Towards Quantized Model Parallelism for Graph-Augmented MLPs Based on Gradient-Free ADMM Framework

Junxiang Wang[†], Hongyi Li[‡], Zheng Chai[§], Yongchao Wang[‡], Yue Cheng[§] and Liang Zhao[†]
[†]Department of Computer Science and Informatics, Emory University, Atlanta, Georgia, USA, 30030
[‡]The State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, Shaanxi, China, 710071
[§] Department of Computer Science, University of Virginia, Charlottesville, Virginia, USA, 22904

***Abstract*— While Graph Neural Networks (GNNs) are popular in the deep learning community, they suffer from several challenges including over-smoothing, over-squashing, and gradient vanishing. Recently, a series of models have attempted to relieve these issues by first augmenting the node features and then imposing node-wise functions based on Multi-Layer Perceptron (MLP), which are widely referred to as GA-MLP models. However, while GA-MLP models enjoy deeper architectures for better accuracy, their efficiency largely deteriorates. Moreover, popular acceleration techniques such as stochastic-version or data-parallelism cannot be effectively applied due to the dependency among samples (i.e., nodes) in graphs. To address these issues, in this paper, instead of data parallelism, we propose a parallel graph deep learning Alternating Direction Method of Multipliers (pdADMM-G) framework to achieve model parallelism: parameters in each layer of GA-MLP models can be updated in parallel. The extended pdADMM-G-Q algorithm reduces communication costs by introducing the quantization technique. Theoretical convergence to a (quantized) stationary point of the pdADMM-G algorithm and the pdADMM-G-Q algorithm is provided with a sublinear convergence rate $o(1/k)$, where k is the number of iterations. Extensive experiments demonstrate the convergence of two proposed algorithms. Moreover, they lead to a more massive speedup and better performance than all state-of-the-art comparison methods on nine benchmark datasets. Last but not least, the proposed pdADMM-G-Q algorithm reduces communication overheads by up to 45% without loss of performance. Our code is available at https://github.com/xianggebenben/pdADMM-G.**

***Index Terms*—Model Parallelism, Graph Neural Networks, Alternating Direction Method of Multipliers, Convergence, Quantization**

I. Introduction

Graph Neural Networks (GNNs) have accomplished state-of-the-art performance in various graph applications such as node classification and link prediction. This is because they handle graph-structured data via aggregating neighbor information and extending operations and definitions of the deep learning approach [1]. However, their performance has significantly been restricted via their depths due to the over-smoothing problem (i.e. the representations of different nodes in a graph tend to be similar when stacking multiple layers) [2], the over-squashing problem (i.e. the information flow among distant nodes distorts along the long-distance interactions) [3], and the gradient vanishing problem (i.e. the signals of gradients decay with the depths of GNN models) [2]. These challenges still exist even though some models such as GraphSAGE [4] have been proposed to alleviate them.

On the other hand, the Graph Augmented Multi-Layer Perceptron (GA-MLP) models have recently received fast-increasing attention as an alternative to deal with the aforementioned drawbacks of conventional GNNs via the augmentation of graph features. GA-MLP models augment node representations of graphs and feed them into Multi-Layer Perceptron (MLP) models. Compared with GNNs, GA-MLP models are more resistant to the over-smoothing problem [2] and therefore demonstrate outstanding performance. For example, Wu et al. showed that a two-layer GA-MLP approximates the performance of the GNN models on multiple datasets [5].

GA-MLP models are supposed to perform better with the increase of their depths. However, similar to GNNs, GA-MLP models still suffer from the gradient vanishing problem, which is caused by the mechanism of the classic backpropagation algorithm. This is because gradient signals diminish during the transmission among deep layers. Moreover, while the models go deeper, efficiency will become an issue, especially for medium- and large-size graphs. Compared to the data such as images and texts, where identically and independently distributed (i.i.d.) samples are assumed, efficiency issues in graph data are much more difficult to handle due to the dependency among data samples (i.e., nodes in graphs). Such dependency seriously troubles the effectiveness of using typical acceleration techniques such as sampling-based methods, and data-parallelism distributed learning in solving the efficiency issue. Therefore, parallelizing the computation along layers is a natural workaround, but the backpropagation prevents the gradients of different layers from being calculated in parallel. This is because the calculation of the gradient in one layer is dependent on its previous layers.

To handle these challenges, recently gradient-free optimization methods such as the Alternating Direction Method of Multipliers (ADMM) have been investigated to overcome the difficulties of the backpropagation algorithm. The spirit of ADMM is to decouple a neural network into layerwise subproblems such that each of them can be solved efficiently. ADMM does not require gradient calculation and therefore can avoid the gradient vanishing problem. Existing literature has shown its great potential. For example, Talyor et al. and

* Junxiang Wang and Hongyi Li contribute equally to this work, and Yongchao Wang and Liang Zhao are corresponding authors.

Wang et al. proposed ADMM to train MLP models [6], [7]. Extensive experiments have demonstrated that the ADMM has outperformed most comparison methods such as Gradient Descent (GD).

In this paper, we propose a novel parallel graph deep learning Alternating Direction Method of Multipliers (pdADMM-G) optimization framework to train large-scale GA-MLP models, and the extended pdADMM-G-Q algorithm reduces the communication cost of the pdADMM-G algorithm by the quantization techniques. Our contributions to this paper include:

- We propose a novel reformulation of GA-MLP models, which splits a neural network into independent layer partitions and allow for ADMM to achieve model parallelism.
- We propose a novel pdADMM-G framework to train a GA-MLP model. All subproblems generated by the ADMM algorithm are discussed. The extended pdADMM-G-Q algorithm reduces communication costs by introducing the quantization technique.
- We provide the theoretical convergence guarantee of the proposed pdADMM-G algorithm and the pdADMM-G-Q algorithm. Specifically, they converge to a (quantized) stationary point of GA-MLP models when the hyperparameters are sufficiently large, and their sublinear convergence rates are $o(1/k)$.
- We conduct extensive experiments on nine benchmark datasets to show the convergence, the massive speedup of the proposed pdADMM-G algorithm and the pdADMM-G-Q algorithm, as well as their outstanding performance when compared with all state-of-the-art optimizers. Moreover, the proposed pdADMM-G-Q algorithm reduces communication overheads by up to 45%.

The organization of this paper is shown as follows: In Section II, we summarize recent related research work to this paper. In Section III, we propose the pdADMM-G algorithm and the pdADMM-G-Q algorithm to train deep GA-MLP models. Section IV details the convergence properties of the proposed pdADMM-G algorithm and the pdADMM-G-Q algorithm. Extensive experiments on nine benchmark datasets to demonstrate the convergence, speedup, communication savings, and outstanding performance of the pdADMM-G algorithm and the pdADMM-G-Q algorithm are shown in Section V, and Section VI concludes this work.

II. Related Work

This section summarizes existing literature related to this research.

Distributed Deep Learning. With the increase of public datasets and layers of neural networks, it is imperative to establish distributed deep learning systems for large-scale applications. Many systems have been established to satisfy such needs. Famous systems include Terngrad [8], Horovod [9], SINGA [10] Mxnet [11], TicTac [12] and Poseidon [13]. They applied some parallelism techniques to reduce computational time, and therefore improved the speedup. Existing parallelism techniques can be classified into two categories: data parallelism and model parallelism. Data parallelism focuses on distributing data across different processors and then aggregating results from them into a server. Scaling GD is one of the most common ways to reach data parallelism [14]. For example, the distributed architecture, Poseidon, is achieved by scaling GD through overlapping communication and computation over networks. The recently proposed ADMM [6], [7] is another way to achieve data parallelism. However, data parallelism suffers from the bottleneck of a neural network: for GD, the gradient should be transmitted through all processors; for ADMM, the parameters in one layer are subject to those in its previous layer. As a result, this leads to heavy communication costs and time delays. Model parallelism, however, can solve this challenge because model parallelism splits a neural network into many independent partitions. In this way, each partition can be optimized independently and reduce layer dependency. For instance, Parpas and Muir proposed a parallel-in-time method from the perspective of dynamic systems [15]; Huo et al. introduced a feature replay algorithm to achieve model parallelism [16]. Zhuang et al. broke layer dependency by introducing the delayed gradient [17].

Deep Learning on Graphs. Graphs are ubiquitous structures and are popular in real-world applications. There is a surge of interest to apply deep learning techniques to graphs. For a comprehensive summary please refer to [18]. It classified existing GNN models into four categories: Recurrent Graph Neural Networks (RecGNNs), Convolutional Graph Neural Networks (ConvGNNs), Graph Autoencoders (GAEs), and Spatial-Temporal Graph Neural Networks (STGNNs). RecGNNs learn node representation with recurrent neural networks via the message passing mechanisms [19], [20], [21]; ConvGNNs generalize the operations of convolution to graph data and stack multiple convolution layers to extract high-level node features [22], [23], [24]; GAEs encode node information into a latent space and reconstruct graphs from the encoded node representation [25], [26], [27]; the idea of STGNNs is to capture spatial dependency and temporal dependency simultaneously [28], [29], [30].

III. The pdADMM-G Algorithm

We propose the pdADMM-G algorithm to solve GA-MLP models in this section. Specifically, Section III-A formulates the GA-MLP model training problem , and Section III-B proposes the pdADMM-G algorithm. Section III-C extends the proposed pdADMM-G algorithm to the pdADMM-G-Q algorithm for quantization.

A. Problem Formulation

Consider a graph $G = (V, E)$, where V and E are sets of nodes and edges, respectively, $|V|$ is the number of nodes, let $\Psi = \{\psi_1(A), \cdots, \psi_K(A)\}$ be a set of (usually multi-hop) operators $\psi_i(A) : \mathbb{R}^{|V|} \to \mathbb{R}^{|V|} (i = 1, \cdots, K)$ that are functions of the adjacency matrix $A \in \{0, 1\}^{|V| \times |V|}$, and $\mathbb{R}^{|V|}$ is the domain of $\psi_i(A)$ $(i = 1, \cdots, K)$. $X_k = H\psi_k(A)$ is the augmentation of node features by the k-hop operator, where $H \in \mathbb{R}^{d \times |V|}$ is a matrix of node features, and d is the

Notations	Descriptions
L	Number of layers.
W_l	The weight matrix for the l-th layer.
b_l	The intercept vector for the l-th layer.
z_l	The auxiliary variable of the linear mapping for the l-th layer.
$f_l(z_l)$	The nonlinear activation function for the l-th layer.
p_l	The input for the l-th layer.
q_l	The output for the l-th layer.
X	The node representation of the graph.
A	The adjacency matrix of the graph.
y	The predefined label vector.
$R(z_L, y)$	The risk function for the L-th layer.
n_l	The number of neurons for the l-th layer.
u_l	The dual variable for the l-th layer.

TABLE I: Important Notations

dimension of features. $X_k (k = 1, \cdots, K)$ are stacked into $X = [X_1; \cdots; X_K]$ by column. Then the GA-MLP training problem is formulated as follows [7]:

Problem 1.

$$\min_{W_l, b_l, z_l, p_l} R(z_L; y),$$
$$s.t.\ z_l = W_l p_l + b_l,\ p_{l+1} = f_l(z_l) (l = 1, \cdots, L-1),$$

where $p_1 = X \in \mathbb{R}^{n_0 \times |V|}$ is the input of deep GA-MLP models, where $n_0 = Kd$ is the dimension of input and y is a predefined label vector. $p_l \in \mathbb{R}^{n_l \times |V|}$ is the input for the l-th layer, also the output for the $(l-1)$-th layer, and n_l is the number of neurons for the l-th layer. $R(z_L; y)$ is a risk function for the L-th layer, which is convex and continuous; $z_l = W_l p_l + b_l$ and $p_{l+1} = f_l(z_l)$ are linear and nonlinear mappings for the l-th layer, respectively, and $W_l \in \mathbb{R}^{n_l \times n_{l-1}}$ and $b_l \in \mathbb{R}^{n_l}$ are the weight matrix and the intercept vector for the l-th layer, respectively.

In Problem 1, Ψ can be considered as a prepossessing step to augment node features via A, and hence it is predefined. One common choice can be $\Psi = \{I, A, A^2, \cdots, A^{K-1}\}$.

Problem 1 can be addressed by deep learning Alternating Direction Method of Multipliers (dlADMM) [7]. However, parameters in one layer are dependent on its neighboring layers and hence can not achieve parallelism. For example, the update of p_{l+1} on the $(l+1)$-th layer needs to wait before z_l on the l-th layer is updated. In order to address layer dependency, we relax Problem 1 to Problem 2 as follows:

Problem 2.

$$\min_{\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{q}} F(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{q}) = R(z_L; y)$$
$$+ (\nu/2)(\sum_{l=1}^{L} \|z_l - W_l p_l - b_l\|_2^2 + \sum_{l=1}^{L-1} \|q_l - f_l(z_l)\|_2^2),$$
$$s.t.\ p_{l+1} = q_l,$$

where $\mathbf{p} = \{p_l\}_{l=1}^{L}$, $\mathbf{W} = \{W_l\}_{l=1}^{L}$, $\mathbf{b} = \{b_l\}_{l=1}^{L}$, $\mathbf{z} = \{z_l\}_{l=1}^{L}$, $\mathbf{q} = \{q_l\}_{l=1}^{L-1}$, and $\nu > 0$ is a tuning parameter. As $\nu \to \infty$, Problem 2 approaches Problem 1. We reduce layer dependency by splitting the output of the l-th layer and the input of the $(l+1)$-th layer into two variables p_{l+1} and q_l, respectively.

B. The pdADMM-G Algorithm

The high-level overview of the pdADMM-G algorithm is shown in Figure 1. Specifically, the inputs of GA-MLP models

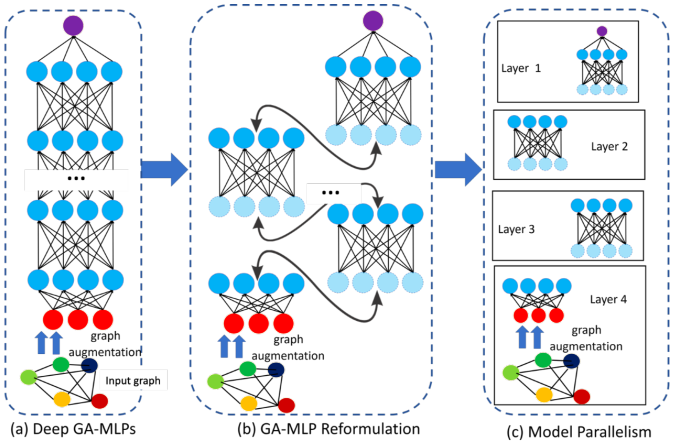


Fig. 1: The overall pdADMM-G optimization algorithm: it splits GA-MLP models into layerwise components.

are augmented by $H\psi_k(A)$ $(k = 1, \cdots, K)$, and then GA-MLP models are split into multiple layers, each of which can be optimized by an independent client. Therefore, layerwise training can be implemented in parallel. Moreover, the gradient vanishing problem can be avoided in this way. This is because the accumulated gradient calculated by the backpropagation algorithm is split into layerwise components.

Now we follow the ADMM routine to solve Problem 2. The augmented Lagrangian function is formulated mathematically as follows:

$$L_\rho(\mathbf{p}, \mathbf{W}, \mathbf{b}, \mathbf{z}, \mathbf{q}, \mathbf{u})$$
$$= F(\mathbf{p}, \mathbf{W}, \mathbf{b}, \mathbf{z}, \mathbf{q}) + \sum_{l=1}^{L-1} (u_l^T (p_{l+1} - q_l) + (\rho/2)\|p_{l+1} - q_l\|_2^2)$$
$$= R(z_L; y) + \phi(p_1, W_1, b_1, z_1) + \sum_{l=2}^{L} \phi(p_l, W_l, b_l, z_l, q_{l-1}, u_{l-1})$$
$$+ (\nu/2) \sum_{l=1}^{L-1} \|q_l - f_l(z_l)\|_2^2,$$

where $\phi(p_1, W_1, b_1, z_1) = (\nu/2)\|z_1 - W_1 p_1 - b_1\|_2^2$, $\phi(p_l, W_l, b_l, z_l, q_{l-1}, u_{l-1}) = (\nu/2)\|z_l - W_l p_l - b_l\|_2^2 + u_{l-1}^T (p_l - q_{l-1}) + (\rho/2)\|p_l - q_{l-1}\|_2^2$, $u_l (l = 1, \cdots, L-1)$ are dual variables, $\rho > 0$ is a parameter, and $\mathbf{u} = \{u_l\}_{l=1}^{L-1}$. The detail of the pdADMM-G algorithm is shown in Algorithm 1. Specifically, Lines 5-9 update primal variables $\mathbf{p}$, $\mathbf{W}$, $\mathbf{b}$, $\mathbf{z}$ and $\mathbf{q}$, respectively, while Line 11 updates the dual variable $\mathbf{u}$. Due to space limit, the details of all subproblems are shown in Section A in the Appendix.

Our proposed pdADMM-G algorithm can be efficient for training deep GA-MLP models via the greedy layerwise training strategy [31]. Specifically, we begin by training a shallow GA-MLP model. Next, more layers are increased to the GA-MLP model and their parameters are trained, then we introduce even more layers and iterate this process until the whole deep GA-MLP model is included. The pdADMM-G algorithm can achieve excellent performance as well as reduce training costs by this strategy.

Last but not least, we compare the computational costs of the proposed pdADMM-G algorithm with the state-of-the-art backpropagation algorithm, on which the gradient descent is based. We show that they share the same level of compu-

Algorithm 1 The pdADMM-G Algorithm to Solve Problem 2

Require: y, $p_1 = X$, ρ, ν.
Ensure: $\mathbf{p}, \mathbf{W}, \mathbf{b}, \mathbf{z}, \mathbf{q}$.
Initialize $k = 0$.
while $\mathbf{p}^k, \mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^k$ not converged **do**
$p_l^{k+1} \leftarrow \arg\min_{p_l} L_\rho(\mathbf{p}, \mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^k, \mathbf{u}^k)$ for different l in parallel.
$W_l^{k+1} \leftarrow \arg\min_{W_l} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^k, \mathbf{u}^k)$ for different l in parallel.
$b_l^{k+1} \leftarrow \arg\min_{b_l} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}, \mathbf{z}^k, \mathbf{q}^k, \mathbf{u}^k)$ for different l in parallel.
$z_l^{k+1} \leftarrow \arg\min_{z_l} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}, \mathbf{q}^k, \mathbf{u}^k)$ for different l in parallel.
$q_l^{k+1} \leftarrow \arg\min_{q_l} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}, \mathbf{u}^k)$ for different l in parallel.
$r_l^k \leftarrow p_{l+1}^{k+1} - q_l^{k+1} (l = 1, \cdots, L)$ in parallel # Compute residuals.
$u_l^{k+1} \leftarrow u_l^k + \rho(p_{l+1}^{k+1} - q_l^{k+1})$ for different l in parallel.
$k \leftarrow k + 1$.
end while
Output $\mathbf{p}, \mathbf{W}, \mathbf{b}, \mathbf{z}, \mathbf{q}$.

tational costs. For the backpropagation algorithm, the most costly operation is the matrix multiplication $z_l = W_l p_l + b_l$ in the forward pass, where $W_l \in \mathbb{R}^{n_l \times n_{l-1}}$ and $p_l \in \mathbb{R}^{n_{l-1} \times |V|}$, which requires a time complexity of $O(n_l n_{l-1}|V|)$ [32]; for the proposed pdADMM-G algorithm, the most costly operation is to compute the derivative $\nabla_{W_l}\phi$, and it also involves the matrix multiplication, and hence its time complexity is again $O(n_l n_{l-1}|V|)$. However, the proposed pdADMM-G algorithm trains the whole GA-MLP model in a model parallelism fashion [33], and therefore all computational costs can be split into different independent clients for parallel training; whereas the backpropagation algorithm is implemented sequentially, and thus it is less efficient than the proposed pdADMM-G algorithm.

C. Quantization Extension of pdADMM-G (pdADMM-G-Q)

In the proposed pdADMM-G algorithm, p_l and q_l are transmitted back and forth among layers (i.e. clients). However, the communication overheads of p_l and q_l surge for a large-scale graph G with millions of nodes. To alleviate this challenge, the quantization technique is commonly utilized to reduce communication costs by mapping continuous values into a discrete set [34]. In other words, p_l is required to fit into a countable set Δ, which is shown as follows:

Problem 3.

$$\min_{\boldsymbol{p},\boldsymbol{W},\boldsymbol{b},\boldsymbol{z},\boldsymbol{q}} F(\boldsymbol{p},\boldsymbol{W},\boldsymbol{b},\boldsymbol{z},\boldsymbol{q}) = R(z_L; y) + (\nu/2)\Big(\sum_{l=1}^{L} \|z_l - W_l p_l - b_l\|_2^2 + \sum_{l=1}^{L-1} \|q_l - f_l(z_l)\|_2^2\Big), \quad s.t.\ p_{l+1} = q_l,\ p_l \in \Delta = \{\delta_1, \cdots, \delta_m\},$$

where $\delta_i (i = 1, \cdots, m) \in \Delta$ are quantized values, which can be integers or low-precision values. $m = |\Delta|$ is the cardinality of Δ. To address Problem 3, we rewrite it into the following form:

$$\min_{\mathbf{p},\mathbf{W},\mathbf{b},\mathbf{z},\mathbf{q}} R(z_L; y) + \sum_{l=2}^{L} \mathbb{I}(p_l) + (\nu/2)\Big(\sum_{l=1}^{L} \|z_l - W_l p_l - b_l\|_2^2 + \sum_{l=1}^{L-1} \|q_l - f_l(z_l)\|_2^2\Big), \quad s.t.\ p_{l+1} = q_l,$$

where the indicator function $\mathbb{I}(p_l)$ is defined as follows: $\mathbb{I}(p_l) = 0$ if $p_l \in \Delta$, and $\mathbb{I}(p_l) = +\infty$ if $p_l \notin \Delta$. The augmented Lagrangian of Problem 3 is shown as follows:

$$\beta_\rho(\mathbf{p}, \mathbf{W}, \mathbf{b}, \mathbf{z}, \mathbf{q}, \mathbf{u}) = L_\rho(\mathbf{p}, \mathbf{W}, \mathbf{b}, \mathbf{z}, \mathbf{q}, \mathbf{u}) + \sum_{l=2}^{L} \mathbb{I}(p_l),$$

where L_ρ is the augmented Lagrangian of Problem 2. The extended pdADMM-G-Q algorithm follows the same routine as the pdADMM-G algorithm, where L_ρ is replaced with β_ρ. Due to space limit, the solutions to all subproblems generated by two proposed algorithms are shown in Section B in the Appendix.

IV. Convergence Analysis

In this section, the theoretical convergence of the proposed pdADMM-G algorithm and the pdADMM-G-Q algorithm is provided. Due to space limit, we only provide sketches of proofs in this section, and their details are available in Section C in the Appendix. Our problem formulations are more difficult than existing ADMM literature: the term $\|q_l - f_l(z_l)\|_2^2$ is coupled in the objective, while it is separable in the existing ADMM formulations. To address this, we impose a mild condition that $\partial f_l(z_l)$ is bounded in Assumption 1, and prove that u_l is controlled via q_l and z_l in Lemma 5 in Section C in the Appendix.

Firstly, the proper function, Lipschitz continuity, and coercivity are defined as follows:

Definition 1 (Proper Functions)**.** *[35]. For a convex function $g(x) : \mathbb{R} \to \mathbb{R} \bigcup \{\pm\infty\}$, g is called proper if $\forall x \in \mathbb{R}$, $g(x) > -\infty$, and $\exists x_0 \in \mathbb{R}$ such that $g(x_0) < +\infty$.*

Definition 2. *(Lipschitz Continuity) A function $g(x)$ is Lipschitz continuous if there exists a constant $D > 0$ such that $\forall x_1, x_2$, the following holds*

$$\|g(x_1) - g(x_2)\| \leq D\|x_1 - x_2\|.$$

Definition 3. *(Coercivity) A function $h(x)$ is coerce over the feasible set $\mathscr{F}$ means that $h(x) \to \infty$ if $x \in \mathscr{F}$ and $\|x\| \to \infty$.*

Next, the definition of a quantized stationary point [34] is shown as follows:

Definition 4. *(Quantized Stationary Point) The p_l is a quantized stationary point of of Problem 3 if there exists $\tau > 0$ such that*

$$p_l \in \arg\min_{\delta \in \Delta} \|\delta - (p_l - \nabla_{p_l} F(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{q})/\tau)\|.$$

The quantized stationary point is an extension of the stationary point in the discrete setting, and any global solution p_l to Problem 3 is a quantized stationary point to Problem

3 (Lemma 3.7 in [34]). Then the following assumption is required for convergence analysis.

Assumption 1. *$f_l(z_l)$ is Lipschitz continuous with coefficient $S > 0$, $R(Z_L; y)$ is proper, and $F(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{z}, \boldsymbol{q})$ is coercive. Moreover, $\partial f_l(z_l)$ is bounded, i.e. there exists $M > 0$ such that $\|\partial f_l(z_l)\| \leq M$.*

Assumption 1 is mild to satisfy: most common activation functions such as Rectified Linear Unit (ReLU) [33] and leaky ReLU[36] satisfy Assumption 1. The risk function $R(z_l; y)$ is only required to be proper, which shows that the convergence condition of our proposed pdADMM-G is milder than that of the dlADMM, which requires $R(z_l; y)$ to be Lipschitz differentiable [7]. Due to the space limit, detailed proofs are provided in Section C in the Appendix. The technical proofs follow a similar routine as dlADMM [7]. The difference consists in the fact that the dual variable u_l is controlled by q_l and z_l (Lemma 6 in Section C in the Appendix), which holds under Assumption 1, while u_l can be controlled only by z_l in the convergence proof of dlADMM. The first lemma shows that the objective keeps decreasing when ρ is sufficiently large.

Lemma 1 (Objective Decrease)**.** *For both the pdADMM-G algorithm and the pdADMM-G-Q algorithm, if $\rho > \max(4\nu S^2, (\sqrt{17}+1)\nu/2)$, there exist $C_1 = \nu/2 - 2\nu^2 S^2/\rho > 0$ and $C_2 = \rho/2 - 2\nu^2/\rho - \nu/2 > 0$ such that it holds for any $k \in \mathbb{N}$ that*

$$
\begin{aligned}
&L_\rho(\boldsymbol{p}^k, \boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k) - L_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{q}^{k+1}, \boldsymbol{u}^{k+1}) \\
&\geq \sum_{l=2}^{L} (\tau_l^{k+1}/2)\|p_l^{k+1} - p_l^k\|_2^2 + \sum_{l=1}^{L} (\theta_l^{k+1}/2)\|W_l^{k+1} - W_l^k\|_2^2 \\
&+ \sum_{l=1}^{L} (\nu/2)\|b_l^{k+1} - b_l^k\|_2^2 + \sum_{l=1}^{L-1} C_1 \|z_l^{k+1} - z_l^k\|_2^2 \\
&+ (\nu/2)\|z_L^{k+1} - z_L^k\|_2^2 + \sum_{l=1}^{L-1} C_2 \|q_l^{k+1} - q_l^k\|_2^2, \qquad (1) \\
&\beta_\rho(\boldsymbol{p}^k, \boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k) - \beta_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{q}^{k+1}, \boldsymbol{u}^{k+1}) \\
&\geq \sum_{l=1}^{L} (\theta_l^{k+1}/2)\|W_l^{k+1} - W_l^k\|_2^2 + \sum_{l=1}^{L} (\nu/2)\|b_l^{k+1} - b_l^k\|_2^2 \\
&+ \sum_{l=1}^{L-1} C_1 \|z_l^{k+1} - z_l^k\|_2^2 + (\nu/2)\|z_L^{k+1} - z_L^k\|_2^2 \\
&+ \sum_{l=1}^{L-1} C_2 \|q_l^{k+1} - q_l^k\|_2^2. \qquad (2)
\end{aligned}
$$

Sketch of Proof. They can be proven via the optimality conditions of all subproblems, and Assumption 1. □

Lemma 2 shows that the objective is bounded from below when ρ is large enough, and all variables are bounded.

Lemma 2 (Bounded Objective)**.** *(1). For the pdADMM-G algorithm, if $\rho > \nu$, then $L_\rho(\boldsymbol{p}^k, \boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k)$ is lower bounded. Moreover, $\boldsymbol{p}^k, \boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k$, and $\boldsymbol{u}^k$ are bounded, i.e. there exist $\mathbb{N}_{\boldsymbol{p}}$, $\mathbb{N}_{\boldsymbol{W}}$, $\mathbb{N}_{\boldsymbol{b}}$, $\mathbb{N}_{\boldsymbol{z}}$, $\mathbb{N}_{\boldsymbol{q}}$, and $\mathbb{N}_{\boldsymbol{u}} > 0$, such that $\|\boldsymbol{p}^k\| \leq \mathbb{N}_{\boldsymbol{p}}$, $\|\boldsymbol{W}^k\| \leq \mathbb{N}_{\boldsymbol{W}}$, $\|\boldsymbol{b}^k\| \leq \mathbb{N}_{\boldsymbol{b}}$, $\|\boldsymbol{z}^k\| \leq \mathbb{N}_{\boldsymbol{z}}$, $\|\boldsymbol{q}^k\| \leq \mathbb{N}_{\boldsymbol{q}}$, and $\|\boldsymbol{u}^k\| \leq \mathbb{N}_{\boldsymbol{u}}$.*
(2). For the pdADMM-G-Q algorithm, if $\rho > \nu$, then $\beta_\rho(\boldsymbol{p}^k, \boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k)$ is lower bounded. Moreover, $\boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k$, and $\boldsymbol{u}^k$ are bounded, i.e. there exist $\overline{\mathbb{N}}_{\boldsymbol{W}}$, $\overline{\mathbb{N}}_{\boldsymbol{b}}$, $\overline{\mathbb{N}}_{\boldsymbol{z}}$, $\overline{\mathbb{N}}_{\boldsymbol{q}}$, and $\overline{\mathbb{N}}_{\boldsymbol{u}} > 0$, such that $\|\boldsymbol{W}^k\| \leq \overline{\mathbb{N}}_{\boldsymbol{W}}$, $\|\boldsymbol{b}^k\| \leq \overline{\mathbb{N}}_{\boldsymbol{b}}$, $\|\boldsymbol{z}^k\| \leq \overline{\mathbb{N}}_{\boldsymbol{z}}$, $\|\boldsymbol{q}^k\| \leq \overline{\mathbb{N}}_{\boldsymbol{q}}$, and $\|\boldsymbol{u}^k\| \leq \overline{\mathbb{N}}_{\boldsymbol{u}}$.

Sketch of Proof. We only show the sketch proof of (1) because (2) follows the same routine as (1). In order to prove the boundness of L_ρ, we should prove the following:

$$
\begin{aligned}
&L_\rho(\mathbf{p}^k, \mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^k, \mathbf{u}^k) \\
&\geq F(\mathbf{p}^k, \mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^{'}) + ((\rho - \nu)/2)\|p_{l+1}^k - q_l^k\|_2^2 \\
&> -\infty,
\end{aligned}
$$

where $p_{l+1}^k = q_l^{'}$. Therefore, $F(\mathbf{p}^k, \mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^{'})$ and $((\rho - \nu)/2)\|p_{l+1}^k - q_l^k\|_2^2$ are upper bounded by $L_\rho(\mathbf{p}^k, \mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^k, \mathbf{u}^k)$ and hence $L_\rho(\mathbf{p}^0, \mathbf{W}^0, \mathbf{b}^0, \mathbf{z}^0, \mathbf{q}^0, \mathbf{u}^0)$ (Lemma 1). The boundness of variables can be obtained via Assumption 1. □

Based on Lemmas 1 and 2, the following theorem ensures that the objective is convergent.

Theorem 1 (Convergent Objective)**.** *(1). For the pdADMM-G algorithm, if $\rho > \max(4\nu S^2, (\sqrt{17}+1)\nu/2)$, then $L_\rho(\boldsymbol{p}^k, \boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k)$ is convergent. Moreover, $\lim_{k\to\infty} \|\boldsymbol{p}^{k+1} - \boldsymbol{p}^k\|_2^2 = 0$, $\lim_{k\to\infty} \|\boldsymbol{W}^{k+1} - \boldsymbol{W}^k\|_2^2 = 0$, $\lim_{k\to\infty} \|\boldsymbol{b}^{k+1} - \boldsymbol{b}^k\|_2^2 = 0$, $\lim_{k\to\infty} \|\boldsymbol{z}^{k+1} - \boldsymbol{z}^k\|_2^2 = 0$, $\lim_{k\to\infty} \|\boldsymbol{q}^{k+1} - \boldsymbol{q}^k\|_2^2 = 0$, $\lim_{k\to\infty} \|\boldsymbol{u}^{k+1} - \boldsymbol{u}^k\|_2^2 = 0$.*
(2). For the pdADMM-G-Q algorithm, if $\rho > \max(4\nu S^2, (\sqrt{17}+1)\nu/2)$, then $\beta_\rho(\boldsymbol{p}^k, \boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k)$ is convergent. Moreover, $\lim_{k\to\infty} \|\boldsymbol{W}^{k+1} - \boldsymbol{W}^k\|_2^2 = 0$, $\lim_{k\to\infty} \|\boldsymbol{b}^{k+1} - \boldsymbol{b}^k\|_2^2 = 0$, $\lim_{k\to\infty} \|\boldsymbol{z}^{k+1} - \boldsymbol{z}^k\|_2^2 = 0$, $\lim_{k\to\infty} \|\boldsymbol{q}^{k+1} - \boldsymbol{q}^k\|_2^2 = 0$, $\lim_{k\to\infty} \|\boldsymbol{u}^{k+1} - \boldsymbol{u}^k\|_2^2 = 0$.

Sketch of Proof. This theorem can be derived by taking the limit on both sides of Inequality (1). □

The third lemma guarantees that the subgradient of the objective is upper bounded, which is stated as follows:

Lemma 3 (Bounded Subgradient)**.** *(1). For the pdADMM-G algorithm, there exists a constant $C > 0$ and $g^{k+1} \in \partial L_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{q}^{k+1}, \boldsymbol{u}^{k+1})$ such that*

$$
\begin{aligned}
\|g^{k+1}\| \leq C(&\|\boldsymbol{p}^{k+1} - \boldsymbol{p}^k\| + \|\boldsymbol{W}^{k+1} - \boldsymbol{W}^k\| + \|\boldsymbol{b}^{k+1} - \boldsymbol{b}^k\| \\
&+ \|\boldsymbol{z}^{k+1} - \boldsymbol{z}^k\| + \|\boldsymbol{q}^{k+1} - \boldsymbol{q}^k\| + \|\boldsymbol{u}^{k+1} - \boldsymbol{u}^k\|).
\end{aligned}
$$

(2). For the pdADMM-G-Q algorithm, there exists a constant $\overline{C} > 0$, $\overline{g}_{\boldsymbol{W}}^{k+1} \in \nabla_{\boldsymbol{W}^{k+1}} \beta_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{q}^{k+1}, \boldsymbol{u}^{k+1})$, $\overline{g}_{\boldsymbol{b}}^{k+1} \in \nabla_{\boldsymbol{b}^{k+1}} \beta_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{q}^{k+1}, \boldsymbol{u}^{k+1})$, $\overline{g}_{\boldsymbol{z}}^{k+1} \in \partial_{\boldsymbol{z}^{k+1}} \beta_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{q}^{k+1}, \boldsymbol{u}^{k+1})$, $\overline{g}_{\boldsymbol{q}}^{k+1} \in \nabla_{\boldsymbol{q}^{k+1}} \beta_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{q}^{k+1}, \boldsymbol{u}^{k+1})$, $\overline{g}_{\boldsymbol{u}}^{k+1} \in \nabla_{\boldsymbol{u}^{k+1}} \beta_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{q}^{k+1}, \boldsymbol{u}^{k+1})$ such

that

$$\begin{aligned}
&\|\overline{g}_{\boldsymbol{W}}^{k+1}\| \le \overline{C}(\|\boldsymbol{W}^{k+1}-\boldsymbol{W}^{k}\|+\|\boldsymbol{b}^{k+1}-\boldsymbol{b}^{k}\|\\
&+\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^{k}\|+\|\boldsymbol{q}^{k+1}-\boldsymbol{q}^{k}\|+\|\boldsymbol{u}^{k+1}-\boldsymbol{u}^{k}\|),\\
&\|\overline{g}_{\boldsymbol{b}}^{k+1}\| \le \overline{C}(\|\boldsymbol{W}^{k+1}-\boldsymbol{W}^{k}\|+\|\boldsymbol{b}^{k+1}-\boldsymbol{b}^{k}\|\\
&+\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^{k}\|+\|\boldsymbol{q}^{k+1}-\boldsymbol{q}^{k}\|+\|\boldsymbol{u}^{k+1}-\boldsymbol{u}^{k}\|),\\
&\|\overline{g}_{\boldsymbol{z}}^{k+1}\| \le \overline{C}(\|\boldsymbol{W}^{k+1}-\boldsymbol{W}^{k}\|+\|\boldsymbol{b}^{k+1}-\boldsymbol{b}^{k}\|\\
&+\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^{k}\|+\|\boldsymbol{q}^{k+1}-\boldsymbol{q}^{k}\|+\|\boldsymbol{u}^{k+1}-\boldsymbol{u}^{k}\|),\\
&\|\overline{g}_{\boldsymbol{q}}^{k+1}\| \le \overline{C}(\|\boldsymbol{W}^{k+1}-\boldsymbol{W}^{k}\|+\|\boldsymbol{b}^{k+1}-\boldsymbol{b}^{k}\|\\
&+\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^{k}\|+\|\boldsymbol{q}^{k+1}-\boldsymbol{q}^{k}\|+\|\boldsymbol{u}^{k+1}-\boldsymbol{u}^{k}\|),\\
&\|\overline{g}_{\boldsymbol{u}}^{k+1}\| \le \overline{C}(\|\boldsymbol{W}^{k+1}-\boldsymbol{W}^{k}\|+\|\boldsymbol{b}^{k+1}-\boldsymbol{b}^{k}\|\\
&+\|\boldsymbol{z}^{k+1}-\boldsymbol{z}^{k}\|+\|\boldsymbol{q}^{k+1}-\boldsymbol{q}^{k}\|+\|\boldsymbol{u}^{k+1}-\boldsymbol{u}^{k}\|).
\end{aligned}$$

Sketch of Proof. To prove this lemma, the subgradient is proven to be upper bounded by the linear combination of $\|\mathbf{p}^{k+1}-\mathbf{p}^{k}\|$, $\|\mathbf{W}^{k+1}-\mathbf{W}^{k}\|$, $\|\mathbf{b}^{k+1}-\mathbf{b}^{k}\|$, $\|\mathbf{z}^{k+1}-\mathbf{z}^{k}\|$, $\|\mathbf{q}^{k+1}-\mathbf{q}^{k}\|$, and $\|\mathbf{u}^{k+1}-\mathbf{u}^{k}\|$. □

Now based on Theorem 1, and Lemma 3, the convergence of the pdADMM-G algorithm to a stationary point is presented in the following theorem.

Theorem 2 (Convergence of the pdADMM-G algorithm)**.** *If $\rho > \max(4\nu S^2, (\sqrt{17}+1)\nu/2)$, then for the variables $(\boldsymbol{p},\boldsymbol{W},\boldsymbol{b},\boldsymbol{z},\boldsymbol{q},\boldsymbol{u})$ in Problem 2, starting from any $(\boldsymbol{p}^0,\boldsymbol{W}^0,\boldsymbol{b}^0,\boldsymbol{z}^0,\boldsymbol{q}^0,\boldsymbol{u}^0)$, $(\boldsymbol{p}^k,\boldsymbol{W}^k,\boldsymbol{b}^k,\boldsymbol{z}^k,\boldsymbol{q}^k,\boldsymbol{u}^k)$ has at least a limit point $(\boldsymbol{p}^*,\boldsymbol{W}^*,\boldsymbol{b}^*,\boldsymbol{z}^*,\boldsymbol{q}^*,\boldsymbol{u}^*)$, and any limit point is a stationary point of Problem 2. That is, $0 \in \partial L_\rho(\boldsymbol{p}^*,\boldsymbol{W}^*,\boldsymbol{b}^*,\boldsymbol{z}^*,\boldsymbol{q}^*,\boldsymbol{u}^*)$. In other words,*

$$\begin{aligned}
&p_{l+1}^*=q_l^*,\ \nabla_{\boldsymbol{p}^*}L_\rho(\boldsymbol{p}^*,\boldsymbol{W}^*,\boldsymbol{b}^*,\boldsymbol{z}^*,\boldsymbol{q}^*,\boldsymbol{u}^*)=0,\\
&\nabla_{\boldsymbol{W}^*}L_\rho(\boldsymbol{p}^*,\boldsymbol{W}^*,\boldsymbol{b}^*,\boldsymbol{z}^*,\boldsymbol{q}^*,\boldsymbol{u}^*)=0,\nabla_{\boldsymbol{b}^*}L_\rho(\boldsymbol{p}^*,\boldsymbol{W}^*,\boldsymbol{b}^*,\boldsymbol{z}^*,\boldsymbol{q}^*,\boldsymbol{u}^*)=0,\\
&0\in\partial_{\boldsymbol{z}^*}L_\rho(\boldsymbol{p}^*,\boldsymbol{W}^*,\boldsymbol{b}^*,\boldsymbol{z}^*,\boldsymbol{q}^*,\boldsymbol{u}^*),\nabla_{\boldsymbol{q}^*}L_\rho(\boldsymbol{p}^*,\boldsymbol{W}^*,\boldsymbol{b}^*,\boldsymbol{z}^*,\boldsymbol{q}^*,\boldsymbol{u}^*)=0.
\end{aligned}$$

Sketch of Proof. This theorem can be derived directly from Lemma 2 and Lemma 3. □

Theorem 2 shows that our proposed pdADMM-G algorithm converges for sufficiently large ρ, which is consistent with previous literature [7]. Similarly, the convergence of the proposed pdADMM-G-Q algorithm is shown as follows:

Theorem 3 (Convergence of the pdADMM-G-Q algorithm)**.** *If $\rho > \max(4\nu S^2, (\sqrt{17}+1)\nu/2)$, then for the variables $(\boldsymbol{p},\boldsymbol{W},\boldsymbol{b},\boldsymbol{z},\boldsymbol{q},\boldsymbol{u})$ in Problem 3, starting from any $(\boldsymbol{p}^0,\boldsymbol{W}^0,\boldsymbol{b}^0,\boldsymbol{z}^0,\boldsymbol{q}^0,\boldsymbol{u}^0)$, $(\boldsymbol{p}^k,\boldsymbol{W}^k,\boldsymbol{b}^k,\boldsymbol{z}^k,\boldsymbol{q}^k,\boldsymbol{u}^k)$ has at least a limit point $(\boldsymbol{p}^*,\boldsymbol{W}^*,\boldsymbol{b}^*,\boldsymbol{z}^*,\boldsymbol{q}^*,\boldsymbol{u}^*)$, and any limit point $(\boldsymbol{W}^*,\boldsymbol{b}^*,\boldsymbol{z}^*,\boldsymbol{q}^*,\boldsymbol{u}^*)$ is a stationary point of Problem 3. Moreover, if τ_l^{k+1} is bounded, then $\boldsymbol{p}^*$ is a quantized stationary point of Problem 3. That is*

$$\begin{aligned}
&p_{l+1}^*=q_l^*,\ \nabla_{\boldsymbol{W}^*}\beta_\rho(\boldsymbol{p}^*,\boldsymbol{W}^*,\boldsymbol{b}^*,\boldsymbol{z}^*,\boldsymbol{q}^*,\boldsymbol{u}^*)=0,\\
&\nabla_{\boldsymbol{b}^*}\beta_\rho(\boldsymbol{p}^*,\boldsymbol{W}^*,\boldsymbol{b}^*,\boldsymbol{z}^*,\boldsymbol{q}^*,\boldsymbol{u}^*)=0,\\
&0\in\partial_{\boldsymbol{z}^*}\beta_\rho(\boldsymbol{p}^*,\boldsymbol{W}^*,\boldsymbol{b}^*,\boldsymbol{z}^*,\boldsymbol{q}^*,\boldsymbol{u}^*),\\
&\nabla_{\boldsymbol{q}^*}\beta_\rho(\boldsymbol{p}^*,\boldsymbol{W}^*,\boldsymbol{b}^*,\boldsymbol{z}^*,\boldsymbol{q}^*,\boldsymbol{u}^*)=0,\\
&p_l^*\in\arg\min_{\delta\in\Delta}\|\delta-(p_l^*-\nabla_{p_l^*}F(\boldsymbol{p}^*,\boldsymbol{W}^*,\boldsymbol{b}^*,\boldsymbol{z}^*,\boldsymbol{q}^*)/\tau_l^*)\|.
\end{aligned}$$

where τ_l^* is a limit point of τ_l^k.

Dataset	Node#	Edge#	Class#	Feature#	Training Set#	Validation Set#	Test Set#
Cora	2,485	10,556	7	1,433	140	500	1,000
PubMed	19,717	88,648	3	500	60	500	1,000
Citeseer	2,110	9,104	6	3,703	120	500	1,000
Amazon Computers	13,381	491,722	10	767	200	1,000	1,000
Amazon Photo	7,487	238,162	8	745	160	1,000	1,000
Coauthor CS	18,333	163,788	15	6,805	300	1,000	1,000
Coauthor Physics	34,493	495,924	5	8,415	100	1,000	1,000
Flickr	89,250	899,756	7	500	44,625	22,312	22,313
Ogbn-Arxiv	169,343	1,166,243	40	128	90,941	29,799	48,603

TABLE II: Dataset statistics.

Sketch of Proof. This theorem is proven using a similar procedure as Theorem 2, and the definition of the quantized stationary point. □

The only difference between Theorems 2 and 3 is that $\mathbf{p}^*$ is a stationary point in Problem 2 and a quantized stationary point in Problem 3. Next, the following theorem ensures the sublinear convergence rate $o(1/k)$ of the proposed pdADMM-G algorithm and the pdADMM-G-Q algorithm.

Theorem 4 (Convergence Rate)**.** *(1). For the pdADMM-G algorithm and a sequence $(\boldsymbol{p}^k,\boldsymbol{W}^k,\boldsymbol{b}^k,\boldsymbol{z}^k,\boldsymbol{q}^k,\boldsymbol{u}^k)$, define $c_k = \min_{0\le i\le k}(\sum_{l=2}^{L}(\tau_l^{i+1}/2)\|p_l^{i+1}-p_l^i\|_2^2+\sum_{l=1}^{L}(\theta_l^{i+1}/2)\|W_l^{i+1}-W_l^i\|_2^2+\sum_{l=1}^{L}(\nu/2)\|b_l^{i+1}-b_l^i\|_2^2+\sum_{l=1}^{L-1}C_1\|z_l^{i+1}-z_l^i\|_2^2+(\nu/2)\|z_L^{i+1}-z_L^i\|_2^2+\sum_{l=1}^{L-1}C_2\|q_l^{i+1}-q_l^i\|_2^2)$ where $C_1=\nu/2-2\nu^2S^2/\rho>0$ and $C_2=\rho/2-2\nu^2/\rho-\nu/2>0$, then the convergence rate of c_k is $o(1/k)$.*
(2). For the pdADMM-G-Q algorithm and a sequence $(\boldsymbol{W}^k,\boldsymbol{b}^k,\boldsymbol{z}^k,\boldsymbol{q}^k,\boldsymbol{u}^k)$, define $d_k = \min_{0\le i\le k}(\sum_{l=1}^{L}(\theta_l^{i+1}/2)\|W_l^{i+1}-W_l^i\|_2^2+\sum_{l=1}^{L}(\nu/2)\|b_l^{i+1}-b_l^i\|_2^2+\sum_{l=1}^{L-1}C_1\|z_l^{i+1}-z_l^i\|_2^2+(\nu/2)\|z_L^{i+1}-z_L^i\|_2^2+\sum_{l=1}^{L-1}C_2\|q_l^{i+1}-q_l^i\|_2^2)$ where $C_1=\nu/2-2\nu^2S^2/\rho>0$ and $C_2=\rho/2-2\nu^2/\rho-\nu/2>0$, then the convergence rate of d_k is $o(1/k)$.

Sketch of Proof. (1). In order to prove the convergence rate $o(1/k)$, c_k satisfies three conditions: (a) $c_k \ge c_{k+1}$, (b) $\sum_{k=0}^{\infty}c_k$ is bounded, and (c) $c_k \ge 0$.
(2). d_k can be proven using a similar procedure as (1). □

V. Experiments

In this section, we evaluate the performance of the proposed pdADMM-G algorithm and the proposed pdADMM-G-Q algorithm on GA-MLP models using nine benchmark datasets. Convergence and computational overheads are demonstrated on different datasets. Speedup and test performance are compared with several state-of-the-art optimizers. All experiments were conducted on the Amazon Web Services (AWS) p2.16xlarge instance, with 16 NVIDIA K80 GPUs, 64vCPUs, a processor Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz, and 732GiB of RAM.

A. Datasets and Settings

Nine benchmark datasets were used for experimental evaluation, whose statistics are shown in Table II. Each dataset is split into a training set, a validation set, and a test set. Due to space limit, their details can be found in Section D1 in the Appendix.

When it comes to experimental settings, we set $K = 4$ for the multi-hop operator Ψ, and defined a diagonal degree matrix D where $D_{ii} = \sum_{j=1}^{|V|} A_{ij}$, and a renormalized adjacency matrix $\tilde{A} = (D + I)^{-1/2}(A + I)(D + I)^{-1/2} \in \mathbb{R}^{|V|\times|V|}$[1]. Moreover, we set $\Psi = \{I, \tilde{A}, \tilde{A}^2, \tilde{A}^3\}$ [2]. For all GA-MLP models, the activation function was set to ReLU. The loss function was set to the cross-entropy loss. For the pdADMM-G-Q algorithm, $\Delta = \{-1, 0, 1, \cdots, 20\}$ in Problem 3, and $\mathbf{p}$ was quantized by default.

B. Comparison Methods

GD and its variants are state-of-the-art optimizers and hence served as comparison methods. For GD-based methods, all datasets were used for training models in a full-batch fashion. All hyperparameters were chosen by maximizing the performance of validation sets. Due to space limit, hyperparameter settings of all methods are shown in Section D2 in the Appendix. The following are their brief introductions:

1. Gradient Descent (GD) [37]. The GD and its variants are the most popular deep learning optimizers. The GD updates parameters simply based on their gradients.

2. Adaptive learning rate method (Adadelta) [38]. The Adadelta is proposed to overcome the sensitivity to hyperparameter selection.

3. Adaptive gradient algorithm (Adagrad) [39]. Adagrad is an improved version of GD: rather than fixing the learning rate during training, it adapts the learning rate to the hyperparameter.

4. Adaptive momentum estimation (Adam) [40]. Adam is the most popular optimization method for deep learning models. It estimates the first and second momentum in order to correct the biased gradient, and thus accelerates empirical convergence.

C. Convergence

Firstly, in order to validate the convergence of two proposed algorithms, we set up a GA-MLP model with 10 layers, each of which has $1{,}000$ neurons. The number of epochs was set to 100. ν and ρ were set to 0.01 and 1, respectively.

Figure 2 demonstrates objectives and residuals of two proposed algorithms on four datasets. Overall, the objectives and residuals of the two proposed algorithms are convergent. From Figure 2(a) and Figure 2(c), the objectives of the two proposed algorithms decrease drastically at the first 50 epochs and then drop smoothly to the end. The objectives on the PubMed dataset achieve the lowest among all four datasets, whereas these on the Coauthor CS dataset are the highest, which still reach near 10^5 at the 100-th epoch. As for residuals, even though the residuals of the pdADMM-G-Q algorithm are higher than these of the pdADMM-G algorithm initially, they both converge sublinearly to 0, which is consistent with Theorem 2 and Theorem 3. Specifically, as shown in Figure 2(b) and Figure 2(d), the residuals on the Cora dataset decrease more slowly with fluctuation than these on other datasets, while residuals on the Amazon Computers and Amazon Photo datasets demonstrate the fastest decreasing speed at the first 40 epochs before reaching a plateau less than 10^{-6}. The residuals on the PubMed dataset accomplish the lowest values among all four datasets again with a value of less than 10^{-7}.

D. Speedup

Next, we investigate the speedup of the pdADMM-G algorithm in the large deep GA-MLP models. The running time per epoch was an average of 10 epochs. ρ and ν were both set to 10^{-3}. We investigate the speedup concerning two factors: the number of layers and the number of GPUs.

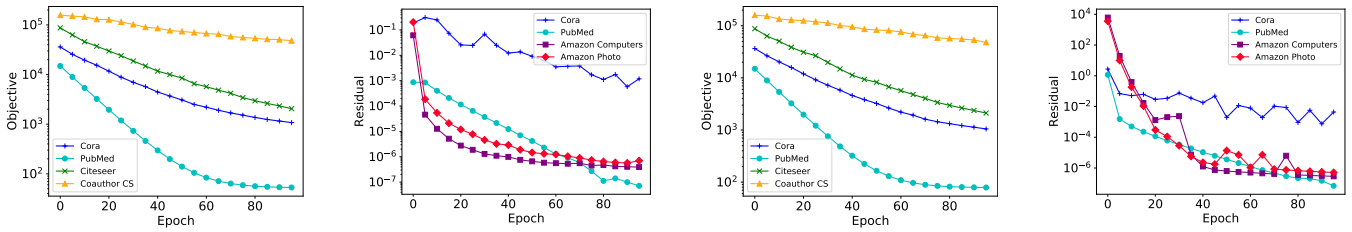
For the relationship between the speedup and the number of layers, the pdADMM-G algorithm in the GA-MLP models with $4{,}000$ neurons was tested. The number of layers ranged from 8 to 17. The speedup on small datasets and large datasets are shown in Figure 3(a) and Figure 3(b), respectively. Overall, the speedup of the proposed pdADMM-G increases linearly with the number of layers. For example, the speedups on the Cora dataset and the Amazon Computers dataset rise from 3 and 3.5 gradually to 4 and 4.5, respectively. The speedup on the PubMed dataset achieves the lowest with a value of less than 3, whereas that on the Coauthor CS dataset at least doubles that on any other small dataset, with a peak of 6. Moreover, the speedup is more obvious on large datasets. For example, when the slopes of speedups are compared, the slope on the Flickr dataset is at least five times much steeper than that on the Coauthor CS dataset. The same trend is applied to the Ogbn-Arxiv dataset. This means that our proposed pdADMM-G algorithm is more suitable for large datasets.

For the relationship between the speedup and the number of GPUs, we set up a large GA-MLP model with 16 layers and $4{,}000$ neurons and kept all hyperparameters in the previous experiment. The speedup of our proposed pdADMM-G algorithm was compared with all comparison methods. Figure 4 shows all speedups on two large datasets. The proposed pdADMM-G algorithm achieves a higher speedup than any GD-based method. For example, the speedups of 8 GPUs are nearly 8 on the Flickr dataset and the Ogbn-Arxiv dataset, while the best speedups achieved via comparison methods are in the vicinity of 6 and 5 on two datasets, respectively. We also observe that while speedups of all methods scale linearly with the number of GPUs, the slopes of our proposed pdADMM-G algorithm are steeper than these of any comparison method. For example, the slope of our proposed pdADMM-G algorithm on the Flickr dataset is more than 10 times steeper than that of Adam. All comparison methods show similar flat slopes, and they achieve a higher slope of the speedup on the Ogbn-Arxiv dataset than that on the Flickr dataset.

In summary, the speedup of our proposed pdADMM-G algorithm scales linearly with the number of layers and the number of GPUs. Moreover, its speedup is superior to any other comparison method significantly by more than 10 times.

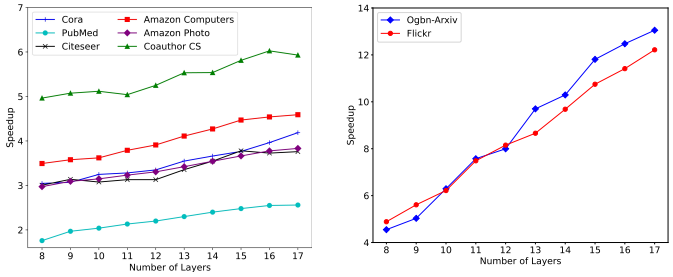
E. Communication Overheads

Then, it is necessary to explore how many communication overheads can be reduced using the proposed quantization technique on different quantization levels. To achieve this, we established a large GA-MLP model with 10 layers, each of which consists of $1{,}000$ neurons. We set up three quantization cases: no quantization, the quantization concerning $\mathbf{p}$ only, and the quantization concerning both $\mathbf{p}$ and $\mathbf{q}$. For every quantization case, we also set up two different quantization



(a). pdADMM-G Objective. (b). pdADMM-G Residual. (c). pdADMM-G-Q Objective. (d). pdADMM-G-Q Residual.

Fig. 2: Convergence curves of the pdADMM-G algorithm and the pdADMM-G-Q algorithm in four datasets: they both converge.



(a). The speedup on small datasets.

(b). The speedup on large datasets.

Fig. 3: The speedup of the proposed pdADMM-G on different datasets concerning the number of layers: the speedup increases linearly with the number of layers, and the slopes of speedups are higher on large datasets than those on small datasets.

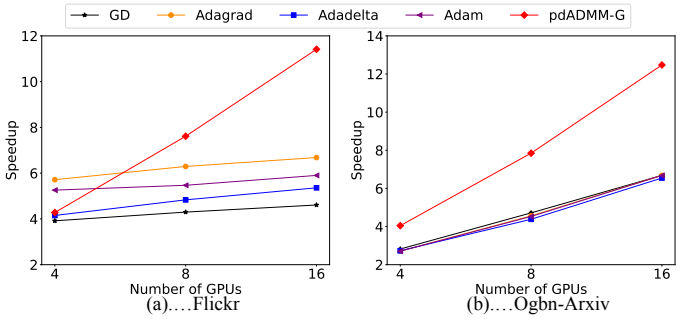


Fig. 4: The speedup of all methods on two large datasets concerning the number of GPUs: speedups of the proposed pdADMM-G are higher than these of all comparison methods.

Dataset	Cora	PubMed	Citeseer
GD	0.730±0.022	0.638± 0.080	0.637±0.040
Adadelta	0.671±0.064	0.705±0.038	0.620±0.016
Adagrad	0.726±0.025	0.753± 0.015	0.601±0.037
Adam	0.725±0.036	0.742±0.007	0.631±0.018
pdADMM-G	0.784±0.003	**0.784±0.004**	0.709±0.003
pdADMM-G-Q	**0.788±0.003**	0.782±0.003	**0.712± 0.001**
Dataset	Amazon Computers	Amazon Photo	Coauthor CS
GD	0.646±0.032	0.735± 0.169	0.884±0.010
Adadelta	0.136±0.060	0.369± 0.045	0.787±0.086
Adagrad	0.688±0.023	0.813± 0.018	0.887±0.007
Adam	0.724±0.010	0.855±0.009	0.883±0.009
pdADMM-G	**0.735±0.006**	**0.856±0.011**	**0.915±0.004**
pdADMM-G-Q	0.687± 0.054	0.832±0.010	0.914±0.003
Dataset	Coauthor Physics	Flickr	Ogbn-Arxiv
GD	0.909±0.007	0.466±0.007	0.361±0.063
Adadelta	0.915±0.014	0.461±0.008	0.523± 0.030
Adagrad	0.916±0.012	0.481±0.003	0.567±0.016
Adam	0.912±0.016	0.512 ±0.004	**0.674± 0.006**
pdADMM-G	**0.921±0.003**	**0.513±0.002**	0.647±0.002
pdADMM-G-Q	0.919±0.002	0.507±0.003	0.655±0.002

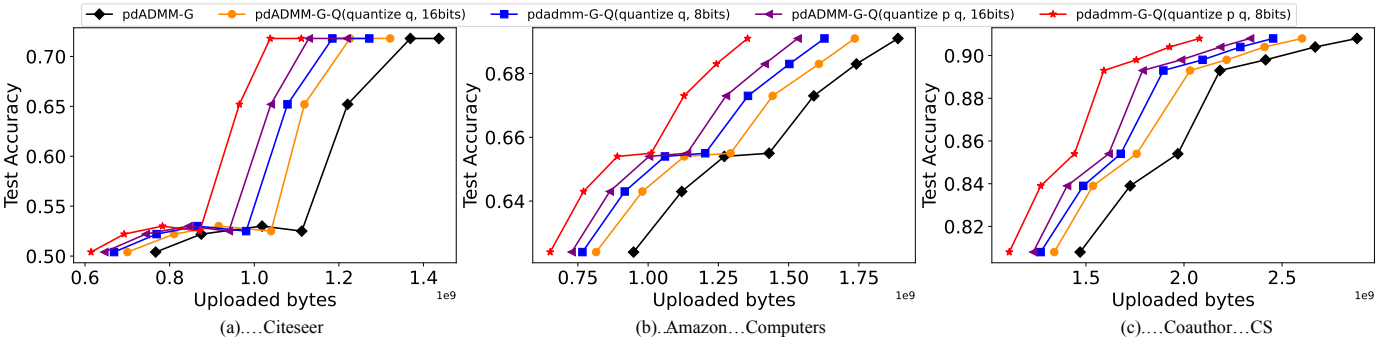
TABLE III: The test performance of all methods when the number of neurons is 100: two proposed algorithms outperform all comparison methods.

sizes: 8 bits and 16 bits. Figure 5 demonstrates the relationship between the test accuracy and communication overheads for different quantization cases and sizes on three datasets. Overall, communication overheads can be reduced significantly by the proposed quantization technique. The amount of reduction depends on different quantization cases and sizes. Generally speaking, the more variables are quantized and the fewer bits are compressed, then the more savings in communications can be achieved. Take the Citeseer dataset as an example, while all algorithms reach the same test accuracy above 70%, the proposed pdADMM-G (i.e. no quantization) consumes the most communication costs with the value of around 1.4×10^9 bytes. If the variable $\mathbf{p}$ is quantized using 16 bits, the communication overhead drops by 10%, and then using 8 bits saves another 5%. When variables $\mathbf{p}$ and $\mathbf{q}$ are both quantized, the communication overhead tumbles down to 1.2×10^9 bytes, which means decreases by 16.7% when it is compared with the case where only $\mathbf{p}$ is quantized. When variables are compressed to 8 bits instead of 16 bits, the communication overhead slips further to 1.1×10^9, a nearly 30% decline. The same trend is applied to the other two datasets, and they accomplish a shrink of communication overheads by 33% and 45%, respectively. This demonstrates that our proposed quantization technique is effective for reducing unnecessary communication costs without loss of performance. We also observe that the Coauthor CS dataset is the largest dataset among the three, and it accomplishes the greatest communication reduction.

F. Performance

Finally, we evaluate the performance of two proposed algorithms against all comparison methods on nine benchmark datasets. We set up two standard GA-MLP models with 10 layers but different neurons: the first model has 100 neurons, while the second model has 500 neurons. Following the greedy layerwise training strategy [31], we firstly trained a two-layer GA-MLP model, and then three more layers were added to training, and finally, all 10 layers were involved. The number of epochs was set to 200. We repeated all experiments five times and reported their means and the standard deviations. Due to space limit, hyperparameter settings and the performance of validation sets are shown in Section D2 and D3 in the Appendix, respectively.

Table III demonstrates the performance of all methods when



(a)…Citeseer (b)..Amazon…Computers (c)….Coauthor…CS

Fig. 5: Communication overheads of two proposed algorithms on three datasets: the quantization concerning both **p** and **q** using 8 bits reduces the communication overheads by up to 45% without loss of performance.

Dataset	Cora	PubMed	Citeseer
GD	0.757±0.024	0.699±0.655	0.680±0.014
Adadelta	0.717±0.063	0.722±0.696	0.564±0.028
Adagrad	0.776±0.013	0.759±0.761	0.650 ±0.038
Adam	0.771±0.020	0.778±0.767	0.662 ±0.021
pdADMM-G	**0.786±0.005**	0.786±0.786	**0.713±0.007**
pdADMM-G-Q	**0.786±0.005**	**0.788±0.787**	0.712±0.005
Dataset	Amazon Computers	Amazon Photo	Coauthor CS
GD	0.707±0.012	0.817±0.005	0.906±0.005
Adadelta	0.243±0.063	0.380±0.069	0.880±0.011
Adagrad	**0.753±0.009**	0.866±0.007	0.911±0.003
Adam	0.739±0.022	**0.880± 0.016**	0.898±0.013
pdADMM-G	0.751±0.008	0.873±0.004	**0.920±0.002**
pdADMM-G-Q	0.748±0.004	0.865±0.007	0.919±0.003
Dataset	Coauthor Physics	Flickr	Ogbn-Arxiv
GD	0.917±0.004	0.466±0.001	0.436±0.042
Adadelta	0.917±0.004	0.462±0.001	0.584±0.031
Adagrad	0.914±0.004	0.487±0.005	0.630±0.016
Adam	0.914±0.002	**0.517±0.002**	**0.682±0.010**
pdADMM-G	**0.918±0.003**	0.515±0.002	0.655±0.001
pdADMM-G-Q	**0.918±0.002**	0.512±0.003	0.657±0.002

TABLE IV: The test performance of all methods when the number of neurons is 500: two proposed algorithms outperform all comparison methods.

the number of neurons is 100. In summary, the two proposed algorithms outperform all comparison methods slightly: they occupy the best algorithms on eight datasets out of the total nine datasets. For example, they both achieve 78% test accuracy on the Cora dataset, whereas the best comparison method is GD, which only reaches 73% test accuracy, and is at least 6% lower than the two proposed algorithms. As another example, two proposed algorithms accomplish 78% test accuracy on the PubMed dataset, 4% better than that achieved by Adagrad, whose performance is the best aside from the two proposed algorithms. The Citeseer dataset shows the largest performance gap between the two proposed algorithms and all comparison methods. Two proposed algorithms reach the level of 70% test accuracy, whereas all comparison methods fall in the vicinity of 60% test accuracy. In other words, the two proposed algorithms outperform all comparison methods by more than 10%. For two proposed algorithms, the proposed pdADMM-G algorithm outperforms marginally the proposed pdADMM-G-Q algorithm due to the quantization technique. Their largest performance gap is 5%, which is achieved on the Amazon Computers dataset. The Adam is the best comparison method overall, and it serves as the best algorithm on the Ogbn-Arxiv dataset. The Adadelta performs the worst among all comparison methods, whose performance is significantly lower than any other method on several datasets such as the Amazon Computers dataset, the Amazon Photo dataset, and the Coauthor CS dataset. Last but not least, the standard deviations of all methods remain low, and this shows that they are robust to different initializations.

Table IV shows the performance of all methods when the number of neurons is 500. In general, two proposed algorithms still reach a better performance than all comparison methods, but the gap is more narrow. For example, in Table III, the proposed pdADMM-G algorithm achieves the best on the Amazon Computers dataset. However, it is surpassed by Adagrad slightly in Table IV. We also observe that a GA-MLP model with 500 neurons performs better than that with 100 neurons, which are trained by the same algorithm. This makes sense since the wider a model is, the more expressiveness it is equipped with.

VI. Conclusion

The GA-MLP models are attractive to the deep learning community due to potential resistance to some problems of GNNs such as over-smoothing and over-squashing. In this paper, we propose a novel pdADMM-G algorithm to achieve parallel training of GA-MLP models, which is accomplished by breaking the layer dependency. The extended pdADMM-G-Q algorithm reduces communication overheads by the introduction of the quantization technique. Their theoretical convergence to a (quantized) stationary point of the problem is guaranteed with a sublinear convergence rate $o(1/k)$, where k is the number of iterations. Extensive experiments verify that the two proposed algorithms not only converge in terms of objectives and residuals, and accelerate the training of deep GA-MLP models, but also stand out among all the existing state-of-the-art optimizers on nine benchmark datasets. Moreover, the pdADMM-G-Q algorithm reduces communication overheads by up to 45% without loss of performance.

References

[1] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[2] L. Chen, Z. Chen, and J. Bruna, "On graph neural networks versus graph-augmented mlps," in *Ninth International Conference on Learning Representations*, 2021.

[3] J. Topping, F. D. Giovanni, B. P. Chamberlain, X. Dong, and M. M. Bronstein, "Understanding over-squashing and bottlenecks on graphs via curvature," in *International Conference on Learning Representations*, 2022.

[4] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1025–1035, 2017.

[5] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *International conference on machine learning*, pp. 6861–6871, PMLR, 2019.

[6] G. Taylor, R. Burmeister, Z. Xu, B. Singh, A. Patel, and T. Goldstein, "Training neural networks without gradients: A scalable admm approach," in *International conference on machine learning*, pp. 2722–2731, 2016.

[7] J. Wang, F. Yu, X. Chen, and L. Zhao, "Admm for efficient deep learning with global convergence," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 111–119, 2019.

[8] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," in *Advances in neural information processing systems*, pp. 1509–1519, 2017.

[9] A. Sergeev and M. Del Balso, "Horovod: fast and easy distributed deep learning in tensorflow," *preprint arXiv:1802.05799*, 2018.

[10] B. C. Ooi, K.-L. Tan, S. Wang, W. Wang, Q. Cai, G. Chen, J. Gao, Z. Luo, A. K. Tung, Y. Wang, *et al.*, "Singa: A distributed deep learning platform," in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 685–688, ACM, 2015.

[11] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," *preprint arXiv:1512.01274*, 2015.

[12] S. H. Hashemi, S. A. Jyothi, and R. H. Campbell, "Tictac: Accelerating distributed deep learning with communication scheduling," in *Proceedings of the 2nd SysML Conference*, 2019.

[13] H. Zhang, Z. Zheng, S. Xu, W. Dai, Q. Ho, X. Liang, Z. Hu, J. Wei, P. Xie, and E. P. Xing, "Poseidon: An efficient communication architecture for distributed deep learning on {GPU} clusters," in *2017 {USENIX} Annual Technical Conference ({USENIX}{ATC} 17)*, pp. 181–193, 2017.

[14] M. Zinkevich, M. Weimer, L. Li, and A. J. Smola, "Parallelized stochastic gradient descent," in *Advances in neural information processing systems*, pp. 2595–2603, 2010.

[15] P. Parpas and C. Muir, "Predict globally, correct locally: Parallel-in-time optimal control of neural networks," *preprint arXiv:1902.02542*, 2019.

[16] Z. Huo, B. Gu, and H. Huang, "Training neural networks using features replay," in *Advances in Neural Information Processing Systems*, pp. 6659–6668, 2018.

[17] H. Zhuang, Y. Wang, Q. Liu, and Z. Lin, "Fully decoupled neural network learning using delayed gradients," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[18] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, 2020.

[19] C. Gallicchio and A. Micheli, "Graph echo state networks," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, 2010.

[20] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *International Conference on Learning Representations (ICLR)*, 2016.

[21] H. Dai, Z. Kozareva, B. Dai, A. Smola, and L. Song, "Learning steady-states of iterative algorithms over graphs," in *International conference on machine learning*, pp. 1106–1114, PMLR, 2018.

[22] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *International Conference on Learning Representations (ICLR)*, 2014.

[23] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *preprint arXiv:1506.05163*, 2015.

[24] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 3844–3852, 2016.

[25] S. Cao, W. Lu, and Q. Xu, "Deep neural networks for learning graph representations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[26] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1225–1234, 2016.

[27] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao, and C. Zhang, "Adversarially regularized graph autoencoder for graph embedding," in *IJCAI International Joint Conference on Artificial Intelligence*, 2018.

[28] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *International Conference on Neural Information Processing*, pp. 362–373, Springer, 2018.

[29] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.

[30] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 5308–5317, 2016.

[31] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems*, vol. 19, 2006.

[32] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2022.

[33] J. Wang, Z. Chai, Y. Cheng, and L. Zhao, "Toward model parallelism for deep neural network based on gradient-free admm framework," in *Proceedings of the 20th IEEE International Conference on Data Mining*, ICDM '20, 2020.

[34] T. Huang, P. Singhania, M. Sanjabi, P. Mitra, and M. Razaviyayn, "Alternating direction method of multipliers for quantization," in *International Conference on Artificial Intelligence and Statistics*, pp. 208–216, PMLR, 2021.

[35] R. T. Rockafellar and R. J.-B. Wets, *Variational analysis*, vol. 317. Springer Science & Business Media, 2009.

[36] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[37] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, Springer, 2010.

[38] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *preprint arXiv:1212.5701*, 2012.

[39] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[41] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[42] J. Wang and L. Zhao, "Nonconvex generalization of alternating direction method of multipliers for nonlinear equality constrained problems," *Results in Control and Optimization*, p. 100009, 2021.

[43] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block admm with o (1/k) convergence," *Journal of Scientific Computing*, vol. 71, no. 2, pp. 712–736, 2017.

[44] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI magazine*, vol. 29, no. 3, pp. 93–93, 2008.

[45] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pp. 43–52, 2015.

[46] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann, "Pitfalls of graph neural network evaluation," *Relational Representation Learning Workshop (R2L),NeurIPS*, 2018.

[47] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna, "Graphsaint: Graph sampling based inductive learning method," in *International Conference on Learning Representations*, 2020.

[48] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," in *Advances in neural information processing systems*, pp. 22118–22133, 2020.

APPENDIX

A. Solutions to Subproblems of the pdADMM-G Algorithm

We discuss how to solve all subproblems generated by pdADMM-G in detail.

1) Update $\boldsymbol{p}_l^{k+1}$: The variable $\mathbf{p}_l^{k+1}$ is updated as follows:

$$p_l^{k+1} \leftarrow \arg\min_{p_l} L_\rho(\mathbf{p}, \mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^k, \mathbf{u}^k) = \phi(p_l, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k).$$

Because W_l and p_l are coupled in ϕ, solving p_l should require the time-consuming operation of matrix inversion of W_l. To handle this, we apply similar quadratic approximation techniques as used in dlADMM [7] as follows:

$$p_l^{k+1} \leftarrow \arg\min_{p_l} U_l(p_l; \tau_l^{k+1}), \tag{3}$$

where $U_l(p_l; \tau_l^{k+1}) = \phi(p_l^k, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k) + \nabla_{p_l^k}\phi^T(p_l^k, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k)(p_l - p_l^k) + (\tau_l^{k+1}/2)\|p_l - p_l^k\|_2^2$, and $\tau_l^{k+1} > 0$ is a parameter. τ_l^{k+1} should satisfy $\phi(p_l^{k+1}, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k) \le U_l(p_l^{k+1}; \tau_l^{k+1})$. The solution to Equation (3) is: $p_l^{k+1} \leftarrow p_l^k - \nabla_{p_l^k}\phi(p_l^k, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k)/\tau_l^{k+1}$.

2) Update $\boldsymbol{W}_l^{k+1}$: The variable $\mathbf{W}_l^{k+1}$ is updated as follows:

$$W_l^{k+1} \leftarrow \arg\min_{W_l} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^k, \mathbf{u}^k) = \arg\min_{W_l} \begin{cases} \phi(p_1^{k+1}, W_1, b_1^k, z_1^k), & l = 1, \\ \phi(p_l^{k+1}, W_l, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k), & 1 < l \le L. \end{cases}$$

Similar to updating p_l, the following subproblem should be solved instead:

$$W_l^{k+1} \leftarrow \arg\min_{W_l} V_l(W_l; \theta_l^{k+1}), \tag{4}$$

where

$$V_1(W_1; \theta_1^{k+1}) = \phi(p_1^{k+1}, W_1^k, b_1^k, z_1^k) + \nabla_{W_1^k}\phi^T(p_1^{k+1}, W_1^k, b_1^k, z_1^k)(W_1 - W_1^k) + (\theta_l^{k+1}/2)\|W_1 - W_1^k\|_2^2,$$
$$V_l(W_l; \theta_l^{k+1}) = \phi(p_l^{k+1}, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k) + \nabla_{W_l^k}\phi^T(p_l^{k+1}, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k)(W_l - W_l^k) + (\theta_l^{k+1}/2)\|W_l - W_l^k\|_2^2,$$

and θ_l^{k+1} is a parameter, which should satisfy $\phi(p_1^{k+1}, W_1^{k+1}, b_1^k, z_1^k) \le V(W_1^{k+1}; \theta_1^{k+1})$ and $\phi(p_l^{k+1}, W_l^{k+1}, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k) \le V(W_l^{k+1}; \theta_l^{k+1})(1 < l < L)$. The solution to Equation (4) is shown as follows:

$$W_l^{k+1} \leftarrow W_l^k - \begin{cases} \nabla_{W_1^k}\phi(p_1^{k+1}, W_1^k, b_1^k, z_1^k)/\theta_l^{k+1}, & l = 1, \\ \nabla_{W_l^k}\phi(p_l^{k+1}, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k)/\theta_l^{k+1}, & 1 < l \le L. \end{cases}$$

3) Update $\boldsymbol{b}_l^{k+1}$: The variable $\mathbf{b}_l^{k+1}$ is updated as follows:

$$b_l^{k+1} \leftarrow \arg\min_{b_l} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}, \mathbf{z}^k, \mathbf{q}^k, \mathbf{u}^k) = \arg\min_{b_l} \begin{cases} \phi(p_1^{k+1}, W_1^{k+1}, b_1, z_1^k), & l = 1, \\ \phi(p_l^{k+1}, W_l^{k+1}, b_l, z_l^k, q_{l-1}^k, u_{l-1}^k), & 1 < l \le L. \end{cases}$$

Similarly, we solve the following subproblems instead:

$$b_1^{k+1} \leftarrow \arg\min_{b_1} \phi(p_1^{k+1}, W_1^{k+1}, b_1^k, z_1^k) + \nabla_{b_1^k}\phi^T(p_1^{k+1}, W_1^{k+1}, b_1^k, z_1^k)(b_l - b_l^k) + (\nu/2)\|b_l - b_l^k\|_2^2,$$

$$b_l^{k+1} \leftarrow \arg\min_{b_l} \phi(p_l^{k+1}, W_l^{k+1}, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k) + \nabla_{b_l^k}\phi^T(p_l^{k+1}, W_l^{k+1}, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k)(b_l - b_l^k) + (\nu/2)\|b_l - b_l^k\|_2^2 (1 < l \le L). \tag{5}$$

The solution to Equation (5) is:

$$b_l^{k+1} \leftarrow b_l^k - \begin{cases} \nabla_{b_1^k}\phi(p_1^{k+1}, W_1^{k+1}, b_1^k, z_1^k)/\nu, & l = 1, \\ \nabla_{b_l^k}\phi(p_l^{k+1}, W_l^{k+1}, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k)/\nu, & 1 < l \le L. \end{cases}$$

4) Update $\mathbf{z}^{k+1}$: The variable $\mathbf{z}^{k+1}$ is updated as follows:

$$z_l^{k+1} \leftarrow \arg\min_{z_l} (\nu/2)\|z_l - W_l^{k+1} p_l^{k+1} - b_l^{k+1}\|_2^2 + (\nu/2)\|q_l^k - f_l(z_l)\|_2^2 + (\nu/2)\|z_l - z_l^k\|_2^2 (l < L), \tag{6}$$

$$z_L^{k+1} \leftarrow \arg\min_{z_l} R(z_L; y) + (\nu/2)\|z_L - W_L^{k+1} p_L^{k+1} - b_L^{k+1}\|_2^2. \tag{7}$$

where a quadratic term $(\nu/2)\|z_l - z_l^k\|_2^2$ is added in Equation (6) to control z_l^{k+1} to close to z_l^k. Equation (7) is convex, which can be solved by Fast Iterative Soft Thresholding Algorithm (FISTA) [41].
For Equation (6), nonsmooth activations usually lead to closed-form solutions [7], [42]. For example, for ReLU $f_l(z_l) = \max(z_l, 0)$, the solution to Equation (6) is shown as follows:

$$z_l^{k+1} = \begin{cases} \min((W_l^{k+1} p_l^{k+1} + b_l^{k+1} + z_l^k)/2, 0), & z_l^{k+1} \le 0, \\ \max((W_l^{k+1} p_l^{k+1} + b_l^{k+1} + q_l^k + z_l^k)/3, 0), & z_l^{k+1} \ge 0. \end{cases}$$

For smooth activations such as tanh and sigmoid, a lookup-table is recommended [7].

5) Update $\boldsymbol{q}^{k+1}$: The variable $\mathbf{q}^{k+1}$ is updated as follows:

$$q_l^{k+1} \leftarrow \arg\min_{q_l} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}, \mathbf{u}^k) = \arg\min_{q_l} \phi(p_{l+1}^{k+1}, W_{l+1}^{k+1}, b_{l+1}^{k+1}, z_{l+1}^{k+1}, q_l, u_l^k). \tag{8}$$

Equation (8) has a closed-form solution as follows:

$$q_l^{k+1} \leftarrow (\rho p_{l+1}^{k+1} + u_l^k + \nu f_l(z_l^{k+1}))/(\rho + \nu).$$

6) Update $\boldsymbol{u}^{k+1}$: The variable $\mathbf{u}^{k+1}$ is updated as follows:

$$u_l^{k+1} \leftarrow u_l^k + \rho(p_{l+1}^{k+1} - q_l^{k+1}). \tag{9}$$

B. Solutions to Subproblems of the pdADMM-G-Q Algorithm

Obviously, the only difference between the pdADMM-G-Q algorithm and the pdADMM-G algorithm is the p_l-subproblem, which is outlined in the following:

$$p_l^{k+1} \leftarrow \arg\min_{p_l} U_l(p_l; \tau_l^{k+1}) + \mathbb{I}(p_l), \tag{10}$$

where U_l follows Equation (3). The solution to Equation (10) is [34]: $p_l^{k+1} \leftarrow \arg\min_{\delta \in \Delta} \|\delta - (p_l^k - \nabla_{p_l^k} \phi(p_l^k, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k)/\tau_l^{k+1})\|$.

For the pdADMM-G-Q algorithm, the variable $\mathbf{p}$ is only required to be quantized (i.e. $p_l \in \Delta$) when the p_l-subproblem is solved (i.e. Equation (10)), and the variable $\mathbf{q}$ can be any real number when it is updated (i.e. Equation (8)). However, $\mathbf{q}$ is guaranteed to fit into Δ by the linear constraint $p_{l+1} = q_l$. This design is convenient for the convergence analysis, which is detailed in the next section. One variant of the pdADMM-G-Q algorithm is to quantize $\mathbf{p}$ and $\mathbf{q}$ (i.e. $p_l, q_l \in \Delta$) when they are updated. In this case, the solution to Equation (8) is $q_l^{k+1} \leftarrow \arg\min_{\delta \in \Delta} \|\delta - (\rho p_{l+1}^{k+1} + u_l^k + \nu f_l(z_l^{k+1}))/(\rho + \nu)\|$.

C. Convergence Proofs

1) Preliminary Results:

Lemma 4. *It holds for every $k \in \mathbb{N}$ and $l = 1, \cdots, L-1$ that*

$$u_l^k = \nu(q_l^k - f_l(z_l^k)).$$

Proof. This follows directly from the optimality condition of q_l^k and Equation (9). □

Lemma 5. *It holds for every $k \in \mathbb{N}$ and $l = 1, \cdots, L-1$ that*

$$\|u_l^{k+1} - u_l^k\| \le \nu\|q_l^{k+1} - q_l^k\| + \nu S\|z_l^{k+1} - z_l^k\|.$$

Proof.

$$\begin{aligned} &\|u_l^{k+1} - u_l^k\| \\ &= \|\nu(q_l^{k+1} - f_l(z_l^{k+1})) - \nu(q_l^k - f_l(z_l^k))\| \text{(Lemma 4)} \\ &\le \nu\|q_l^{k+1} - q_l^k\| + \nu\|f_l(z_l^{k+1}) - f_l(z_l^k)\| \text{(triangle inequality)} \\ &\le \nu\|q_l^{k+1} - q_l^k\| + \nu S\|z_l^{k+1} - z_l^k\| \text{(Assumption 1).} \end{aligned}$$

□

Lemma 6. *It holds for every $k \in \mathbb{N}$ and $l = 1, \cdots, L-1$ that*

$$\|u_l^{k+1} - u_l^k\|_2^2 \leq 2\nu^2(\|q_l^{k+1} - q_l^k\|_2^2 + S^2\|z_l^{k+1} - z_l^k\|_2^2).$$

Proof.

$$\begin{aligned}\|u_l^{k+1} - u_l^k\|_2^2 &= \nu^2\|q_l^{k+1} - f_l(z_l^{k+1}) - q_l^k + f_l(z_l^k)\|_2^2 \text{(Lemma 4)}\\ &\leq 2\nu^2(\|q_l^{k+1} - q_l^k\|_2^2 + \|f_l(z_l^{k+1}) - f_l(z_l^k)\|_2^2) \text{(mean inequality)}\\ &\leq 2\nu^2(\|q_l^{k+1} - q_l^k\|_2^2 + S^2\|z_l^{k+1} - z_l^k\|_2^2) \text{(Assumption 1).}\end{aligned}$$

□

Lemma 7. *For every $k \in \mathbb{N}$, it holds that*

$$L_\rho(\boldsymbol{p}^k, \boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k) - L_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k) \geq \sum_{l=2}^{L}(\tau_l^{k+1}/2)\|p_l^{k+1} - p_l^k\|_2^2, \tag{11}$$

$$L_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k) - L_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k) \geq \sum_{l=1}^{L}(\theta_l^{k+1}/2)\|W_l^{k+1} - W_l^k\|_2^2, \tag{12}$$

$$L_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k) - L_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k) \geq (\nu/2)\sum_{l=1}^{L}\|b_l^{k+1} - b_l^k\|_2^2, \tag{13}$$

$$L_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k) - L_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{q}^k, \boldsymbol{u}^k) \geq (\nu/2)\sum_{l=1}^{L}\|z_l^{k+1} - z_l^k\|_2^2, \tag{14}$$

$$\beta_\rho(\boldsymbol{p}^k, \boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k) \geq \beta_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k), \tag{15}$$

$$\beta_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^k, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k) - \beta_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k) \geq \sum_{l=1}^{L}(\theta_l^{k+1}/2)\|W_l^{k+1} - W_l^k\|_2^2, \tag{16}$$

$$\beta_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^k, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k) - \beta_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k) \geq (\nu/2)\sum_{l=1}^{L}\|b_l^{k+1} - b_l^k\|_2^2, \tag{17}$$

$$\beta_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^k, \boldsymbol{q}^k, \boldsymbol{u}^k) - \beta_\rho(\boldsymbol{p}^{k+1}, \boldsymbol{W}^{k+1}, \boldsymbol{b}^{k+1}, \boldsymbol{z}^{k+1}, \boldsymbol{q}^k, \boldsymbol{u}^k) \geq (\nu/2)\sum_{l=1}^{L}\|z_l^{k+1} - z_l^k\|_2^2. \tag{18}$$

Proof. Generally, all inequalities can be obtained by applying optimality conditions of updating **p**, **W**, **b** and **z**, respectively. We only prove Inequalities (11), (13), (14) and (15). This is because Inequalities (12) and (16) follow the same routine of Inequality (11), Inequality (17) follows the same routine of Inequality (13), and Inequality (18) follows the same routine of Inequality (14).

Firstly, we focus on Inequality (11). The choice of τ_l^{k+1} requires

$$\phi(p_l^{k+1}, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k) \leq U_l(p_l^{k+1}; \tau_l^{k+1}). \tag{19}$$

Moreover, the optimality condition of Equation (3) leads to

$$\nabla_{p_l^k}\phi(p_l^k, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k) + \tau_l^{k+1}(p_l^{k+1} - p_l^k) = 0. \tag{20}$$

Therefore

$$\begin{aligned}&L_\rho(\mathbf{p}^k, \mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^k, \mathbf{u}^k) - L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^k, \mathbf{u}^k)\\ &= \sum_{l=2}^{L}(\phi(p_l^k, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k) - \phi(p_l^{k+1}, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k))\\ &\geq \sum_{l=2}^{L}(\phi(p_l^k, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k) - U_l(p_l^{k+1}; \tau_l^{k+1})) \text{(Inequality (19))}\\ &= \sum_{l=2}^{L}(-\nabla_{p_l^k}\phi^T(p_l^k, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k)(p_l^{k+1} - p_l^k) - (\tau_l^{k+1}/2)\|p_l^{k+1} - p_l^k\|_2^2)\\ &= \sum_{l=2}^{L}(\tau_l^{k+1}/2)\|p_l^{k+1} - p_l^k\|_2^2 \text{(Equation (20)).}\end{aligned}$$

Next, we prove Inequality (13). Because $\nabla_{b_1}\phi(p_1, W_1, b_1, z_1)$ and $\nabla_{b_l}\phi(p_l, W_l, b_l, z_l, q_l, u_l)$ are Lipschitz continuous with coefficient ν. According to Lemma 2.1 in [41], we have

$$\begin{aligned}\phi(p_1^{k+1}, W_1^{k+1}, b_1^{k+1}, z_1^k) \leq{}& \phi(p_1^{k+1}, W_1^{k+1}, b_1^k, z_1^k) + \nabla_{b_1^k}\phi^T(p_1^{k+1}, W_1^{k+1}, b_1^k, z_1^k)(b_1^{k+1} - b_1^k)\\ &+ (\nu/2)\|b_1^{k+1} - b_1^k\|_2^2, \qquad (21)\\ \phi(p_l^{k+1}, W_l^{k+1}, b_l^{k+1}, z_l^k, q_{l-1}^k, u_{l-1}^k) \leq{}& \phi(p_l^{k+1}, W_l^{k+1}, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k)\\ &+ \nabla_{b_l^k}\phi^T(p_l^{k+1}, W_l^{k+1}, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k)(b_l^{k+1} - b_l^k) + (\nu/2)\|b_l^{k+1} - b_l^k\|_2^2. \qquad (22)\end{aligned}$$

Moreover, the optimality condition of Equation (5) leads to

$$\nabla_{b_1^k}\phi(p_1^k, W_1^k, b_1^k, z_1^k) + \nu(b_1^{k+1} - b_1^k) = 0, \tag{23}$$

$$\nabla_{b_l^k}\phi(p_l^k, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k) + \nu(b_l^{k+1} - b_l^k) = 0. \tag{24}$$

Therefore, we have

$$
\begin{aligned}
&L_\rho(\mathbf{p}^{k+1},\mathbf{W}^{k+1},\mathbf{b}^k,\mathbf{z}^k,\mathbf{q}^k,\mathbf{u}^k)-L_\rho(\mathbf{p}^{k+1},\mathbf{W}^{k+1},\mathbf{b}^{k+1},\mathbf{z}^k,\mathbf{q}^k,\mathbf{u}^k)\\
&=\phi(p_1^{k+1},W_1^{k+1},b_1^k,z_1^k)-\phi(p_1^{k+1},W_1^{k+1},b_1^{k+1},z_1^k)\\
&+\sum_{l=2}^{L}(\phi(p_l^{k+1},W_l^{k+1},b_l^k,z_l^k,q_{l-1}^k,u_{l-1}^k)-\phi(p_l^{k+1},W_l^{k+1},b_l^{k+1},z_l^k,q_{l-1}^k,u_{l-1}^k))\\
&\geq -\nabla_{b_1^k}\phi^T(p_1^{k+1},W_1^{k+1},b_1^k,z_1^k)(b_1^{k+1}-b_1^k)-(\nu/2)\|b_1^{k+1}-b_1^k\|_2^2\\
&+\sum_{l=2}^{L}(-\nabla_{b_l^k}\phi^T(p_l^{k+1},W_l^{k+1},b_l^k,z_l^k,q_{l-1}^k,u_{l-1}^k)(b_l^{k+1}-b_l^k)-(\nu/2)\|b_l^{k+1}-b_l^k\|_2^2)\\
&\text{(Inequalities (21) and (22))}\\
&=(\nu/2)\sum_{l=1}^{L}\|b_l^{k+1}-b_l^k\|_2^2\text{(Equations (23) and (24)).}
\end{aligned}
$$

Then we prove Inequality (14). Because z_l^{k+1} minimizes Equation (6) and Equation (7), we have

$$
\begin{aligned}
&(\nu/2)\|z_l^{k+1}-W_l^{k+1}p_l^{k+1}-b_l^{k+1}\|_2^2+(\nu/2)\|q_l^k-f_l(z_l^{k+1})\|_2^2+(\nu/2)\|z_l^{k+1}-z_l^k\|_2^2\\
&\leq(\nu/2)\|z_l^k-W_l^{k+1}p_l^{k+1}-b_l^{k+1}\|_2^2+(\nu/2)\|q_l^k-f_l(z_l^k)\|_2^2,
\end{aligned}
\tag{25}
$$

and

$$
\begin{aligned}
&R(z_L^k;y)+(\nu/2)\|z_L^k-W_L^{k+1}p_L^{k+1}-b_L^{k+1}\|_2^2-R(z_L^{k+1};y)-(\nu/2)\|z_L^{k+1}-W_L^{k+1}p_L^{k+1}-b_L^{k+1}\|_2^2\\
&=R(z_L^k;y)-R(z_L^{k+1};y)+(\nu/2)\|z_L^k-z_L^{k+1}\|_2^2+\nu(z_L^{k+1}-W_L^{k+1}p_L^{k+1}-b_L^{k+1})^T(z_L^k-z_L^{k+1})\\
&(\|a-b\|_2^2-\|a-c\|_2^2=\|b-c\|_2^2+2(c-a)^T(b-c)\text{ where }a=W_L^{k+1}p_L^{k+1}+b_L^{k+1},\ b=z_L^k,\text{ and }c=z_L^{k+1})\\
&\geq s^T(z_L^k-z_L^{k+1})+(\nu/2)\|z_L^k-z_L^{k+1}\|_2^2+\nu(z_L^{k+1}-W_L^{k+1}p_L^{k+1}-b_L^{k+1})^T(z_L^k-z_L^{k+1})\\
&(s\in\partial R(z_L^{k+1};y)\text{ is a subgradient of }R(z_L^{k+1};y))\\
&=(\nu/2)\|z_L^{k+1}-z_L^k\|_2^2\\
&(0\in s+\nu(z_L^{k+1}-W_L^{k+1}p_L^{k+1}-b_L^{k+1})\text{ by the optimality condition of Equation (7)).}
\end{aligned}
\tag{26}
$$

Therefore

$$
\begin{aligned}
&L_\rho(\mathbf{p}^{k+1},\mathbf{W}^{k+1},\mathbf{b}^{k+1},\mathbf{z}^k,\mathbf{q}^k,\mathbf{u}^k)-L_\rho(\mathbf{p}^{k+1},\mathbf{W}^{k+1},\mathbf{b}^{k+1},\mathbf{z}^{k+1},\mathbf{q}^k,\mathbf{u}^k)\\
&=\sum_{i=1}^{L-1}((\nu/2)\|z_l^k-W_l^{k+1}p_l^{k+1}-b_l^{k+1}\|_2^2+(\nu/2)\|q_l^k-f_l(z_l^k)\|_2^2\\
&-(\nu/2)\|z_l^{k+1}-W_l^{k+1}p_l^{k+1}-b_l^{k+1}\|_2^2-(\nu/2)\|q_l^k-f_l(z_l^{k+1})\|_2^2)\\
&+R(z_L^k;y)+(\nu/2)\|z_L^k-W_L^{k+1}p_L^{k+1}-b_L^{k+1}\|_2^2-R(z_L^{k+1};y)-(\nu/2)\|z_L^{k+1}-W_L^{k+1}p_L^{k+1}-b_L^{k+1}\|_2^2\\
&\geq(\nu/2)\sum_{l=1}^{L}\|z_l^{k+1}-z_l^k\|_2^2\text{(Inequalities (25) and (26)).}
\end{aligned}
$$

Finally Inequality (15) follows directly the optimality condition of $\mathbf{p}^{k+1}$. □

Lemma 8. *For every $k\in N$, it holds that*

$$
\begin{aligned}
&L_\rho(\boldsymbol{p}^{k+1},\boldsymbol{W}^{k+1},\boldsymbol{b}^{k+1},\boldsymbol{z}^{k+1},\boldsymbol{q}^k,\boldsymbol{u}^k)-L_\rho(\boldsymbol{p}^{k+1},\boldsymbol{W}^{k+1},\boldsymbol{b}^{k+1},\boldsymbol{z}^{k+1},\boldsymbol{q}^{k+1},\boldsymbol{u}^{k+1})\\
&\geq\sum_{l=1}^{L-1}((\rho/2-2\nu^2/\rho-\nu/2)\|q_l^{k+1}-q_l^k\|_2^2-(2\nu^2S^2/\rho)\|z_l^{k+1}-z_l^k\|_2^2)
\end{aligned}
\tag{27}
$$

$$
\begin{aligned}
&\beta_\rho(\boldsymbol{p}^{k+1},\boldsymbol{W}^{k+1},\boldsymbol{b}^{k+1},\boldsymbol{z}^{k+1},\boldsymbol{q}^k,\boldsymbol{u}^k)-\beta_\rho(\boldsymbol{p}^{k+1},\boldsymbol{W}^{k+1},\boldsymbol{b}^{k+1},\boldsymbol{z}^{k+1},\boldsymbol{q}^{k+1},\boldsymbol{u}^{k+1})\\
&\geq\sum_{l=1}^{L-1}((\rho/2-2\nu^2/\rho-\nu/2)\|q_l^{k+1}-q_l^k\|_2^2-(2\nu^2S^2/\rho)\|z_l^{k+1}-z_l^k\|_2^2).
\end{aligned}
\tag{28}
$$

Proof. We only prove Inequality (27) because Inequality (28) follows the same routine of Inequality (27).

$$
\begin{aligned}
&L_\rho(\mathbf{p}^{k+1},\mathbf{W}^{k+1},\mathbf{b}^{k+1},\mathbf{z}^{k+1},\mathbf{q}^{k},\mathbf{u}^{k})-L_\rho(\mathbf{p}^{k+1},\mathbf{W}^{k+1},\mathbf{b}^{k+1},\mathbf{z}^{k+1},\mathbf{q}^{k+1},\mathbf{u}^{k+1})\\
&=\sum_{l=1}^{L-1}((\nu/2)\|f_l(z_l^{k+1})-q_l^k\|_2^2-(\nu/2)\|f_l(z_l^{k+1})-q_l^{k+1}\|_2^2-(u_l^{k+1})^T(q_l^k-q_l^{k+1})\\
&+(\rho/2)\|q_l^{k+1}-q_l^k\|_2^2-(1/\rho)\|u_l^{k+1}-u_l^k\|_2^2)\\
&=\sum_{l=1}^{L-1}((\nu/2)\|f_l(z_l^{k+1})-q_l^k\|_2^2-(\nu/2)\|f_l(z_l^{k+1})-q_l^{k+1}\|_2^2-\nu(q_l^{k+1}-f_l(z_l^{k+1}))^T(q_l^k-q_l^{k+1})\\
&+(\rho/2)\|q_l^{k+1}-q_l^k\|_2^2-(1/\rho)\|u_l^{k+1}-u_l^k\|_2^2)\text{(Lemma 4)}\\
&\geq\sum_{l=1}^{L-1}(-(\nu/2)\|q_l^{k+1}-q_l^k\|_2^2+(\rho/2)\|q_l^{k+1}-q_l^k\|_2^2-(1/\rho)\|u_l^{k+1}-u_l^k\|_2^2)\\
&(-\nu(q_l-f_l(z_l^{k+1}))=-(\nu/2)\nabla_{q_l}\|q_l-f_l(z_l^{k+1})\|_2^2\text{ is lipschitz continuous concerning }q_l\text{ and Lemma 2.1 in [41])}\\
&\geq\sum_{l=1}^{L-1}(-(\nu/2)\|q_l^{k+1}-q_l^k\|_2^2+(\rho/2)\|q_l^{k+1}-q_l^k\|_2^2-(2\nu^2/\rho)\|q_l^{k+1}-q_l^k\|_2^2-(2\nu^2S^2/\rho)\|z_l^{k+1}-z_l^k\|_2^2)\\
&\text{(Lemma 6)}\\
&=\sum_{l=1}^{L-1}((\rho/2-2\nu^2/\rho-\nu/2)\|q_l^{k+1}-q_l^k\|_2^2-(2\nu^2S^2/\rho)\|z_l^{k+1}-z_l^k\|_2^2).
\end{aligned}
$$

□

2) Proof of Lemma 1:

Proof. We sum up Inequalities (11), (12), (13), (14), and (27) to obtain Inequality (1), and we sum up Inequalities (15), (16), (17), (18), and (28) to obtain Inequality (2). □

3) Proof of Lemma 2:

Proof. (1) There exists $\mathbf{q}^{'}$ such that $p_{l+1}^k=q_l^{'}$ and

$$F(\mathbf{p}^k,\mathbf{W}^k,\mathbf{b}^k,\mathbf{z}^k,\mathbf{q}^{'})\geq\min_{\mathbf{p},\mathbf{W},\mathbf{b},\mathbf{z},\mathbf{q}}\{F(\mathbf{p},\mathbf{W},\mathbf{b},\mathbf{z},\mathbf{q})|p_{l+1}=q_l\}>-\infty.$$

Therefore, we have

$$
\begin{aligned}
&L_\rho(\mathbf{p}^k,\mathbf{W}^k,\mathbf{b}^k,\mathbf{z}^k,\mathbf{q}^k,\mathbf{u}^k)=F(\mathbf{p}^k,\mathbf{W}^k,\mathbf{b}^k,\mathbf{z}^k,\mathbf{q}^k)+\sum_{l=1}^{L}(u_l^k)^T(p_{l+1}^k-q_l^k)+(\rho/2)\|p_{l+1}^k-q_l^k\|_2^2\\
&=R(z_L^k;y)+(\nu/2)(\sum_{l=1}^{L}\|z_l^k-W_l^kp_l^k-b_l^k\|_2^2+\sum_{l=1}^{L-1}\|q_l^k-f_l(z_l^k)\|_2^2)\\
&+\sum_{l=1}^{L-1}((u_l^k)^T(p_{l+1}^k-q_l^k)+(\rho/2)\|p_{l+1}^k-q_l^k\|_2^2)\\
&=R(z_L^k;y)+(\nu/2)(\sum_{l=1}^{L}\|z_l^k-W_l^kp_l^k-b_l^k\|_2^2+\sum_{l=1}^{L-1}\|q_l^k-f_l(z_l^k)\|_2^2)\\
&+\sum_{l=1}^{L-1}(\nu(q_l^k-f_l(z_l^k))^T(q_l^{'}-q_l^k)+(\rho/2)\|p_{l+1}^k-q_l^k\|_2^2)\\
&(p_{l+1}^k=q_l^{'}\text{ and Lemma 4})\\
&\geq R(z_L^k;y)+(\nu/2)(\sum_{l=1}^{L}\|z_l^k-W_l^kp_l^k-b_l^k\|_2^2+\sum_{l=1}^{L-1}\|q_l^{'}-f_l(z_l^k)\|_2^2)\\
&-\sum_{l=1}^{L-1}(\nu/2)\|q_l^{'}-q_l^k\|_2^2+\sum_{l=1}^{L-1}(\rho/2)\|p_{l+1}^k-q_l^k\|_2^2)\\
&(\nu(q_l-f_l(z_l^{k+1}))=(\nu/2)\nabla_{q_l}\|q_l-f_l(z_l^{k+1})\|_2^2\text{ is lipschitz continuous concerning }q_l\text{ and Lemma 2.1 in [41])}\\
&=F(\mathbf{p}^k,\mathbf{W}^k,\mathbf{b}^k,\mathbf{z}^k,\mathbf{q}^{'})+((\rho-\nu)/2)\|p_{l+1}^k-q_l^k\|_2^2>-\infty.
\end{aligned}
$$

Therefore, $F(\mathbf{p}^k,\mathbf{W}^k,\mathbf{b}^k,\mathbf{z}^k,\mathbf{q}^{'})$ and $((\rho-\nu)/2)\|p_{l+1}^k-q_l^k\|_2^2$ are upper bounded by $L_\rho(\mathbf{p}^k,\mathbf{W}^k,\mathbf{b}^k,\mathbf{z}^k,\mathbf{q}^k,\mathbf{u}^k)$ and hence $L_\rho(\mathbf{p}^0,\mathbf{W}^0,\mathbf{b}^0,\mathbf{z}^0,\mathbf{q}^0,\mathbf{u}^0)$ (Lemma 1). From Assumption 1, $(\mathbf{p}^k,\mathbf{W}^k,\mathbf{b}^k,\mathbf{z}^k)$ is bounded. $\mathbf{q}^k$ is also bounded because $(\rho-\nu)/2\|p_{l+1}^k-q_l^k\|_2^2$ is upper bounded. $\mathbf{u}^k$ is bounded because of Lemma 4.
(2). It follows the same routine as (1). □

4) Proof of Theorem 1:

Proof. (1). From Lemmas 1 and 2, we know that $L_\rho(\mathbf{p}^k, \mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^k, \mathbf{u}^k)$ is convergent because a monotone bounded sequence converges. Moreover, we take the limit on both sides of Inequality (1) to obtain

$$
\begin{aligned}
0 &= \lim_{k\to\infty} L_\rho(\mathbf{p}^k, \mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^k, \mathbf{u}^k) \\
&- \lim_{k\to\infty} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1}) \\
&\geq \lim_{k\to\infty} (\sum\nolimits_{l=2}^{L} (\tau_l^{k+1}/2)\|p_l^{k+1} - p_l^k\|_2^2 \\
&+ \sum\nolimits_{l=1}^{L} (\theta_l^{k+1}/2)\|W_l^{k+1} - W_l^k\|_2^2 + \sum\nolimits_{l=1}^{L} (\nu/2)\|b_l^{k+1} - b_l^k\|_2^2 \\
&+ \sum\nolimits_{l=1}^{L-1} C_1\|z_l^{k+1} - z_l^k\|_2^2 + (\nu/2)\|z_L^{k+1} - z_L^k\|_2^2 \\
&+ \sum\nolimits_{l=1}^{L-1} C_2\|q_l^{k+1} - q_l^k\|_2^2) \geq 0.
\end{aligned}
$$

Because $L_\rho(\mathbf{p}^k, \mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^k, \mathbf{u}^k)$ is convergent, then $\lim_{k\to\infty} \|\mathbf{p}^{k+1} - \mathbf{p}^k\|_2^2 = 0$, $\lim_{k\to\infty} \|\mathbf{W}^{k+1} - \mathbf{W}^k\|_2^2 = 0$, $\lim_{k\to\infty} \|\mathbf{b}^{k+1} - \mathbf{b}^k\|_2^2 = 0$, $\lim_{k\to\infty} \|\mathbf{z}^{k+1} - \mathbf{z}^k\|_2^2 = 0$, and $\lim_{k\to\infty} \|\mathbf{q}^{k+1} - \mathbf{q}^k\|_2^2 = 0$. $\lim_{k\to\infty} \|\mathbf{u}^{k+1} - \mathbf{u}^k\|_2^2 = 0$ is derived from Lemma 6 in Section C in the Appendix.
(2). The proof follows the same procedure as (1). □

5) Proof of Lemma 3:

Proof. (1). We know that $\partial L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1}) = \{\nabla_{\mathrm{p}^{k+1}} L_\rho, \nabla_{\mathrm{W}^{k+1}} L_\rho, \nabla_{\mathrm{b}^{k+1}} L_\rho, \partial_{\mathrm{z}^{k+1}} L_\rho, \nabla_{\mathrm{q}^{k+1}} L_\rho, \nabla_{\mathrm{u}^{k+1}} L_\rho\}$ [7]. Specifically, we prove that $\|g\|$ is upper bounded by the linear combination of $\|\mathbf{p}^{k+1} - \mathbf{p}^k\|$,$\|\mathbf{W}^{k+1} - \mathbf{W}^k\|$, $\|\mathbf{b}^{k+1} - \mathbf{b}^k\|$, $\|\mathbf{z}^{k+1} - \mathbf{z}^k\|$, $\|\mathbf{q}^{k+1} - \mathbf{q}^k\|$, and $\|\mathbf{u}^{k+1} - \mathbf{u}^k\|$.
For p_l^{k+1},

$$
\begin{aligned}
&\nabla_{p_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1}) \\
&= \nabla_{p_l^{k+1}} \phi(p_l^{k+1}, W_l^{k+1}, b_l^{k+1}, z_l^{k+1}, q_{l-1}^{k+1}, u_{l-1}^{k+1}) \\
&= \nabla_{p_l^k} \phi(p_l^k, W_l^k, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k) + \tau_l^{k+1}(p_l^{k+1} - p_l^k) - \tau_l^{k+1}(p_l^{k+1} - p_l^k) \\
&+ \nu(W_l^{k+1})^T W_l^{k+1} p_l^{k+1} - \nu(W_l^k)^T W_l^k p_l^k + \nu(W_l^{k+1})^T b_l^{k+1} - \nu(W_l^k)^T b_l^k - \nu(W_l^{k+1})^T z_l^{k+1} + \nu(W_l^k)^T z_l^k \\
&+ (u_{l-1}^{k+1} - u_{l-1}^k) + \rho(p_l^{k+1} - p_l^k) - \rho(q_{l-1}^{k+1} - q_{l-1}^k) \\
&= -\tau_l^{k+1}(p_l^{k+1} - p_l^k) + \nu(W_l^{k+1})^T W_l^{k+1} p_l^{k+1} - \nu(W_l^k)^T W_l^k p_l^k + \nu(W_l^{k+1})^T b_l^{k+1} - \nu(W_l^k)^T b_l^k \\
&- \nu(W_l^{k+1})^T z_l^{k+1} + \nu(W_l^k)^T z_l^k + (u_{l-1}^{k+1} - u_{l-1}^k) + \rho(p_l^{k+1} - p_l^k) - \rho(q_{l-1}^{k+1} - q_{l-1}^k).
\end{aligned}
$$

So

$$
\begin{aligned}
&\|\nabla_{p_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1})\| \\
&= \|\tau_l^{k+1}(p_l^{k+1} - p_l^k) + \nu(W_l^{k+1})^T W_l^{k+1} p_l^{k+1} - \nu(W_l^k)^T W_l^k p_l^k + \nu(W_l^{k+1})^T b_l^{k+1} - \nu(W_l^k)^T b_l^k \\
&- \nu(W_l^{k+1})^T z_l^{k+1} + \nu(W_l^k)^T z_l^k + (u_{l-1}^{k+1} - u_{l-1}^k) + \rho(p_l^{k+1} - p_l^k) - \rho(q_{l-1}^{k+1} - q_{l-1}^k)\| \\
&\leq \tau_l^{k+1}\|p_l^{k+1} - p_l^k\| + \nu\|(W_l^{k+1})^T W_l^{k+1} p_l^{k+1} - (W_l^k)^T W_l^k p_l^k\| + \nu\|(W_l^{k+1})^T b_l^{k+1} - (W_l^k)^T b_l^k\| \\
&+ \nu\|(W_l^{k+1})^T z_l^{k+1} - (W_l^k)^T z_l^k\| + \|u_{l-1}^{k+1} - u_{l-1}^k\| + \rho\|p_l^{k+1} - p_l^k\| + \rho\|q_{l-1}^{k+1} - q_{l-1}^k\| \text{ (triangle inequality)} \\
&= \tau_l^{k+1}\|p_l^{k+1} - p_l^k\| + \nu\|(W_l^{k+1})^T W_l^{k+1}(p_l^{k+1} - p_l^k) + (W_l^{k+1})^T(W_l^{k+1} - W_l^k) p_l^k + (W_l^{k+1} - W_l^k)^T W_l^k p_l^k\| \\
&+ \nu\|(W_l^{k+1})^T(b_l^{k+1} - b_l^k) + (W_l^{k+1} - W_l^k)^T b_l^k\| + \nu\|(W_l^{k+1})^T(z_l^{k+1} - z_l^k) + (W_l^{k+1} - W_l^k)^T z_l^k\| \\
&+ \|u_{l-1}^{k+1} - u_{l-1}^k\| + \rho\|p_l^{k+1} - p_l^k\| + \rho\|q_{l-1}^{k+1} - q_{l-1}^k\| \\
&\leq \tau_l^{k+1}\|p_l^{k+1} - p_l^k\| + \nu\|W_l^{k+1}\|^2\|p_l^{k+1} - p_l^k\| + \nu\|W_l^{k+1}\|\|W_l^{k+1} - W_l^k\|\|p_l^k\| + \nu\|W_l^{k+1} - W_l^k\|\|W_l^k\|\|p_l^k\| \\
&+ \nu\|W_l^{k+1}\|\|b_l^{k+1} - b_l^k\| + \nu\|W_l^{k+1} - W_l^k\|\|b_l^k\| + \nu\|W_l^{k+1}\|\|z_l^{k+1} - z_l^k\| + \nu\|W_l^{k+1} - W_l^k\|\|z_l^k\| \\
&+ \nu(\|q_{l-1}^{k+1} - q_{l-1}^k\| + S\|z_{l-1}^{k+1} - z_{l-1}^k\|) + \rho\|p_l^{k+1} - p_l^k\| + \rho\|q_{l-1}^{k+1} - q_{l-1}^k\| \\
&\text{(triangle inequality, Cauthy-Schwartz inequality and Lemma 5)} \\
&\leq \tau_l^{k+1}\|p_l^{k+1} - p_l^k\| + \nu\mathbb{N}_{\mathbf{W}}^2\|p_l^{k+1} - p_l^k\| + 2\nu\mathbb{N}_{\mathbf{W}}\mathbb{N}_{\mathbf{p}}\|W_l^{k+1} - W_l^k\| + \nu\mathbb{N}_{\mathbf{W}}\|b_l^{k+1} - b_l^k\| + \nu\mathbb{N}_{\mathbf{b}}\|W_l^{k+1} - W_l^k\| \\
&+ \nu\mathbb{N}_{\mathbf{W}}\|z_l^{k+1} - z_l^k\| + \nu\mathbb{N}_{\mathbf{z}}\|W_l^{k+1} - W_l^k\| + 2\nu^2(\|q_{l-1}^{k+1} - q_{l-1}^k\|_2^2 + S^2\|z_{l-1}^{k+1} - z_{l-1}^k\|_2^2) + \rho\|p_l^{k+1} - p_l^k\| + \rho\|q_{l-1}^{k+1} - q_{l-1}^k\| \\
&\text{(Lemma 2).}
\end{aligned}
$$

For W_1^{k+1},

$$\begin{aligned}
&\nabla_{W_1^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1}) \\
&= \nabla_{W_1^{k+1}} \phi(p_1^{k+1}, W_1^{k+1}, b_1^{k+1}, z_1^{k+1}) \\
&= \nabla_{W_1^k} \phi(p_1^{k+1}, W_1^k, b_1^k, z_1^k) + \theta_1^{k+1}(W_1^{k+1} - W_1^k) + \nu(W_1^{k+1} - W_1^k) p_1^{k+1}(p_1^{k+1})^T + \nu(b_1^{k+1} - b_1^k)(p_1^{k+1})^T \\
&- \nu(z_1^{k+1} - z_1^k)(p_1^{k+1})^T - \theta_1^{k+1}(W_1^{k+1} - W_1^k) \\
&= \nu(W_1^{k+1} - W_1^k) p_1^{k+1}(p_1^{k+1})^T + \nu(b_1^{k+1} - b_1^k)(p_1^{k+1})^T - \nu(z_1^{k+1} - z_1^k)(p_1^{k+1})^T - \theta_1^{k+1}(W_1^{k+1} - W_1^k)
\end{aligned}$$

(The optimality condition of Equation (4)).

So

$$\begin{aligned}
&\|\nabla_{W_1^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1})\| \\
&= \|\nu(W_1^{k+1} - W_1^k) p_1^{k+1}(p_1^{k+1})^T + \nu(b_1^{k+1} - b_1^k)(p_1^{k+1})^T - \nu(z_1^{k+1} - z_1^k)(p_1^{k+1})^T - \theta_1^{k+1}(W_1^{k+1} - W_1^k)\| \\
&\le \nu\|W_1^{k+1} - W_1^k\|\|p_1^{k+1}\|^2 + \nu\|b_1^{k+1} - b_1^k\|\|p_1^{k+1}\| + \nu\|z_1^{k+1} - z_1^k\|\|p_1^{k+1}\| + \theta_1^{k+1}\|W_1^{k+1} - W_1^k\|
\end{aligned}$$

(triangle inequality and Cauthy-Schwartz inequality)

$$\le \nu\|W_1^{k+1} - W_1^k\|_{\mathbb{N}_\mathbf{p}}^2 + \nu\|b_1^{k+1} - b_1^k\|_{\mathbb{N}_\mathbf{p}} + \nu\|z_1^{k+1} - z_1^k\|_{\mathbb{N}_\mathbf{p}} + \theta_1^{k+1}\|W_1^{k+1} - W_1^k\| \text{ (Theorem 1).}$$

For $W_l^{k+1} (1 < l \le L)$,

$$\begin{aligned}
&\nabla_{W_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1}) \\
&= \nabla_{W_l^{k+1}} \phi(p_l^{k+1}, W_l^{k+1}, b_l^{k+1}, z_l^{k+1}, p_{l-1}^{k+1}, u_{l-1}^{k+1}) \\
&= \nabla_{W_l^k} \phi(p_l^{k+1}, W_l^k, b_l^k, z_l^k, p_{l-1}^k, u_{l-1}^k) + \theta_l^{k+1}(W_l^{k+1} - W_1^k) + \nu(W_l^{k+1} - W_l^k) p_l^{k+1}(p_l^{k+1})^T \\
&+ \nu(b_l^{k+1} - b_l^k)(p_l^{k+1})^T - \nu(z_l^{k+1} - z_l^k)(p_l^{k+1})^T - \theta_l^{k+1}(W_l^{k+1} - W_l^k) \\
&= \nu(W_l^{k+1} - W_l^k) p_l^{k+1}(p_l^{k+1})^T + \nu(b_l^{k+1} - b_l^k)(p_l^{k+1})^T - \nu(z_l^{k+1} - z_l^k)(p_l^{k+1})^T - \theta_l^{k+1}(W_l^{k+1} - W_l^k)
\end{aligned}$$

(The optimality condition of Equation (4)).

So

$$\begin{aligned}
&\|\nabla_{W_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1})\| \\
&= \|\nu(W_l^{k+1} - W_l^k) p_l^{k+1}(p_l^{k+1})^T + \nu(b_l^{k+1} - b_l^k)(p_l^{k+1})^T - \nu(z_l^{k+1} - z_l^k)(p_l^{k+1})^T - \theta_l^{k+1}(W_l^{k+1} - W_l^k)\| \\
&\le \nu\|W_l^{k+1} - W_l^k\|\|p_l^{k+1}\|^2 + \nu\|b_l^{k+1} - b_l^k\|\|p_l^{k+1}\| + \nu\|z_l^{k+1} - z_l^k\|\|p_l^{k+1}\| + \theta_l^{k+1}\|W_l^{k+1} - W_l^k\|
\end{aligned}$$

(triangle inequality and Cauthy-Schwartz inequality)

$$\le \nu\|W_l^{k+1} - W_l^k\|_{\mathbb{N}_\mathbf{p}}^2 + \nu\|b_l^{k+1} - b_l^k\|_{\mathbb{N}_\mathbf{p}} + \nu\|z_l^{k+1} - z_l^k\|_{\mathbb{N}_\mathbf{p}} + \theta_l^{k+1}\|W_l^{k+1} - W_l^k\| \text{ (Theorem 1).}$$

For b_1^{k+1},

$$\begin{aligned}
&\nabla_{b_1^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1}) \\
&= \nabla_{b_1^{k+1}} \phi(p_1^{k+1}, W_1^{k+1}, b_1^{k+1}, z_1^{k+1}) \\
&= \nabla_{b_1^k} \phi(p_1^{k+1}, W_1^{k+1}, b_1^k, z_1^k) + \nu(b_1^{k+1} - b_1^k) + \nu(z_1^k - z_1^{k+1}) \\
&= \nu(z_1^k - z_1^{k+1}) \text{ (The optimality condition of Equation (5)).}
\end{aligned}$$

So $\|\nabla_{b_1^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1})\| = \nu\|z_1^{k+1} - z_1^k\|$.
For $b_l^{k+1} (1 < l \le L)$,

$$\begin{aligned}
&\nabla_{b_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1}) \\
&= \nabla_{b_l^{k+1}} \phi(p_l^{k+1}, W_l^{k+1}, b_l^{k+1}, z_l^{k+1}, q_{l-1}^k, u_{l-1}^k) \\
&= \nabla_{b_l^k} \phi(p_l^{k+1}, W_l^{k+1}, b_l^k, z_l^k, q_{l-1}^k, u_{l-1}^k) + \nu(b_l^{k+1} - b_l^k) + \nu(z_l^k - z_l^{k+1}) \\
&= \nu(z_l^k - z_l^{k+1}) \text{ (The optimality condition of Equation (5)).}
\end{aligned}$$

So $\|\nabla_{b_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1})\| = \nu\|z_l^{k+1} - z_l^k\|$.
For $z_l^{k+1} (l < L)$,

$$\begin{aligned}
&\partial_{z_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1}) \\
&= \partial_{z_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k}, \mathbf{u}^{k}) + \nu(z_l^{k+1} - z_l^k) - \nu(z_l^{k+1} - z_l^k) - \nu\partial f_l(z_l^{k+1}) \circ (q_l^{k+1} - q_l^k) (\circ \text{ is Hadamard product}) \\
&= -\nu(z_l^{k+1} - z_l^k) - \nu\partial f_l(z_l^{k+1}) \circ (q_l^{k+1} - q_l^k)\ (0 \in \partial_{z_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k}, \mathbf{u}^{k}) + \nu(z_l^{k+1} - z_l^k)).
\end{aligned}$$

So

$$\begin{aligned}
&\|\partial_{z_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1})\| \\
&= \| - \nu(z_l^{k+1} - z_l^k) - \nu\partial f_l(z_l^{k+1}) \circ (q_l^{k+1} - q_l^k)\| \\
&\leq \nu\|z_l^{k+1} - z_l^k\| + \nu\|\partial f_l(z_l^{k+1})\|\|q_l^{k+1} - q_l^k\| (\text{Cauchy-Schwartz inequality and triangle inequality}) \\
&\leq \nu\|z_l^{k+1} - z_l^k\| + \nu M\|q_l^{k+1} - q_l^k\| (\|\partial f_l(z_l^{k+1})\| \leq M).
\end{aligned}$$

For z_L^{k+1}, $\partial_{z_L^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1}) = 0$ by the optimality condition of Equation (7).
For q_l^{k+1},

$$\begin{aligned}
&\nabla_{q_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1}) \\
&= \nabla_{q_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k}) + u_l^{k+1} - u_l^k \\
&= u_l^{k+1} - u_l^k\ (\nabla_{q_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k}) = 0 \text{ by the optimality condition of Equation (8)}).
\end{aligned}$$

So $\|\nabla_{q_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1})\| = \|u_l^{k+1} - u_l^k\|$.
For u_l^{k+1},

$$\nabla_{u_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1}) = (p_{l+1}^{k+1} - q_l^{k+1}) = (u_l^{k+1} - u_l^k)/\rho.$$

So $\|\nabla_{u_l^{k+1}} L_\rho(\mathbf{p}^{k+1}, \mathbf{W}^{k+1}, \mathbf{b}^{k+1}, \mathbf{z}^{k+1}, \mathbf{q}^{k+1}, \mathbf{u}^{k+1})\| = \|u_l^{k+1} - u_l^k\|/\rho$.
In summary, we prove that $\nabla_{\mathbf{p}^{k+1}} L_\rho, \nabla_{\mathbf{W}^{k+1}} L_\rho, \nabla_{\mathbf{b}^{k+1}} L_\rho, \partial_{\mathbf{z}^{k+1}} L_\rho, \nabla_{\mathbf{q}^{k+1}} L_\rho, \nabla_{\mathbf{u}^{k+1}} L_\rho$ are upper bounded by the linear combination of $\|\mathbf{p}^{k+1} - \mathbf{p}^k\|$,$\|\mathbf{W}^{k+1} - \mathbf{W}^k\|$, $\|\mathbf{b}^{k+1} - \mathbf{b}^k\|$, $\|\mathbf{z}^{k+1} - \mathbf{z}^k\|$, $\|\mathbf{q}^{k+1} - \mathbf{q}^k\|$, and $\|\mathbf{u}^{k+1} - \mathbf{u}^k\|$.
(2). It follows exactly the proof of (1) except for p_l^{k+1} . □

6) The proof of Theorem 2:

Proof. From Lemma 2(1), $(\mathbf{p}^k, \mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^k, \mathbf{u}^k)$ has at least a limit point $(\mathbf{p}^*, \mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{q}^*, \mathbf{u}^*)$ because a bounded sequence has at least a limit point. From Lemma 3 and Theorem 1, $\|g^{k+1}\| \to 0$ as $k \to \infty$. According to the definition of general subgradient (Definition 8.3 in [35]), we have $0 \in \partial L_\rho(\mathbf{p}^*, \mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{q}^*, \mathbf{u}^*)$. In other words, every limit point $(\mathbf{p}^*, \mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{q}^*, \mathbf{u}^*)$ is a stationary point. □

7) The proof of Theorem 3:

Proof. From Lemma 2(2), $(\mathbf{W}^k, \mathbf{b}^k, \mathbf{z}^k, \mathbf{q}^k, \mathbf{u}^k)$ has at least a limit point $(\mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{q}^*, \mathbf{u}^*)$ because a bounded sequence has at least a limit point. $\mathbf{p}^k$ has at least a limit point $\mathbf{p}^*$ because $\mathbf{p}^k \in \Delta$ and Δ is finite. From Lemma 3(2) and Theorem 1, $\|\overline{g}_{\mathbf{W}}^{k+1}\| \to 0$, $\|\overline{g}_{\mathbf{b}}^{k+1}\| \to 0$, $\|\overline{g}_{\mathbf{z}}^{k+1}\| \to 0$, $\|\overline{g}_{\mathbf{q}}^{k+1}\| \to 0$, $\|\overline{g}_{\mathbf{u}}^{k+1}\| \to 0$ as $k \to \infty$. According to the definition of general subgradient (Defintion 8.3 in [35]), we have $\nabla_{\mathbf{W}^*} \beta_\rho(\mathbf{p}^*, \mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{q}^*, \mathbf{u}^*) = 0$, $\nabla_{\mathbf{b}^*} \beta_\rho(\mathbf{p}^*, \mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{q}^*, \mathbf{u}^*) = 0$, $0 \in \partial_{\mathbf{z}^*} \beta_\rho(\mathbf{p}^*, \mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{q}^*, \mathbf{u}^*)$, $\nabla_{\mathbf{q}^*} \beta_\rho(\mathbf{p}^*, \mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{q}^*, \mathbf{u}^*) = 0$ and $\nabla_{\mathbf{u}^*} \beta_\rho(\mathbf{p}^*, \mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{q}^*, \mathbf{u}^*) = 0$ (i.e. $p_{l+1}^* = q_l^*$). In other words, every limit point $(\mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{u}^*)$ is a stationary point of Problem 3. Moreover, τ_l^k has a limit point τ_l^* because it is bounded. Let $\boldsymbol{\tau}^k = \{\tau_l^k\}_{l=2}^L$. Consider a subsequence $(\mathbf{p}^s, \mathbf{W}^s, \mathbf{b}^s, \mathbf{z}^s, \mathbf{q}^s, \mathbf{u}^s, \boldsymbol{\tau}^{s+1}) \to (\mathbf{p}^*, \mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{q}^*, \mathbf{u}^*, \boldsymbol{\tau}^*)$. Because $u_l^{s+1} = u_l^s + \rho(p_{l+1}^s - q_l^s)$ and $u_l^{s+1} \to u_l^s$, thus $p_{l+1}^s \to q_l^s$, and $p_{l+1}^{s+1} \to q_l^{s+1}$. Because $q_l^{s+1} \to q_l^s$, then $p_{l+1}^{s+1} \to p_{l+1}^s$ for any l. In other words, $\mathbf{p}^{s+1} \to \mathbf{p}^s$. Because $\mathbf{p}^s \to \mathbf{p}^*$, then $\mathbf{p}^{s+1} \to \mathbf{p}^*$. The optimality condition of $\mathbf{p}^{s+1}$ (i.e. Equation (10)) leads to

$$p_l^{s+1} \leftarrow \arg\min_{\delta \in \Delta} \|\delta - p_l^s - \nabla_{p_l^s} \phi(p_l^s, W_l^s, b_l^s, z_l^s, q_{l-1}^s, u_{l-1}^s)/\tau_l^{s+1})\|.$$

Taking $s \to \infty$ on both sides, we have

$$p_l^* \leftarrow \arg\min_{\delta \in \Delta} \|\delta - (p_l^* - \nabla_{p_l^*} \phi(p_l^*, W_l^*, b_l^*, z_l^*, q_{l-1}^*, u_{l-1}^*)/\tau_l^*)\|.$$

Because $\nabla_{p_l^*} F(\mathbf{p}^*, \mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{q}^*) = \nu W_l^T(z_l^* - W_l^* p_l^* - b_l^*) = \nabla_{p_l^*} \phi(p_l^*, W_l^*, b_l^*, z_l^*, q_{l-1}^*, u_{l-1}^*)$. Then

$$p_l^* \leftarrow \arg\min_{\delta \in \Delta} \|\delta - (p_l^* - \nabla_{p_l^*} F(\mathbf{p}^*, \mathbf{W}^*, \mathbf{b}^*, \mathbf{z}^*, \mathbf{q}^*)/\tau_l^*)\|.$$

Namely, $\mathbf{p}^*$ is a quantized stationary point of Problem 3. □

8) The proof of Theorem 4:

Proof. (1). To prove this, we will first show that c_k satisfies two conditions: (1). $c_k \geq c_{k+1}$. (2). $\sum_{k=0}^{\infty} c_k$ is bounded. We then conclude the convergence rate of $o(1/k)$ based on these two conditions. Specifically, first, we have

$$\begin{aligned}
c_k =& \min_{0\leq i\leq k}(\sum_{l=2}^{L}(\tau_l^{i+1}/2)\|p_l^{i+1}-p_l^i\|_2^2+\sum_{l=1}^{L}(\theta_l^{i+1}/2)\|W_l^{i+1}-W_l^i\|_2^2+\sum_{l=1}^{L}(\nu/2)\|b_l^{i+1}-b_l^i\|_2^2\\
&+\sum_{l=1}^{L-1}C_1\|z_l^{i+1}-z_l^i\|_2^2+(\nu/2)\|z_L^{i+1}-z_L^i\|_2^2+\sum_{l=1}^{L-1}C_2\|q_l^{i+1}-q_l^i\|_2^2)\\
\geq& \min_{0\leq i\leq k+1}(\sum_{l=2}^{L}(\tau_l^{i+1}/2)\|p_l^{i+1}-p_l^i\|_2^2+\sum_{l=1}^{L}(\theta_l^{i+1}/2)\|W_l^{i+1}-W_l^i\|_2^2+\sum_{l=1}^{L}(\nu/2)\|b_l^{i+1}-b_l^i\|_2^2\\
&+\sum_{l=1}^{L-1}C_1\|z_l^{i+1}-z_l^i\|_2^2+(\nu/2)\|z_L^{i+1}-z_L^i\|_2^2+\sum_{l=1}^{L-1}C_2\|q_l^{i+1}-q_l^i\|_2^2)\\
=& c_{k+1}.
\end{aligned}$$

Therefore c_k satisfies the first condition. Second,

$$\begin{aligned}
&\sum_{k=0}^{\infty}c_k\\
&=\sum_{k=0}^{\infty}\min_{0\leq i\leq k}(\sum_{l=2}^{L}(\tau_l^{i+1}/2)\|p_l^{i+1}-p_l^i\|_2^2+\sum_{l=1}^{L}(\theta_l^{i+1}/2)\|W_l^{i+1}-W_l^i\|_2^2+\sum_{l=1}^{L}(\nu/2)\|b_l^{i+1}-b_l^i\|_2^2\\
&+\sum_{l=1}^{L-1}C_1\|z_l^{i+1}-z_l^i\|_2^2+(\nu/2)\|z_L^{i+1}-z_L^i\|_2^2+\sum_{l=1}^{L-1}C_2\|q_l^{i+1}-q_l^i\|_2^2)\\
&\leq\sum_{k=0}^{\infty}(\sum_{l=2}^{L}(\tau_l^{k+1}/2)\|p_l^{k+1}-p_l^k\|_2^2+\sum_{l=1}^{L}(\theta_l^{k+1}/2)\|W_l^{k+1}-W_l^k\|_2^2+\sum_{l=1}^{L}(\nu/2)\|b_l^{k+1}-b_l^k\|_2^2\\
&+\sum_{l=1}^{L-1}C_1\|z_l^{k+1}-z_l^k\|_2^2+(\nu/2)\|z_L^{k+1}-z_L^k\|_2^2+\sum_{l=1}^{L-1}C_2\|q_l^{k+1}-q_l^k\|_2^2)\\
&\leq L_\rho(\mathbf{p}^0,\mathbf{W}^0,\mathbf{b}^0,\mathbf{z}^0,\mathbf{q}^0,\mathbf{u}^0)-L_\rho(\mathbf{p}^*,\mathbf{W}^*,\mathbf{b}^*,\mathbf{z}^*,\mathbf{q}^*,\mathbf{u}^*)\text{(Lemma 1).}
\end{aligned}$$

So $\sum_{k=0}^{\infty}c_k$ is bounded and c_k satisfies the second condition. Finally, it has been proved that the sufficient conditions of convergence rate $o(1/k)$ are: (1) $c_k \geq c_{k+1}$, and (2) $\sum_{k=0}^{\infty}c_k$ is bounded, and (3) $c_k \geq 0$ (Lemma 1.2 in [43]). Since we have proved the first two conditions and the third one $c_k \geq 0$ is obvious, the convergence rate of $o(1/k)$ is proven.
(2). It follows the same procedure as (1). □

D. More Experimental Results

1) Datasets Details: 1. Cora [44]. The Cora dataset consists of 2708 scientific publications classified into one of seven classes. The citation network consists of 5429 links. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 1433 unique words.
2. PubMed [44]. PubMed comprises 30M+ citations for biomedical literature that have been collected from sources such as MEDLINE, life science journals, and published online e-books. It also includes links to text content from PubMed Central and other publishers' websites.
3. Citeseer [44]. The Citeseer dataset was collected from the Tagged.com social network website. It contains 5.6 million users and 858 million links between them. Each user has 4 features and is manually labeled as "spammer" or "not spammer". Each link represents an action between two users and includes a timestamp and a type. The network contains 7 anonymized types of links. The original task on the dataset is to identify (i.e., classify) the spammer users based on their relational and non-relational features.
4. Amazon Computers and Amazon Photo [45]. Amazon Computers and Amazon Photo are segments of the Amazon co-purchase graph, where nodes represent goods, edges indicate that two goods are frequently bought together, node features are bag-of-words encoded product reviews, and class labels are given by the product category.
5. Coauthor CS and Coauthor Physics [46]. Coauthor CS and Coauthor Physics are co-authorship graphs based on the Microsoft Academic Graph from the KDD Cup 2016 challenge 3. Here, nodes are authors, that are connected by an edge if they co-authored a paper; node features represent paper keywords for each author's papers, and class labels indicate the most active fields of study for each author.
6. Flickr [47]. In Flickr, one node in the graph represents one image uploaded to Flickr. If two images share some common properties (e.g., same geographic location, same gallery, comments by the same user, etc.), there is an edge between the nodes of these two images. Node features are bag-of-word representation of the images and labels are classes of images.
7. Ogbn-Arxiv [48]. The Ogbn-Arxiv dataset is a directed graph, representing the citation network between all Computer Science (CS) ARXIV papers indexed by MAG. Each node is an ARXIV paper and each directed edge indicates that one paper cites another one. Each paper comes with a 128-dimensional feature vector obtained by averaging the embeddings of words in its title and abstract. In addition, all papers are also associated with the year that the the corresponding paper was published.

2) The Settings of All Hyperparameters: This section provides more details on the hyperparameter settings of all datasets, which are shown in the following tables.

Dataset	Cora	PubMed	Citeseer
Learning Rate(GD)	10^{-1}	5×10^{-2}	10^{-1}
Learning Rate(Adadelta)	10^{-3}	10^{-3}	10^{-3}
Learning Rate(Adagrad)	10^{-3}	10^{-3}	10^{-3}
Learning Rate(Adam)	10^{-4}	10^{-4}	10^{-3}
ρ, ν(pdADMM-G)	10^{-4}	10^{-4}	10^{-4}
ρ, ν(pdADMM-G-Q)	10^{-4}	10^{-3}	10^{-3}
Dataset	Amazon Computers	Amazon Photo	Coauthor CS
Learning Rate(GD)	10^{-2}	10^{-2}	10^{-1}
Learning Rate(Adadelta)	10^{-3}	10^{-3}	10^{-3}
Learning Rate(Adagrad)	10^{-3}	10^{-3}	10^{-3}
Learning Rate(Adam)	10^{-3}	10^{-3}	10^{-3}
ρ, ν(pdADMM-G)	10^{-3}	10^{-3}	10^{-2}
ρ, ν(pdADMM-G-Q)	10^{-3}	10^{-3}	10^{-2}
Dataset	Coauthor Physics	Flickr	Ogbn-Arxiv
Learning Rate(GD)	10^{-1}	10^{-3}	10^{-2}
Learning Rate(Adadelta)	10^{-3}	10^{-2}	10^{-1}
Learning Rate(Adagrad)	10^{-3}	10^{-3}	10^{-3}
Learning Rate(Adam)	10^{-3}	10^{-3}	10^{-3}
ρ, ν(pdADMM-G)	10^{-2}	10^{-4}	10^{-4}
ρ, ν(pdADMM-G-Q)	10^{-2}	10^{-4}	10^{-4}

TABLE V: Hyperparameter settings of all methods on nine benchmark datasets when the number of neurons is 100.

Dataset	Cora	PubMed	Citeseer
Learning Rate(GD)	10^{-1}	5×10^{-3}	10^{-1}
Learning Rate(Adadelta)	10^{-3}	10^{-4}	10^{-3}
Learning Rate(Adagrad)	10^{-3}	10^{-3}	10^{-3}
Learning Rate(Adam)	10^{-4}	10^{-4}	10^{-4}
ρ, ν(pdADMM-G)	10^{-4}	10^{-4}	10^{-3}
ρ, ν(pdADMM-G-Q)	10^{-4}	10^{-3}	10^{-3}
Dataset	Amazon Computers	Amazon Photo	Coauthor CS
Learning Rate(GD)	10^{-2}	10^{-2}	10^{-1}
Learning Rate(Adadelta)	10^{-3}	10^{-3}	10^{-3}
Learning Rate(Adagrad)	10^{-3}	10^{-3}	10^{-3}
Learning Rate(Adam)	10^{-4}	10^{-4}	10^{-4}
ρ, ν(pdADMM-G)	10^{-3}	10^{-3}	10^{-3}
ρ, ν(pdADMM-G-Q)	10^{-3}	10^{-3}	10^{-3}
Dataset	Coauthor Physics	Flickr	Ogbn-Arxiv
Learning Rate(GD)	10^{-2}	10^{-2}	10^{-2}
Learning Rate(Adadelta)	10^{-3}	10^{-2}	10^{-1}
Learning Rate(Adagrad)	10^{-3}	10^{-3}	10^{-3}
Learning Rate(Adam)	10^{-4}	10^{-3}	10^{-3}
ρ, ν(pdADMM-G)	10^{-2}	10^{-4}	10^{-4}
ρ, ν(pdADMM-G-Q)	10^{-2}	10^{-4}	10^{-4}

TABLE VI: Hyperparameter settings of all methods on nine benchmark datasets when the number of neurons is 500.

3) The Performance of Validation Sets: This section provides more experimental results on the validation sets of all datasets, which are shown in the following tables.

Dataset	Cora	PubMed	Citeseer
GD	0.704±0.037	0.626± 0.072	0.619±0.045
Adadelta	0.652±0.064	0.720±0.035	0.620±0.022
Adagrad	0.720± 0.022	0.762± 0.012	0.604 ±0.027
Adam	0.720±0.034	0.745± 0.014	0.624±0.014
pdADMM-G	0.750±0.005	0.788±0.004	**0.724±0.005**
pdADMM-G-Q	**0.754± 0.002**	**0.793±0.002**	0.722±0.002
Dataset	Amazon Computers	Amazon Photo	Coauthor CS
GD	0.654±0.033	0.730±0.165	0.875±0.007
Adadelta	0.136±0.062	0.343±0.046	0.781±0.084
Adagrad	0.750±0.095	0.808±0.018	0.889±0.006
Adam	0.740±0.010	**0.850±0.006**	0.887±0.009
pdADMM-G	**0.753±0.005**	0.846±0.014	0.913±0.003
pdADMM-G-Q	0.688±0.063	0.822±0.013	**0.916±0.003**
Dataset	Coauthor Physics	Flickr	Ogbn-Arxiv
GD	0.921±0.009	0.464±0.008	0.378±0.004
Adadelta	0.918±0.014	0.461±0.006	0.514±0.014
Adagrad	0.928±0.005	0.480±0.003	0.574±0.008
Adam	0.919± 0.010	0.512±0.004	**0.681±0.003**
pdADMM-G	0.933±0.001	**0.514±0.001**	0.649±0.012
pdADMM-G-Q	**0.935±0.002**	0.506±0.004	0.661±0.004

TABLE VII: The validation performance of all methods when the number of neurons is 100.

Dataset	Cora	PubMed	Citeseer
GD	0.731±0.018	0.651±0.034	0.679±0.008
Adadelta	0.716±0.061	0.688±0.024	0.597±0.025
Adagrad	0.765±0.014	0.776±0.006	0.668±0.028
Adam	**0.758±0.013**	0.778±0.008	0.668±0.020
pdADMM-G	0.753±0.004	**0.792±0.004**	0.729±0.003
pdADMM-G-Q	0.757±0.005	**0.792±0.003**	**0.730±0.004**
Dataset	Amazon Computers	Amazon Photo	Coauthor CS
GD	0.727± 0.012	0.809±0.012	0.897±0.003
Adadelta	0.246±0.073	0.371±0.075	0.884±0.003
Adagrad	0.766±0.011	0.860±0.003	**0.912±0.004**
Adam	0.750±0.017	**0.872±0.020**	0.893±0.013
pdADMM-G	**0.778±0.007**	0.861±0.005	**0.912±0.003**
pdADMM-G-Q	0.764±0.008	0.850±0.009	0.910±0.003
Dataset	Coauthor Physics	Flickr	Ogbn-Arxiv
GD	0.928±0.001	0.466±0.001	0.451±0.033
Adadelta	0.932±0.006	0.462±0.004	0.591±0.017
Adagrad	**0.935±0.005**	0.488±0.007	0.646±0.010
Adam	0.933±0.007	**0.516±0.002**	**0.692±0.008**
pdADMM-G	0.932±0.001	0.514±0.003	0.661±0.005
pdADMM-G-Q	0.933±0.002	0.514±0.001	0.667±0.003

TABLE VIII: The validation performance of all methods when the number of neurons is 500.