{\rho}\|y^{k-1} - y^k\|^2 + \frac{8L_q^2 n}{\rho}\|y^k - y^{\star k}\|^2 - 2\alpha_3 n\|y^k - y^{\star k}\|^2 - 2\alpha_4\sum_{i=1}^{n}\|y_i^k - y^{\star k}\|^2. \qquad (66)
\end{aligned}$$

Choosing $\alpha_3 \geq \frac{8L_q^2 n}{\rho}$, and using the assumption $\|y^{k-1} - y^k\|^2 \leq \|y^k - y^{\star k}\|^2$, we get

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k - \lambda_i^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 + 2\rho n\|y^k - y^{k-1}\|^2 \\
&\leq \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1} - \lambda_i^{\star k-1}\|^2 + \frac{2L_q^2}{\rho}\sum_{i=1}^{n}\|y_i^{k-1} - y^{\star k-1}\|^2 + 2\rho n\|y^{k-1} - y^{k-2}\|^2 + \frac{4L_q^2 n}{\rho}\|y^{\star k-1} - y^{k-1}\|^2 \\
&\quad - 2\alpha_4\sum_{i=1}^{n}\|y_i^k - y^{\star k}\|^2. \qquad (67)
\end{aligned}$$

Let $\alpha_4 = \alpha_5 + \alpha_6$, we write (67) as

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k - \lambda_i^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 + 2\rho n\|y^k - y^{k-1}\|^2 \\
&\leq \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1} - \lambda_i^{\star k-1}\|^2 + \frac{2L_q^2}{\rho}\sum_{i=1}^{n}\|y_i^{k-1} - y^{\star k-1}\|^2 + 2\rho n\|y^{k-1} - y^{k-2}\|^2 + \frac{4L_q^2 n}{\rho}\|y^{k-1} - y^{\star k-1}\|^2 \\
&\quad - 2\alpha_5\sum_{i=1}^{n}\|y_i^k - y^{\star k}\|^2 - 2\alpha_6\sum_{i=1}^{n}\|y_i^k - y^{\star k}\|^2. \qquad (68)
\end{aligned}$$

Re-arrange terms, we obtain

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k - \lambda_i^{\star k}\|^2 + 2\alpha_5\sum_{i=1}^{n}\|y_i^k - y^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 + 2\rho n\|y^k - y^{k-1}\|^2 \\
&\leq \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1} - \lambda_i^{\star k-1}\|^2 + \frac{2L_q^2}{\rho}\sum_{i=1}^{n}\|y_i^{k-1} - y^{\star k-1}\|^2 + \frac{4L_q^2 n}{\rho}\|y^{k-1} - y^{\star k-1}\|^2 + 2\rho n\|y^{k-1} - y^{k-2}\|^2 \\
&\quad - 2\alpha_6\sum_{i=1}^{n}\|y_i^k - y^{\star k}\|^2. \qquad (69)
\end{aligned}$$

Let $\beta_1 = \alpha_5$, and $\beta_2 = \alpha_6$, Hence, we can write (69) as

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k - \lambda_i^{\star k}\|^2 + 2\beta_1\sum_{i=1}^{n}\|y_i^k - y^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 + 2\rho n\|y^k - y^{k-1}\|^2 \\
&\leq \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1} - \lambda_i^{\star k-1}\|^2 + \frac{2L_q^2}{\rho}\sum_{i=1}^{n}\|y_i^{k-1} - y^{\star k-1}\|^2 + \frac{4L_q^2 n}{\rho}\|y^{k-1} - y^{\star k-1}\|^2 + 2\rho n\|y^{k-1} - y^{k-2}\|^2 \\
&\quad - 2\beta_2\sum_{i=1}^{n}\|y_i^k - y^{\star k}\|^2. \qquad (70)
\end{aligned}$$

Note that,

$$\begin{aligned}
\alpha =&\alpha_1 + \alpha_2 \\
=&\alpha_1 + \alpha_3 + \beta_1 + \beta_2 \\
\geq&2.5\rho + \frac{8L_q^2 n}{\rho} + \beta_1 + \beta_2 \qquad (71)
\end{aligned}$$

Hence, $\beta_1 + \beta_2 \leq \alpha - 2.5\rho - \frac{8L_q^2 n}{\rho}$. That concludes the proof.

C. Proof of Theorem 1

From the statement of Lemma 2, we can write

$$\begin{aligned}
&\frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^k - \lambda_i^{\star k}\|^2 + 2\beta_1\sum_{i=1}^{n}\|y_i^k - y^{\star k}\|^2 + \rho n\|y^k - y^{\star k}\|^2 + 2\rho n\|y^k - y^{k-1}\|^2 \\
&\leq \frac{1}{\rho}\sum_{i=1}^{n}\|\lambda_i^{k-1} - \lambda_i^{\star k-1}\|^2 + \frac{2L_q^2}{\rho}\sum_{i=1}^{n}\|y_i^{k-1} - y^{\star k-1}\|^2 + \frac{4L_q^2 n}{\rho}\|y^{k-1} - y^{\star k-1}\|^2 + 2\rho n\|y^{k-1} - y^{k-2}\|^2 \\
&\quad - 2\beta_2\sum_{i=1}^{n}\|y_i^k - y^{\star k}\|^2. \qquad (72)
\end{aligned}$$

Choosing α that satisfies (23) for $\beta_1 \geq \frac{L_q^2}{\rho}$, $\rho \geq 2L_q$, and $\beta_2 > 0$ and from the definition of Lyapunov function V_k (24), we can write

$$V^k \leq V^{k-1} - \left(2\beta_2\sum_{i=1}^{n}\|y_i^k - y^{\star k}\|^2\right). \qquad (73)$$

Therefore, V^k decreases in each iteration k. Consequently, since $V_k \geq 0$, this implies that every term in V^k goes to 0, i.e. $\lambda_i^k \rightarrow \lambda_i^{\star k}$, $y_i^k \rightarrow y^{\star k}$, and $y^k \rightarrow y^{\star k}$, which completes the proof.