LINEARLY-CONSTRAINED NONSMOOTH OPTIMIZATION FOR TRAINING AUTOENCODERS*

WEI LIU†, XIN LIU†, AND XIAOJUN CHEN‡

Abstract. A regularized minimization model with l_1-norm penalty (RP) is introduced for training the autoencoders that belong to a class of two-layer neural networks. We show that the RP can act as an exact penalty model which shares the same global minimizers, local minimizers, and d(irectional)-stationary points with the original regularized model under mild conditions. We construct a bounded box region that contains at least one global minimizer of the RP, and propose a linearly constrained regularized minimization model with l_1-norm penalty (LRP) for training autoencoders. A smoothing proximal gradient algorithm is designed to solve the LRP. Convergence of the algorithm to a generalized d-stationary point of the RP and LRP is delivered. Comprehensive numerical experiments convincingly illustrate the efficiency as well as the robustness of the proposed algorithm.

Key words. autoencoders, neural network, penalty method, smoothing approximation, finite-sum optimization.

AMS subject classifications. 90C26, 90C30

1. Introduction.

A deep neural network (DNN) [28] aims to solve a finite-sum minimization problem

$$\min_{W_\ell, b_\ell, \ell=1\ldots,L} \frac{1}{N}\sum_{n=1}^{N} \psi_n(\varphi_{n,L}(W_1,\ldots,W_L,b_1,\ldots,b_L)). \tag{1.1}$$

Here $\varphi_{n,L}(W_1,\ldots,W_L,b_1,\ldots,b_L) = \sigma_L(W_L\sigma_{L-1}(\cdots\sigma_1(W_1x_n+b_1)+\cdots)+b_L)$ denotes the outputs of the L-th hidden layer, and ψ_n denotes the loss function measuring the output $\varphi_{n,L}(W_1,\ldots,W_L,b_1,\ldots,b_L)$ and its corresponding true output for $n = 1,\ldots,N$, where $\{x_n\}_{n=1}^N$ is the data set, $W_\ell \in \mathbb{R}^{N_\ell\times N_{\ell-1}}$, $b_\ell \in \mathbb{R}^{N_\ell}$, $\sigma_\ell : \mathbb{R}^{N_\ell} \mapsto \mathbb{R}^{N_\ell}$ ($\ell = 1,\ldots,L$) are the weight matrices, the bias vectors and the activation functions, respectively.

A broad class of methods, based on stochastic gradient descent (SGD), are proposed to solve (1.1), such as the vanilla SGD [9], the Adadelta [38], and the Adam [20]. In SGD methods, the gradient of the objective function is calculated by the chain rule, which is applicable to smooth activation functions, such as sigmoid, hyperbolic, and softmax functions [14]. However if a nonsmooth activation function is used, such as the rectified linear unit (ReLU) or the leaky ReLU [27], the subgradient of the objective function in (1.1) is difficult to calculate. At least the chain rule is no longer useful (see [8, Theorem 10.6]). As shown by recent studies, such nonsmooth activation functions have some advantages over the aforementioned smooth ones, as they can pursue the sparsity of the network [13]. The readers may refer to Glorot et al. [13] and Jarrett

*Submitted on 29 March, 2021. Revised on 17 January, 2022

Funding: This work is supported partly by the National Natural Science Foundation of China (No. 12125108, 11971466 and 11991021), Hong Kong Research Grants Council grant PolyU15300120, Key Research Program of Frontier Sciences, Chinese Academy of Sciences (No. ZDBS-LY-7022) and the CAS AMSS-PolyU Joint Laboratory in Applied Mathematics.

†State Key Laboratory of Scientific and Engineering Computing, Academy of Mathmatics and Systems Science, Chinese Academy of Sciences, and University of Chinese Academy of Sciences, China (liuwei175@lsec.cc.ac.cn, liuxin@lsec.cc.ac.cn).

‡Department of Applied Mathmatics, The Hong Kong Polytechnic University, Hong Kong, China (xiaojun.chen@polyu.edu.hk).

et al. [18] for the numerical comparisons between the DNN with smooth activation functions and those with nonsmooth ones. Due to excellent numerical behavior, the ReLU activation function has been widely used since 2010 [1, 11, 30, 33, 37]. In practice, the SGD based approaches are still used to tackle the problem with nonsmooth activation functions. The exactness in calculating the subgradient of a nonsmooth function is usually neglected in SGD methods. Gradients in a neighborhood are often used to approximate the one at a nonsmooth point. Certainly, such approximation may cause theoretical and numerical troubles in some cases. Hence, it is worthwhile to develop algorithms for solving problem (1.1) with nonsmooth activation functions and deal with the nonsmoothness appropriately.

In [5], Carreira-Perpiñán and Wang reformulate problem (1.1) as the following constrained optimization problem with $u_{n,0} = x_n$ for all $n = 1, 2, \ldots, N$,

$$\begin{aligned} &\min_{\substack{W_\ell, b_\ell, u_{n,\ell} \\ \ell=1,\ldots,L, n=1,\ldots,N}} \quad \frac{1}{N}\sum_{n=1}^{N} \psi_n(u_{n,L}) \\ &\quad \text{s.t.} \qquad u_{n,\ell} = \sigma_\ell(W_\ell u_{n,\ell-1} + b_\ell), \quad n = 1, \ldots, N,\ \ell = 1, \ldots, L, \end{aligned} \tag{1.2}$$

and propose a method of auxiliary coordinates to solve (1.2). Moreover, an alternating direction method of multipliers (ADMM) [34] and a block coordinate descent method (BCD) [22] are proposed to solve the constrained model and its l_2-norm penalty problem, respectively. However, these methods are less efficient than SGD based approaches, and lack of theoretical guarantee (see [39]).

More recently, Cui et al. [10] use an l_1-norm penalty method to replace the constraints in (1.2) by adding $\sum_{n=1}^{N}\sum_{\ell=1}^{L} \|u_{n,\ell} - \sigma_\ell(W_\ell u_{n,\ell-1} + b_\ell)\|_1$ in the objective function. They provide an exact penalty analysis and establish the convergence of the sequence generated by their proposed algorithm to a directional stationary point, which will be defined in Definition 2.1. To the best of our knowledge, this is the first mathematically rigorous method for training deep neural networks with nonsmooth activation functions. However, some assumptions imposed in their theoretical analysis are restrictive for some applications. For instance, the ReLU does not satisfies the assumptions on activation functions in [10, Corollary 2.2]. Moreover, the boundedness assumption on the sequences of iterates imposed for convergence analysis is not natural since the solution set of (1.2) is unbounded. For example, suppose that $\{\bar{u}_{n,\ell}, \bar{W}_\ell, \bar{b}_\ell\}_{\ell=1,n=1}^{L,N}$ is a global minimizer of model (1.2), it is easy to verify that $\{\hat{u}_{n,\ell}, \hat{W}_\ell, \hat{b}_\ell\}_{\ell=1,n=1}^{L,N}$ is also a global minimizer, where

$$\begin{gathered} \hat{u}_{n,1} = t\,\bar{u}_{n,1}, \hat{W}_1 = t\bar{W}_1, \hat{b}_1 = t\bar{b}_1, \hat{u}_{n,2} = \bar{u}_{n,2}, \hat{W}_2 = \frac{1}{t}\bar{W}_2, \hat{b}_2 = \frac{1}{t}\bar{b}_2, \\ \hat{u}_{n,\ell} = \bar{u}_{n,\ell}, \hat{W}_\ell = \bar{W}_\ell, \hat{b}_\ell = \bar{b}_\ell, \quad \text{for } n = 1, \ldots, N, \ell = 3, ..., L \end{gathered}$$

for any $t > 0$. Let t tend to infinity, if $\bar{W}_1 \neq 0$ or $\bar{b}_1 \neq 0$, then the norm of $\{\hat{u}_{n,\ell}, \hat{W}_\ell, \hat{b}_\ell\}_{\ell=1,n=1}^{L,N}$ tends to infinity.

To overcome the unboundness of the solution set of (1.2), in this paper we consider the regularized model of problem (1.2) in [5], which adds the regularization term $\|W_\ell\|_F^2$ in the objective function. Motivated by the ideas of the exact l_1-norm penalty and directional stationarity in [10], we design a deterministic algorithm for training the autoencoder, a special two-layer network, using ReLU, with guaranteed convergence, and achieve competitive performances comparing with the SGD based approaches in solving large-scale problems. Our proposed model can be generalized to problem (1.2) with certain regularizing term (see (5.1)–(5.3) in the conclusion part.). In fact, the number of layers does not affect the validity of our theoretical analysis on the model. The reason we focus on the autoencoder in this paper is that a large number of layers does increase lots of tedious notations as well as rapidly increasing number of variables which requires further development on the algorithm to maintain the comparability with existing approaches. Such development is out of the main scope of this paper.

1.1. Regularized Autoencoders. Training an autoencoder using ReLU as the activation function can be formulated as the following nonsmooth nonconvex finite-sum minimization problem.

$$\min_{W,b} \frac{1}{N} \sum_{n=1}^{N} \|\sigma(W^\top \sigma(Wx_n + b_1) + b_2) - x_n\|_2^2, \tag{1.3}$$

where $\{x_n \in \mathbb{R}^{N_0}\}_{n=1}^N$ is the given data, $W \in \mathbb{R}^{N_1 \times N_0}$ is the weight matrix, $b_1 \in \mathbb{R}^{N_1}$ and $b_2 \in \mathbb{R}^{N_0}$ are the bias vectors. For convenience, we use $X = (x_1, x_2, \ldots, x_N) \in \mathbb{R}^{N_0 \times N}$ to denote the data matrix, and denote $b = (b_1^\top, b_2^\top)^\top \in \mathbb{R}^{N_1+N_0}$ as the combination of two bias vectors. Here, we select $W^\top$ as the weight matrix of the second layer, which is the transpose of that of the first layer. The consequent model (1.3) is called the autoencoder with tied weight which has been widely used in practice (see in [14, 16]). However, there exists autoencoder without tied weight, namely, the weight matrices of the two layers take W_1 and W_2, respectively (see in [31]). Nevertheless, Li and Nguyen [26] have shown that by using the tied weight, the training speed is increasing and the numerical performance is comparable than that without tied weight. Then, it becomes uncommon to consider the general case. On the other hand, our new model, algorithm and theorectical analysis can be generalized to the autoencoder without tied weight easily.

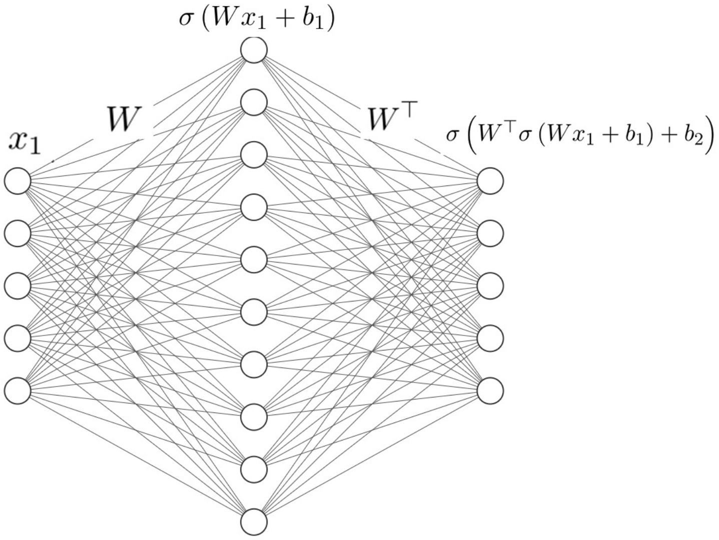


Fig. 1: Illustration of the network of an autoencoder

In this paper, we focus on the ReLU, i.e. $\sigma(y) = y_+ := \max\{0, y\}$. An autoencoder aims to learn a prediction function $\sigma(W^\top \sigma(Wx_n + b_1) + b_2)$ for the given data

$\{x_n \in \mathbb{R}^{N_0}\}_{n=1}^N$ without any label, since $\{x_n\}_{n=1}^N$ is also regarded as the true value of the output layer. Hence, the autoencoder is classified as an unsupervised learning tool. In recent years, autoencoders have been widely used in denoising, dimensionality reduction, and feature learning (e.g., [3, 23, 36]). Besides, autoencoders can be used as a preprocessing tool before training a DNN (e.g., [16, 32]).

In practice, directly solving (1.3) may lead to overfitting or ill-condition. To conquer these issues, the authors of [14] introduce two regularization terms to guarantee the model's robustness. The first class of regularizers is the l_F-norm term $\|W\|_F^2$, called weight decay, which can effectively avoid the overfitting phenomenon [21]. The second class is the l_1-norm that can pursue the sparsity [14, 31]. In this paper, we use the l_F-norm for the weight matrix and the l_1-norm for the auxiliary vectors. To present our optimization model in R^{N_2} with $N_2 = N_0N_1 + N_1 + N_0 + N_1N$, we introduce a vector variable

$$z = (\mathrm{vec}(W)^\top, b^\top, \mathrm{vec}(V)^\top)^\top \in \mathbb{R}^{N_2}, \tag{1.4}$$

where $V = (v_1, v_2, \ldots, v_N) \in \mathbb{R}^{N_1 \times N}$ is an auxiliary variable with $v_n = (Wx_n + b_1)_+$ for all $n = 1, \ldots, N$, and $\mathrm{vec}(Y) \in \mathbb{R}^{lm}$ denotes the columnwise vectorization of the matrix $Y \in \mathbb{R}^{l\times m}$. Let

$$\mathcal{F}(z) = \frac{1}{N}\sum_{n=1}^{N} \|(W^\top v_n + b_2)_+ - x_n\|_2^2 \quad \text{and} \quad \mathcal{R}(z) = \lambda_1 \sum_{n=1}^{N} e^\top v_n + \lambda_2 \|W\|_F^2$$

denote the fidelity term and regularization term, respectively, where $e = (1, \ldots, 1)^\top \in \mathbb{R}^{N_1}$ and $\lambda_1, \lambda_2 > 0$. We consider the following Regularized (R) minimization model for the autoencoders

$$\text{(R)} \qquad \begin{aligned} &\min_z \ \mathcal{F}(z) + \mathcal{R}(z) \\ &\ \text{s.t. } z \in \Omega_1 := \{z : v_n = (Wx_n + b_1)_+,\ n = 1, \ldots, N\}. \end{aligned}$$

We would like to mention that the equivalent form of problem (R), namely (1.3) with regularizer $\mathcal{R}$, has been widely used in autoencoders (see in [14, 36]).

1.2. Our Focuses and Motivation. The feasible set Ω_1 of problem (R) is nonconvex and the standard constraint qualifications may fail due to the nonsmooth equality constraints in (R). Hence, we introduce the following Regularized minimization model with l_1-norm Penalty (RP) for the autoencoders.

$$\text{(RP)} \qquad \begin{aligned} &\min_z \ \mathcal{O}(z) := \mathcal{F}(z) + \mathcal{R}(z) + \mathcal{P}(z) \\ &\ \text{s.t. } z \in \Omega_2 := \{z : v_n \geq (Wx_n + b_1)_+,\ n = 1, \ldots, N\}, \end{aligned}$$

where $\mathcal{P}(z) := \beta \sum_{n=1}^N e^\top (v_n - (Wx_n + b_1)_+)$ is the penalty term.

Compared with the l_1-norm penalty term $\sum_{n=1}^N \|v_n - (Wx_n + b_1)_+\|_1$ proposed in [10], the subdifferential of $\mathcal{P}(z)$ enjoys an explicit expression. In addition, the feasible set Ω_2 of (RP) is convex and the slater-type constraints qualification holds [8, Section 6.3, Proposition 6.3.1]. However, the solution set of (RP) may be unbounded as that of the model in [10]. To overcome the unboundness and ensure the sequence generated by the algorithm is bounded, we introduce a convex set

$$\Omega_3 := \{z : \|b\|_\infty \leq \alpha\},$$

where

$$\alpha = \max\left\{\frac{\theta}{\lambda_1} + \sqrt{\frac{N_1 N_0 \theta}{\lambda_2}}\|X\|_1, \frac{\theta\sqrt{N_1 N_0 \theta}}{\lambda_1\sqrt{\lambda_2}} + \sqrt{N\theta} + \|X\|_1\right\}, \ \theta > \frac{1}{N}\|X\|_F^2. \tag{1.5}$$

We will show that (RP) has a global solution in Ω_3. Hence, it suffices to solve (RP) restricted to Ω_3. Note that $v_n \geq (Wx_n + b_1)_+$ can be represented by $v_n \geq Wx_n + b_1$ and $v_n \geq 0$. Let $\nu = 2(NN_1 + N_0 + N_1)$,

$$A = \left[\begin{array}{ccc} \overbrace{X^\top \otimes I_{N_1}}^{W\,:\,N_0N_1} & \overbrace{e_N \otimes [I_{N_1} \ \ \mathbf{0}]}^{b\,:\,N_1+N_0} & \overbrace{-\,I_{N_1 N}}^{V\,:\,NN_1} \\ \mathbf{0} & \mathbf{0} & -I_{N_1 N} \\ \mathbf{0} & I_{N_1+N_0} & \mathbf{0} \\ \mathbf{0} & -I_{N_1+N_0} & \mathbf{0} \end{array}\right] \in \mathbb{R}^{\nu \times N_2}, \quad c = \left[\begin{array}{c} \mathbf{0} \\ \mathbf{0} \\ \alpha e_{N_1+N_0} \\ \alpha e_{N_1+N_0} \end{array}\right] \in \mathbb{R}^{\nu},$$

where $\otimes$ represents the Kronecker product, $e_N \in \mathbb{R}^N$, $e_{N_1+N_0} \in \mathbb{R}^{N_1+N_0}$ denote the vector whose elements are all one. We consider the following Linearly constrained Regularized minimization model with l_1-norm Penalty (LRP)

$$\text{(LRP)} \qquad \begin{array}{ll} \min\limits_{z} & \mathcal{O}(z) \\ \text{s.t.} & z \in \mathcal{Z} := \Omega_2 \cap \Omega_3 = \{z : Az \leq c\}. \end{array}$$

1.3. Contribution. We propose a partial penalty model (RP) and establish the equivalence between the models (RP) and (R) regarding global minimizers, local minimizers, and directional stationary points under some mild conditions. Moreover, we show that the solution set of (LRP) is bounded and contains at least one of global minimizers of (RP), and provide conditions such that (LRP) and (RP) have the same local minimizers and directional stationary points in $\mathcal{Z}$.

We propose a smoothing proximal gradient algorithm for solving (LRP), whose subproblem at each iteration is a structured strongly convex quadratic program. We develop a splitting algorithm for solving the subproblem by using the special structure, which is faster than the "quadprog" [35] and the "CVX" [15]. We prove that the sequence generated by our algorithm converges to a generalized directional stationary point of (LRP) without assuming the boundness of sequences or existence of accumulation points.

The numerical experiments demonstrate that our algorithm, equipped with adaptively selected stepsize and smoothing parameters, outperforms the popular SGD methods (e.g., Adam, Adadelda, and vanilla SGD) in acquiring better and more robust solutions to a group of randomly generated data sets and one real data set for autoencoders. More specifically, compared with SGD methods, our algorithm achieves lower training error and objective function values, and obtains sparser solutions to testing problems.

1.4. Notations and Organizations. Let $\mathcal{B}_\epsilon(y)$ be the closed Euclidean ball in $\mathbb{R}^m$ centered at y and radius ϵ. The $m \times m$ identity matrix is denoted by I_m. Given a nonempty closed set Ω and a point y^*, we use $\text{dist}(y^*, \Omega) = \inf_{y \in \Omega} ||y - y^*||_2$ to denote the distance from y^* to Ω. We use $\text{co}(\Omega)$ to represent convex hull of Ω.

The rest of this paper is organized as follows. In section 2, we give theoretical results for the relationship among the three models (R), (RP) and (LRP). In section 3, we propose a smoothing proximal gradient algorithm for solving (LRP) and present the global convergence of the algorithm. In section 4, we illustrate the performance of our proposed algorithm through comprehensive numerical experiments. Concluding remarks are given in the last section.

2. Model Analysis.

In this section, we aim to theoretically investigate the relationship among problems (R), (RP) and (LRP) for autoencoders.

2.1. Preliminaries.

In this subsection, we present some preliminary definitions. Let $\text{Proj}_{\Omega}(y^*) = \arg\min \{\|y - y^*\|_2 : y \in \Omega\}$ denote the orthogonal projection of a vector $y^* \in \mathbb{R}^m$ onto a convex set $\Omega \subseteq \mathbb{R}^m$.

The Clarke subdifferential [8, Section 1.2] of a locally Lipschitz continuous function $f : \mathbb{R}^m \to \mathbb{R}$ at y^* is defined by $\partial f(y^*) = \text{co}\{\lim_{y \to y^*} \nabla f(y) : f \text{ is smooth at } y\}$.

We use $f'(y; d)$ to denote the directional derivative of a directional differentiable function f at y along the direction d, i.e.,

$$f'(y; d) = \lim_{t \downarrow 0} \frac{f(y + td) - f(y)}{t}. \tag{2.1}$$

A function f is said to be regular [8, Definition 2.3.4] at $\bar{y} \in \mathbb{R}^m$ provided that if for all d, the directional derivative $f'(\bar{y}; d)$ exists, and

$$f'(\bar{y}; d) = f^\circ(\bar{y}; d),$$

where $f^\circ(\bar{y}; d) := \limsup_{\substack{y \to \bar{y} \\ t \downarrow 0}} \frac{f(y+td) - f(y)}{t}$ is the generalized directional derivative at $\bar{y}$ along the direction d [8].

It is known that if f is piecewise smooth and Lipschitz continuous in a neighborhood of y, then f is semismooth and directional differentiable at y [29]. The objective functions of (R), (RP) and (LRP) are locally Lipschitz and piecewise smooth. Hence they are semismooth functions and directional differentiable.

Let $\mathcal{T}_{\Omega}(\bar{y}) = \{d : d = \lim_{y \in \Omega, y \to \bar{y}, \tau \downarrow 0} \frac{y - \bar{y}}{\tau}\}$ be the tangent cone of a set Ω at $\bar{y}$.

Definition 2.1. *We call $\bar{z} \in \Omega_1$, $\bar{z} \in \Omega_2$, $\bar{z} \in \mathcal{Z}$ a d(irectional)-stationary point [10] of problems (R), (RP) and (LRP), respectively, if*

$$\mathcal{F}'(\bar{z}; d) + \nabla \mathcal{R}(\bar{z})^\top d \geq 0, \quad \forall d \in \mathcal{T}_{\Omega_1}(\bar{z}), \tag{2.2}$$

$$\mathcal{O}'(\bar{z}; d) \geq 0, \quad \forall d \in \mathcal{T}_{\Omega_2}(\bar{z}), \tag{2.3}$$

$$\mathcal{O}'(\bar{z}; d) \geq 0, \quad \forall d \in \mathcal{T}_{\mathcal{Z}}(\bar{z}). \tag{2.4}$$

We call $\bar{z} \in \Omega_1$, $\bar{z} \in \Omega_2$, $\bar{z} \in \mathcal{Z}$ a generalized d(irectional)-stationary point of problems (R), (RP) and (LRP), respectively, if (2.2)–(2.4) hold with $\mathcal{F}^\circ(\bar{z}; d)$ and $\mathcal{O}^\circ(\bar{z}; d)$ instead of $\mathcal{F}'(\bar{z}; d)$ and $\mathcal{O}'(\bar{z}; d)$.

We call $\bar{z} \in \mathcal{Z}$ a generalized KKT point of (LRP) if there exists a nonnegative vector $\bar{\gamma} \in \mathbb{R}^\nu$ such that

$$0 \in \partial \mathcal{O}(\bar{z}) + A^\top \bar{\gamma}, \quad \bar{\gamma}^\top (A\bar{z} - c) = 0, \quad A\bar{z} - c \leq 0. \tag{2.5}$$

2.2. Global and Local Solutions.

Let $\mathcal{S}^*$, $\mathcal{S}$ and $\mathcal{Z}^*$ be the global solution sets of (R), (RP) and (LRP), respectively. In this subsection, we prove $\mathcal{S}^*$, $\mathcal{S}$ and $\mathcal{Z}^*$ are not empty, and $\mathcal{Z}^* \subset \mathcal{S} = \mathcal{S}^*$.

We define a level set of problem (RP) using θ defined in (1.5) as follows

$$\Omega_\theta = \{z \in \Omega_2 : \mathcal{O}(z) \leq \theta\}.$$

Obviously, $0 \in \Omega_\theta$ since $\mathcal{O}(0) = \frac{1}{N}\|X\|_F^2 < \theta$ and $0 \in \Omega_2$.

Theorem 2.2. *For any $z \in \Omega_\theta$, the following statements hold.*
(a) $\|W\|_F^2 \le \frac{\theta}{\lambda_2}$, $\|V\|_1 \le \frac{\theta}{\lambda_1}$ and $\|b_+\|_\infty \le \alpha$.
(b) $\bar{z} = \mathrm{Proj}_{\Omega_3}(z) \in \mathcal{Z}$ and $\mathcal{O}(\bar{z}) = \mathcal{O}(z)$.
Moreover, the solution set $\mathcal{Z}^*$ of (LRP) is not empty and bounded, and $\mathcal{Z}^* \subset \mathcal{S}$.

Proof. (a) The first two inequalities are from $V \ge 0$, $\mathcal{O}(z) \le \theta$, $\mathcal{F}(z) \ge 0$, $\mathcal{P}(z) \ge 0$ and $\mathcal{R}(z) \ge 0$, which imply

$$\lambda_1 \sum_{n=1}^{N} e^\top v_n \le \theta \ \text{ and } \ \lambda_2 \|W\|_F^2 \le \theta. \tag{2.6}$$

Now we prove $\|b_+\|_\infty \le \alpha$. From the Cauchy inequality and (2.6), we have

$$\sum_{j=1}^{N_1} \sum_{s=1}^{N_0} |W_{j,s}| \le \sqrt{N_1 N_0} \|W\|_F \le \sqrt{\frac{N_1 N_0 \theta}{\lambda_2}}. \tag{2.7}$$

For $n = 1, \dots, N$, combining (2.6) with $z \in \Omega_2$, we obtain that

$$\frac{\theta}{\lambda_1} \ge e^\top v_n \ge e^\top (W x_n + b_1)_+ \ge (W_{j,\cdot} x_n + b_{1,j})_+ \ge W_{j,\cdot} x_n + b_{1,j} \tag{2.8}$$

for all $j = 1, \dots, N_1$. On the other hand, (2.7) yields $\|W_{j,\cdot}\|_\infty \le \sqrt{\frac{N_1 N_0 \theta}{\lambda_2}}$, which implies

$$|W_{j,\cdot} x_n| \le \sqrt{\frac{N_1 N_0 \theta}{\lambda_2}} \|X\|_1. \tag{2.9}$$

Together with (2.8), we can conclude that $b_{1,j}$ satisfies

$$b_{1,j} \le \frac{\theta}{\lambda_1} + \sqrt{\frac{N_1 N_0 \theta}{\lambda_2}} \|X\|_1, \quad \forall j = 1, \dots, N_1. \tag{2.10}$$

From $\mathcal{F}(z) \le \theta$, we have

$$\sqrt{N\theta} \ge (W_{\cdot,j}^\top v_n + b_{2,j})_+ - X_{j,n} \ge W_{\cdot,j}^\top v_n + b_{2,j} - X_{j,n} \tag{2.11}$$

for all $n = 1, \dots, N$ and $j = 1, \dots, N_0$. From $\|v_n\|_1 \le \frac{\theta}{\lambda_1}$ and $\|W_{\cdot,j}\|_1 \le \sqrt{\frac{N_1 N_0 \theta}{\lambda_2}}$, we find

$$|W_{\cdot,j}^\top v_n| \le \frac{\theta \sqrt{N_1 N_0 \theta}}{\lambda_1 \sqrt{\lambda_2}}. \tag{2.12}$$

Together with (2.11), we obtain that

$$b_{2,j} \le \frac{\theta \sqrt{N_1 N_0 \theta}}{\lambda_1 \sqrt{\lambda_2}} + \sqrt{N\theta} + \|X\|_1, \quad \forall j = 1, \dots, N_0. \tag{2.13}$$

Combining (2.10) and (2.13), we finally arrive at the assertion that $\|b_+\|_\infty \le \alpha$.

(b) Let $\bar{z} = (\mathrm{vec}(\bar{W})^\top, \bar{b}^\top, \mathrm{vec}(\bar{V})^\top)^\top$ with $\bar{W} = W$, $\bar{V} = V$ and

$$\bar{b}_{1,j_1} = \begin{cases} b_{1,j_1} & \text{if } b_{1,j_1} \ge -\alpha, \\ -\alpha & \text{otherwise,} \end{cases} \qquad \bar{b}_{2,j_2} = \begin{cases} b_{2,j_2} & \text{if } b_{2,j_2} \ge -\alpha, \\ -\alpha & \text{otherwise} \end{cases} \tag{2.14}$$

for all $j_1 = 1, \dots, N_1$, $j_2 = 1, \dots, N_0$. By part (a), we have $\|\bar{b}\| \le \alpha$. Hence $\bar{z} \in \Omega_3$.

By (2.9) and (2.12), we have

$$b_{1,j_1} + W_{j_1,\cdot}x_n \le \bar{b}_{1,j_1} + \bar{W}_{j_1,\cdot}x_n \le -\alpha + \sqrt{\frac{N_1 N_0 \theta}{\lambda_2}}\|X\|_1 \le 0, \text{ if } \bar{b}_{1,j_1} = -\alpha,$$

$$b_{2,j_2} + W_{\cdot,j_2}^\top v_n \le \bar{b}_{2,j_2} + \bar{W}_{\cdot,j_2}^\top \bar{v}_n \le -\alpha + \frac{\theta\sqrt{N_1 N_0 \theta}}{\lambda_1\sqrt{\lambda_2}} \le 0, \qquad \text{if } \bar{b}_{2,j_2} = -\alpha,$$

which together with (2.14) implies that for all $n = 1, \dots, N$, it holds

$$(\bar{W}x_n + \bar{b}_1)_+ = (Wx_n + b_1)_+ \text{ and } (\bar{W}\bar{v}_n + \bar{b}_2)_+ = (Wv_n + b_2)_+. \tag{2.15}$$

Combining with $\bar{W} = W$ and $\bar{V} = V$, we have $\mathcal{O}(\bar{z}) = \mathcal{O}(z)$. Moreover (2.15), $\bar{W} = W$ and $\bar{V} = V$ yield $\bar{z} \in \Omega_2$. Hence by the definition of $\bar{z}$, $\bar{z} = \text{Proj}_{\Omega_3}(z) \in \Omega_2 \cap \Omega_3 = \mathcal{Z}$.

Now we prove the last statement. By parts (a) and (b), the set $\mathcal{Z} \cap \Omega_\theta$ is a bounded closed set. Hence, there exists $z^* \in \mathcal{Z}^*$ such that $\mathcal{O}(z^*) = \min_{z \in \mathcal{Z}} \mathcal{O}(z) \le \theta$. Assume on contradiction that $z^* \in \mathcal{Z}^*$, but $z^* \notin \mathcal{S}$. Then there exists $\widetilde{z} \in \Omega_2$ such that $\mathcal{O}(\widetilde{z}) < \mathcal{O}(z^*) \le \theta$. As we have proved in (b), $\bar{z} =\text{Proj}_{\Omega_3}(\widetilde{z}) \in \mathcal{Z}$ and $\mathcal{O}(\bar{z}) = \mathcal{O}(\widetilde{z})$, which implies $\mathcal{O}(\bar{z}) < \mathcal{O}(z^*)$. This is a contradiction. Hence $\mathcal{Z}^* \subset \mathcal{S}$. □

The following theorem shows that (RP) is an exact penalty formulation of (R) regarding global minimizers if the penalty parameter β in $\mathcal{P}$ is larger than a computable number.

Theorem 2.3. *The following statements hold.*

(a) The functions $\mathcal{F}$ and $\mathcal{R}$ are Lipschitz continuous over Ω_θ.

(b) Let $L_\mathcal{F}$ and $L_\mathcal{R}$ be Lipschitz modulus of $\mathcal{F}$ and $\mathcal{R}$ over Ω_θ, respectively. Suppose $\beta > L_\mathcal{F} + L_\mathcal{R}$. If $\bar{z} \in \Omega_\theta$ is a global minimizer of (R), then $\bar{z}$ is also a global minimizer of (RP).

(c) Let $\delta := 3\theta + \frac{2N\theta^3}{\lambda_1^2\lambda_2}$ and $\Omega_\delta = \{z \in \Omega_2 : \mathcal{O}(z) \le \delta\}$. Let $L_\mathcal{F}$ and $L_\mathcal{R}$ be Lipschitz modulus of $\mathcal{F}$ and $\mathcal{R}$ over Ω_δ respectively. Suppose $\beta > L_\mathcal{F} + L_\mathcal{R}$. If $\bar{z} \in \Omega_\theta$ is a global minimizer of (RP), then $\bar{z}$ is also a global minimizer of (R).

Proof. (a) From Theorem 2.2 (a), it is clear that $\mathcal{R}$ is Lipschitz continuous over Ω_θ. From Theorem 2.2 (a)-(b), the set $\mathcal{Z} \cap \Omega_\theta$ is bounded. Suppose that $L_\mathcal{F}$ is the Lipschitz constant of $\mathcal{F}$ over $\mathcal{Z} \cap \Omega_\theta$. Let $z_1, z_2 \in \Omega_\theta$. It follows Theorem 2.2 (b) that $\mathcal{O}(\text{Proj}_{\Omega_3}(z_1)) = \mathcal{O}(z_1)$, $\mathcal{O}(\text{Proj}_{\Omega_3}(z_2)) = \mathcal{O}(z_2)$, and $\text{Proj}_{\Omega_3}(z_1), \text{Proj}_{\Omega_3}(z_2) \in \mathcal{Z}$. From $\mathcal{Z} = \Omega_2 \cap \Omega_3$, we have $\text{Proj}_{\Omega_3}(z_1), \text{Proj}_{\Omega_3}(z_2) \in \Omega_\theta$. Hence, it holds that

$$\begin{aligned}\|\mathcal{O}(z_1) - \mathcal{O}(z_2)\|_2 &= \|\mathcal{O}(\text{Proj}_{\Omega_3}(z_1)) - \mathcal{O}(\text{Proj}_{\Omega_3}(z_2))\|_2 \\ &\le L_\mathcal{F}\|\text{Proj}_{\Omega_3}(z_1) - \text{Proj}_{\Omega_3}(z_2)\|_2 \\ &\le L_\mathcal{F}\|z_1 - z_2\|_2,\end{aligned}$$

where the last inequality is from that Ω_3 is a convex set and the projection is Lipschitz continuous with Lipschitz constant 1. Hence we derive that $\mathcal{F}$ is Lipschitz continuous over Ω_θ with the Lipschitz constant $L_\mathcal{F}$.

(b) We first prove that $\beta\text{dist}(z, \Omega_1) \le \mathcal{P}(z)$ for all $z \in \Omega_2$.

For $z \in \Omega_2$, let $\widetilde{z} = (\text{vec}(W)^\top, b^\top, \text{vec}(\widetilde{V})^\top)^\top$ with $\widetilde{v}_n = (Wx_n + b_1)_+$ for all $n = 1, 2, \dots, N$. Then, we have $\widetilde{z} \in \Omega_1$, $\widetilde{v}_n \le v_n$, and

$$\text{dist}(z, \Omega_1) \le \|z - \widetilde{z}\|_2 \le \|\text{vec}(V - \widetilde{V})\|_2 \le \sum_{n=1}^{N} \|v_n - (Wx_n + b_1)_+\|_1 = \frac{1}{\beta}\mathcal{P}(z),$$

where the last inequality comes from the definition of $\widetilde{v}_n$ and $\|\cdot\|_2 \le \|\cdot\|_1$, and the equality is from $z \in \Omega_2$.

Since $\beta > L_{\mathcal{F}} + L_{\mathcal{R}}$, $\mathcal{P}(z) = 0$ for all $z \in \Omega_1$, $\Omega_1 \subset \Omega_2$, and $\Omega_\theta \subset \Omega_2$, we have

$$\begin{aligned}\min_{z\in\Omega_\theta} \mathcal{F}(z) + \mathcal{R}(z) + \mathcal{P}(z) &\ge \min_{z\in\Omega_1\cap\Omega_\theta} \mathcal{F}(z) + \mathcal{R}(z)\\ &= \min_{z\in\Omega_1\cap\Omega_\theta} \mathcal{F}(z) + \mathcal{R}(z) + \mathcal{P}(z)\\ &\ge \min_{z\in\Omega_\theta} \mathcal{F}(z) + \mathcal{R}(z) + \mathcal{P}(z).\end{aligned}$$

Hence we obtain the statement (b).

(c) Let $\bar{z} \in \mathcal{S}$ and $\zeta_2(V) = \frac{1}{N}\sum_{n=1}^N \left\|\left(\bar{W}^\top v_n + \bar{b}_2\right)_+ - x_n\right\|_2^2 + \lambda_1 \sum_{n=1}^N e^\top v_n$. By the definition of $\bar{V}$, $\bar{V}$ is a global minimizer of

$$\min_{V\in\Omega_4} \zeta_1(V) := \zeta_2(V) + \beta \sum_{n=1}^N \left\|v_n - (\bar{W}x_n + \bar{b}_1)_+\right\|_1,$$

where $\Omega_4 = \left\{V : v_n \ge (\bar{W}x_n + \bar{b}_1)_+, n = 1, \ldots, N, \zeta_1(V) \le \delta\right\}$.

It follows from $\mathcal{O}(\bar{z}) \le \mathcal{O}(0) < \theta$ that $\sum_{n=1}^N \left\|(\bar{W}^\top \bar{v}_n + \bar{b}_2)_+ - x_n\right\|_2^2 \le N\theta$, $\lambda_2\|\bar{W}\|_F^2 \le \theta$, $\lambda_1 \sum_{n=1}^N e^\top \bar{v}_n \le \theta$, and $\beta \sum_{n=1}^N e^\top r_n \le \theta$, where $r_n = \bar{v}_n - (\bar{W}x_n + \bar{b}_1)_+ \ge 0$ for $n = 1, 2, \ldots, N$. Let

$$\Omega_5 = \left\{V : v_n = (\bar{W}x_n + \bar{b}_1)_+,\ n = 1, \ldots, N,\ \zeta_2(V) \le \delta\right\}$$

and $\hat{v}_n = (\bar{W}x_n + \bar{b}_1)_+$ for all $n = 1, \ldots, N$. We show that $\hat{V} \in \Omega_5$ as follows.

$$\begin{aligned}\zeta_2(\hat{V}) &= \frac{1}{N}\sum_{n=1}^N \sum_{j=1}^{N_0} \left|\left(\bar{W}_{\cdot,j}^\top (\bar{v}_n - r_n) + \bar{b}_{2,j}\right)_+ - X_{j,n}\right|^2 + \lambda_1 \sum_{n=1}^N e^\top (\bar{v}_n - r_n)\\ &\le \theta + \frac{1}{N}\sum_{n=1}^N \sum_{j=1}^{N_0} \left(\left|X_{j,n} - \left(\bar{W}_{\cdot,j}^\top \bar{v}_n + \bar{b}_{2,j}\right)_+\right| + (\bar{W}_{\cdot,j}^\top r_n)_+\right)^2\\ &\le \theta + \frac{2}{N}\sum_{n=1}^N \sum_{j=1}^{N_0} \left(\left|\left(\bar{W}_{\cdot,j}^\top \bar{v}_n + \bar{b}_{2,j}\right)_+ - X_{j,n}\right|^2 + (\bar{W}_{\cdot,j}^\top r_n)_+^2\right)\\ &\le \theta + 2\theta + \frac{2}{N}\sum_{n=1}^N \sum_{j=1}^{N_0} (\bar{W}_{\cdot,j}^\top r_n)_+^2 \le 3\theta + \frac{2}{N}\sum_{n=1}^N \sum_{j=1}^{N_0} (\bar{W}_{\cdot,j}^\top r_n)^2\\ &\le 3\theta + \frac{2}{N}\|\bar{W}\|_F^2 \left(\sum_{n=1}^N e^\top r_n\right)^2 \le \delta,\end{aligned}$$

where the first inequality comes from $|(a_1 + a_2)_+ + a_3| \le |(a_1)_+ + a_3| + (a_2)_+$ with $a_1, a_2, a_3 \in \mathbb{R}$, the second last inequality uses the fact $r_n \ge 0$, and the last inequality is from $\lambda_2\|\bar{W}\|_F^2 \le \theta$, $\beta \sum_{n=1}^N e^\top r_n \le \theta$, $\beta > L_{\mathcal{R}} \ge \lambda_1$, and the definition of δ. Hence Ω_5 is nonempty. Obviously $\Omega_5 \subset \Omega_4$ and $\{z : W = \bar{W}, b = \bar{b}, V \in \Omega_5\} \subset \Omega_1$.

On the other hand, it is clear that $L_{\mathcal{F}} + L_{\mathcal{R}}$ is also a Lipschitz constant of $\zeta_2(V)$ over Ω_4. Besides, we have $\sum_{n=1}^N \|v_n - (\bar{W}x_n + \bar{b}_1)_+\|_1 \ge \operatorname{dist}(z, \Omega_5)$ for all $z \in \Omega_4$, which is resulted from [17, Proposition 4]. Together with $\beta > L_{\mathcal{F}} + L_{\mathcal{R}}$ and [7, Lemma 3.1], we obtain $\bar{V}$ is also a global minimizer of $\min_{V\in\Omega_5} \zeta_2(V)$. Hence $\bar{z} \in \Omega_1$. From $\Omega_1 \subset \Omega_2$, we obtain $\bar{z} \in \mathcal{S}^*$. We complete the proof. ∎

The above two theorems show that the solution sets $\mathcal{S}^*$, $\mathcal{S}$ of problems (R) and (RP) are the same and contain the solution set $\mathcal{Z}^*$ of (LRP) that is bounded. The following example shows that the solution sets $\mathcal{S}^*$ and $\mathcal{S}$ are unbounded for some data set X.

Example 2.1 Let z^* be a global minimizer of problem (R) with

$$X = (x_1, x_2) = \begin{bmatrix} 0 & 0 \\ 1 & 2 \end{bmatrix} \in \mathbb{R}^{2\times 2},\; W = \begin{bmatrix} w_1, w_2 \end{bmatrix} \in \mathbb{R}^{1\times 2},\; b_1 \in \mathbb{R},\; b_2 = \begin{bmatrix} b_{2,1} \\ b_{2,2} \end{bmatrix} \in \mathbb{R}^2.$$

We set $\widehat{z} = (\mathrm{vec}(W^*)^\top, \hat{b}^\top, \mathrm{vec}(V^*)^\top)^\top \in \mathbb{R}^{N_2}$ with $\hat{b}_1 = b_1^*$, $\hat{b}_{2,2} = b_{2,2}^*$ and $\hat{b}_{2,1} \le \min\{-w_1^*(w_2^* + b_1^*)_+, -w_1^*(2w_2^* + b_1^*)_+\}$.

From $\hat{b}_2 \in \arg\min_{b_2} \sum_{i=n}^{2} \left\|((W^*)^\top(W^* x_n + b_1^*)_+ + b_2)_+ - x_n\right\|_2^2$, we can verify that $\widehat{z}$ is also a global minimizer of (R). Hence $\mathcal{S}^*$ is unbounded, and the solution set $\mathcal{S}$ of problem (RP) with any large $\beta > 0$ is also unbounded.

By the similar argument, we can claim the following relationships among the local minimizers of (R), (RP) and (LRP).

Corollary 2.4. *Let $L_{\mathcal{F}}$ and $L_{\mathcal{R}}$ be Lipschitz modulus of $\mathcal{F}$ and $\mathcal{R}$ over Ω_θ respectively. Suppose $\beta > L_{\mathcal{F}} + L_{\mathcal{R}}$. If $\bar{z} \in \Omega_\theta$ is a local minimizer of* (R) *or* (LRP), *then $\bar{z}$ is also a local minimizer of* (RP). *If $\bar{z} \in \Omega_\theta \cap \Omega_3$ is a local minimizer of* (RP), *then $\bar{z}$ is also a local minimizer of* (LRP).

2.3. Stationary Points. In this subsection, we investigate the relationships among the stationary points of problems (R), (RP) and (LRP).

From $\mathcal{Z} = \Omega_2 \cap \Omega_3$, we have $\{z : z \in \mathcal{Z},\, \mathcal{O}(z) \le \theta\} \subset \Omega_\theta$.

Theorem 2.5. *Let $L_{\mathcal{F}}$ and $L_{\mathcal{R}}$ be the Lipschitz modulus of $\mathcal{F}$ and $\mathcal{R}$ over Ω_θ. Suppose $\beta > L_{\mathcal{F}} + L_{\mathcal{R}}$. If $\bar{z} \in \mathcal{Z}$ with $\mathcal{O}(\bar{z}) < \theta$ is a d-stationary point of* (LRP), *then $\bar{z} \in \Omega_1$ is a d-stationary point of* (RP) *and* (R).

Proof. Firstly, we show $\bar{z} \in \Omega_1$.

Assume on contradiction that $\bar{z} \notin \Omega_1$, we construct $\widetilde{z} = (\mathrm{vec}(\bar{W})^\top, \bar{b}^\top, \mathrm{vec}(\widetilde{V})^\top)^\top$ with $\widetilde{v}_n = (\bar{W}x_n + \bar{b}_1)_+$ for all $n = 1, \ldots, N$. It then follows from $\bar{z} \in \mathcal{Z}$ that $\bar{z} \in \Omega_2$, which further implies that $\bar{v}_n \ge (\bar{W}x_n + \bar{b}_1)_+$ for all $n = 1, \ldots, N$. Hence, we have $\widetilde{V} \le \bar{V}$ and $\widetilde{V} \ne \bar{V}$.

Since $\mathcal{O}(\bar{z}) < \theta$ and $\mathcal{O}$ is locally Lipschitz continuous, there exists $t_1 \in (0, 1]$ such that $\mathcal{O}(\bar{z} + t(\widetilde{z} - \bar{z})) \le \theta$ for all $0 < t < t_1$. Together with $\|\cdot\|_2 \le \|\cdot\|_1$ and the definition of $\bar{z}$ and $\widetilde{z}$, we have

$$\begin{aligned} \mathcal{O}(\bar{z} + t(\widetilde{z} - \bar{z})) - \mathcal{O}(\bar{z}) &= \mathcal{F}(\bar{z} + t(\widetilde{z} - \bar{z})) - \mathcal{F}(\bar{z}) \\ &\quad + \mathcal{R}(\bar{z} + t(\widetilde{z} - \bar{z})) - \mathcal{R}(\bar{z}) + \beta \sum_{n=1}^{N} e^\top(\bar{v}_n + t(\widetilde{v}_n - \bar{v}_n)) - \beta \sum_{n=1}^{N} e^\top \bar{v}_n \\ &\le t(L_{\mathcal{F}} + L_{\mathcal{R}})\|\bar{z} - \widetilde{z}\|_2 + t\beta \sum_{n=1}^{N} e^\top(\widetilde{v}_n - \bar{v}_n) \le t(\beta - (L_{\mathcal{F}} + L_{\mathcal{R}})) \sum_{n=1}^{N} e^\top(\widetilde{v}_n - \bar{v}_n). \end{aligned}$$

Together with $\beta > L_{\mathcal{F}} + L_{\mathcal{R}}$, $\widetilde{V} \ne \bar{V}$, and $\widetilde{v}_n \le \bar{v}_n$ for all $n = 1, \ldots, N$, we arrive at

$$\mathcal{O}'(\bar{z}; \widetilde{z} - \bar{z}) \le (\beta - (L_{\mathcal{F}} + L_{\mathcal{R}})) \sum_{n=1}^{N} e^\top(\widetilde{v}_n - \bar{v}_n) < 0. \tag{2.16}$$

On the other hand, it holds that $\mathcal{T}_{\mathcal{Z}}(\bar{z}) = \{d : (Ad)_i \le 0,\, i \in \mathcal{A}\}$ where $\mathcal{A} = \{i \in \{1, \ldots, \nu\} : (A\bar{z})_i = c_i\}$. Since $\widetilde{W} = \bar{W}, \widetilde{b} = \bar{b}, \widetilde{V} \le \bar{V}$, we have $(A(\widetilde{z} - \bar{z}))_i \le 0$, for all $i \in \mathcal{A}$, i.e. $\widetilde{z} - \bar{z} \in \mathcal{T}_{\mathcal{Z}}(\bar{z})$. This together with (2.16) contradicts to that $\bar{z}$ is a d-stationary point of (LRP) in (2.4), which means $O'(\bar{z}; d) \geq 0$ for all $d \in \mathcal{T}_{\mathcal{Z}}(\bar{z})$. Hence, we have $\bar{z} \in \Omega_1$, which implies $\bar{v}_n = (\bar{W} x_n + \bar{b}_1)_+$ holds for all $n = 1, \dots, N$.

Secondly, we prove that $\bar{z}$ is a d-stationary point of (RP). For any $d \in \mathcal{T}_{\Omega_2}(\bar{z})$, there exists $t_2 \in (0, 1]$ such that $\bar{z} + td \in \Omega_2$ for all $0 \leq t \leq t_2$, since Ω_2 is a convex set. Together with Theorem 2.2 (b), we have $\mathrm{Proj}_{\Omega_3}(\bar{z} + td) \in \mathcal{Z}$ for all $0 \leq t \leq t_2$, and

$$\mathcal{O}'(\bar{z}; d) = \lim_{t \downarrow 0} \frac{\mathcal{O}(\bar{z} + td) - \mathcal{O}(\bar{z})}{t} = \lim_{t \downarrow 0, t \leq t_2} \frac{\mathcal{O}(\mathrm{Proj}_{\Omega_3}(\bar{z} + td)) - \mathcal{O}(\bar{z})}{t}. \tag{2.17}$$

Define a function $\zeta : \mathbb{R}_+ \mapsto \mathbb{R}^{N_2}$ satisfying $\zeta(t) = \mathrm{Proj}_{\Omega_3}(\bar{z} + td)$, then $\zeta(t)$ is a piecewise linear function with respect to t, due to the explicit formula of $\mathrm{Proj}_{\Omega_3}(z)$ for $z \in \Omega_\theta$ (cf. (2.14)). Hence, there exists $t_3 \in (0, t_2]$ such that for all $0 < t < t_3$, we have $\zeta(t) = (1 - \frac{t}{t_3})\zeta(0) + \frac{t}{t_3}\zeta(t_3)$. Together with (2.17), $t_3 \leq t_2$, $\zeta(0) = \bar{z}$, and $\mathrm{Proj}_{\Omega_3}(\bar{z} + t_3 d) - \bar{z} \in \mathcal{T}_{\mathcal{Z}}(\bar{z})$, we arrive at

$$\mathcal{O}'(\bar{z}; d) = \lim_{t \downarrow 0} \frac{\mathcal{O}(\bar{z} + \frac{t}{t_3}(\zeta(t_3) - \bar{z})) - \mathcal{O}(\bar{z})}{t} = \mathcal{O}'\left(\bar{z}; \frac{1}{t_3}(\zeta(t_3) - \bar{z})\right) \geq 0$$

for all $d \in \mathcal{T}_{\Omega_2}(\bar{z})$. Hence $\bar{z}$ is a d-stationary point of (RP).

Finally, we prove that $\bar{z}$ is a d-stationary point of (R). Since the difference between the objective functions of (RP) and (R) is the term $\mathcal{P}$, we only need to prove $\mathcal{P}'(\bar{z}; d) = 0$ for all $d \in \mathcal{T}_{\Omega_1}(\bar{z})$, which together with $\Omega_1 \subset \Omega_2$ and $\mathcal{O}'(\bar{z}; d) \geq 0$ for all $d \in \mathcal{T}_{\Omega_2}(\bar{z})$ yields (2.2).

For a fixed $d \in \mathcal{T}_{\Omega_1}(\bar{z})$, by the definition of $\mathcal{T}_{\Omega_1}(\bar{z})$, let $\{\tau_k\}$ be a sequence of positive numbers with $\tau_k \leq \tau_{k-1}$ converging to zero, and $\{z^{(k)}\} \subset \Omega_1$ a sequence converging to $\bar{z}$ such that $d = \lim_{k \to \infty} d^{(k)}$ with $d^{(k)} = \frac{z^{(k)} - \bar{z}}{\tau_k}$. From $z^{(k)} = \bar{z} + \tau_k d^{(k)} \in \Omega_1$, we have $\mathcal{P}(\bar{z} + \tau_k d^{(k)}) = 0$. Note that $\bar{z} \in \Omega_1$ implies $\mathcal{P}(\bar{z}) = 0$. Hence from the Lipschitz continuity and directional differentiability of $\mathcal{P}$, we obtain

$$\begin{aligned} \mathcal{P}'(\bar{z}; d) &= \lim_{t \downarrow 0} \frac{\mathcal{P}(\bar{z} + td) - \mathcal{P}(\bar{z})}{t} = \lim_{\tau_k \downarrow 0} \frac{\mathcal{P}(\bar{z} + \tau_k d) - \mathcal{P}(\bar{z})}{\tau_k} \\ &= \lim_{\tau_k \downarrow 0} \frac{\mathcal{P}(\bar{z} + \tau_k d) - \mathcal{P}(\bar{z} + \tau_k d^{(k)})}{\tau_k} = 0. \end{aligned}$$

Since $d \in \mathcal{T}_{\Omega_1}(\bar{z})$ is arbitrarily chosen, we complete the proof. ∎

THEOREM 2.6. *Let $L_{\mathcal{F}}$ and $L_{\mathcal{R}}$ be the Lipschitz modulus of $\mathcal{F}$ and $\mathcal{R}$ over Ω_θ. Suppose $\beta > L_{\mathcal{F}} + L_{\mathcal{R}}$. If $\bar{z} \in \mathcal{Z}$ with $\mathcal{O}(\bar{z}) < \theta$ is a generalized KKT point of* (LRP)*, then $\bar{z} \in \Omega_1$ is a generalized d-stationary point of* (LRP)*. In additional, if $\bar{z} \in \mathrm{int}(\Omega_3)$, then $\bar{z}$ is a generalized d-stationary point of* (RP)*. Furthermore, if $\mathcal{P}$ is regular at $\bar{z}$, then $\bar{z}$ is a generalized d-stationary point of* (R).

Proof. By the definition of generalized KKT point of (LRP), [8, Proposition 2.1.2] and $\mathcal{T}_{\mathcal{Z}}(\bar{z}) = \{d : (Ad)_i \leq 0,\ i \in \mathcal{A}\}$ where $\mathcal{A} = \{i \in \{1, \dots, \nu\} : (A\bar{z})_i = c_i\}$, we have

$$0 \leq -(Ad)^\top \bar{\gamma} \leq \max_{\xi \in \partial \mathcal{O}(\bar{z})} \xi^\top d = \mathcal{O}^\circ(\bar{z}; d), \quad \forall d \in \mathcal{T}_{\mathcal{Z}}(\bar{z}),$$

which implies that $\bar{z}$ is a generalized d-stationary point of (LRP).

Now, we prove that $\bar{z} \in \Omega_1$.

Assume on contradiction that $\bar{z} \notin \Omega_1$, we construct the same $\widetilde{z}$ as that in the proof of Theorem 2.5. Since $\mathcal{O}(\bar{z}) < \theta$ and $\mathcal{O}$ is locally Lipschitz continuous, there exists $\epsilon > 0$ such that for all $z \in \mathcal{B}_\epsilon(\bar{z})$, it holds that $\mathcal{O}(z) < \theta$. Furthermore, for any $z \in \mathcal{B}_\epsilon(\bar{z})$, there exists $t_1 \in (0, 1]$ such that $\mathcal{O}(z + t(\widetilde{z} - \bar{z})) \leq \theta$ for all $0 < t < t_1$. Together with $\|\cdot\|_2 \leq \|\cdot\|_1$ and the definition of z and $\widetilde{z}$, we also have

$$\mathcal{O}(z + t(\widetilde{z} - \bar{z})) - \mathcal{O}(z) \leq t(\beta - (L_{\mathcal{F}} + L_{\mathcal{R}})) \sum_{n=1}^{N} e^\top (\widetilde{v}_n - \bar{v}_n).$$

Using a similar method as that in the proof of Theorem 2.5, we have $\bar{z} \in \Omega_1$, which implies $\bar{v}_n = (\bar{W} x_n + \bar{b}_1)_+$ holds for all $n = 1, \ldots, N$.

Since $\bar{z} \in \operatorname{int}(\Omega_3)$ implies $\mathcal{T}_{\mathcal{Z}}(\bar{z}) = \mathcal{T}_{\Omega_2}(\bar{z})$, we obtain that $\bar{z}$ is a generalized d-stationary point of (RP).

Finally, for all $d \in \mathcal{T}_{\Omega_1}(\bar{z})$, we have

$$\begin{aligned}(\mathcal{F} + \mathcal{R})^\circ(\bar{z}; d) = (\mathcal{F} + \mathcal{R})^\circ(\bar{z}; d) + \mathcal{P}'(\bar{z}; d) = (\mathcal{F} + \mathcal{R})^\circ(\bar{z}; d) + \mathcal{P}^\circ(\bar{z}; d)\\ \geq \mathcal{O}^\circ(\bar{z}; d) \geq 0,\end{aligned}$$

where the first equality comes from $\mathcal{P}'(\bar{z}, d) = 0$ (see the last part of the proof of Theorem 2.5), the second equality comes from $\mathcal{P}$ being regular at $\bar{z}$, and the last inequality comes from $d \in \mathcal{T}_{\Omega_1}(\bar{z}) \subset \mathcal{T}_{\Omega_2}(\bar{z})$. Hence, $\bar{z}$ is a generalized d-stationary point of (R). □

We end this section by summarizing our results for the relationship of problems (R), (RP) and (LRP) with $\beta > L_{\mathcal{F}} + L_{\mathcal{R}}$ in the following diagram, where $\bar{z} \in \Omega_\theta$, $L_{\mathcal{F}}$ and $L_{\mathcal{R}}$ are the Lipschitz modulus of $\mathcal{F}$ and $\mathcal{R}$ over Ω_δ, respectively.

R :	global minimizer	local minimizer	d-stationary point	generalized d-stationary point
	$\Downarrow\Uparrow$	$\Downarrow$	$\Uparrow$	$\Uparrow$ $\mathcal{P}$ is regular at $\bar{z}$
RP :	global minimizer	local minimizer	d-stationary point	generalized d-stationary point
	$\bar{z} \in \Omega_3$ $\Downarrow\Uparrow$	$\bar{z} \in \Omega_3$ $\Downarrow\Uparrow$	$\Uparrow$ $\mathcal{O}(\bar{z}) < \theta$	$\Uparrow$ $\bar{z} \in \operatorname{int}(\Omega_3)$, $\mathcal{O}(\bar{z}) < \theta$
LRP :	global minimizer	local minimizer	d-stationary point	generalized d-stationary point

3. A Smoothing Proximal Gradient Algorithm.

In this section, we propose a smoothing proximal gradient algorithm (SPG) for solving problem (LRP). The proposed SPG introduces a smoothing function of the objective function of (LRP) and solves a strongly convex quadratic program over its feasible set $\mathcal{Z} = \{z : Az \leq c\}$ at each iteration. In the rest of this section, we first present the algorithm framework and then establish convergence results of the algorithm.

3.1. Algorithm Framework.

Definition 3.1. *[6] Let $f : \mathbb{R}^m \mapsto \mathbb{R}$ be a continuous function. We call $\widetilde{f} : \mathbb{R}^m \times \mathbb{R}_+ \mapsto \mathbb{R}$ a smoothing function of f, if for all fixed $\mu > 0$, $\widetilde{f}(\cdot, \mu)$ is continuously differentiable, and $\lim_{y \to \bar{y}, \mu \downarrow 0} \widetilde{f}(y, \mu) = f(\bar{y})$.*

In this paper, we adopt the following smoothing function $\widetilde{\sigma}(y, \mu) : \mathbb{R}^m \times \mathbb{R}_+ \mapsto \mathbb{R}^m$ for the ReLU activation function $\sigma = (y)_+$ as follows.

$$\widetilde{\sigma}_i(y, \mu) = \begin{cases} 0 & \text{if } y_i < 0, \\ \dfrac{y_i^2}{2\mu} & \text{if } 0 \leq y_i \leq \mu, \\ y_i - \dfrac{\mu}{2} & \text{if } y_i > \mu \end{cases}$$

for all $i = 1, \ldots, m$, where y_i is the i-th element of $y \in \mathbb{R}^m$. Then, we obtain that $\nabla \widetilde{\sigma}_i(y, \mu) = \min\left\{\max\left\{\frac{y_i}{\mu}, 0\right\}, 1\right\}$, and $\widetilde{\sigma}(y, \mu_1) < \widetilde{\sigma}(y, \mu_2)$ with $\mu_1 > \mu_2$.

We construct a smoothing function of $\mathcal{O}(z)$ over $\mathcal{Z}$ for $\mu > 0$,

$$\widetilde{\mathcal{O}}(z,\mu) : = \widetilde{\mathcal{H}}(z,\mu) + \mathcal{R}(z), \tag{3.1}$$

where $\widetilde{\mathcal{H}}(z,\mu) := \widetilde{\mathcal{F}}(z,\mu) + \widetilde{P}(z,\mu)$, and

$$\widetilde{\mathcal{F}}(z,\mu) = \frac{1}{N}\sum_{n=1}^{N} \|(W^\top v_n + b_2)_+\|_2^2 + \frac{1}{N}\|X\|_F^2 - \frac{2}{N}\sum_{n=1}^{N} x_n^\top \widetilde{\sigma}(W^\top v_n + b_2, \mu),$$

$$\widetilde{P}(z,\mu) = \beta \sum_{n=1}^{N} e^\top \left(v_n - \widetilde{\sigma}(W x_n + b_1, \mu)\right)$$

are the smoothing functions of $\mathcal{F}(z)$ and $\mathcal{P}(z)$, respectively. Here we use the smoothness of $\sum_{n=1}^{N} \|(W^\top v_n + b_2)_+\|_2^2$. It is clear that $\widetilde{\sigma}(W x_n + b_1, \mu) \leq (W x_n + b_1)_+$ and $\widetilde{\mathcal{O}}(z,\mu_1) > \widetilde{\mathcal{O}}(z,\mu_2)$ for $\mu_1 > \mu_2$ and $z \in \mathcal{Z}$. In addition, for all $z \in \mathcal{Z}$ and $\mu > 0$, we have

$$0 \leq \mathcal{O}(z) \leq \widetilde{\mathcal{O}}(z,\mu) \leq \mathcal{O}(z) + (\|X\|_1 + N_1 N \beta)\mu. \tag{3.2}$$

The function $\mathcal{R}$ is a convex quadratic function and the eigenvalues of the Hessian matrix of $\mathcal{R}$ are in $\{0, 2\lambda_2\}$. It is clear that $\widetilde{\mathcal{P}}(\cdot,\mu)$, $\widetilde{\mathcal{F}}(\cdot,\mu)$, $\nabla_z \widetilde{\mathcal{P}}(\cdot,\mu)$ and $\nabla_z \widetilde{\mathcal{F}}(\cdot,\mu)$ are locally Lipschitz continuous for any fixed $\mu > 0$. Moreover, $\mu\widetilde{\mathcal{P}}$, $\mu\widetilde{\mathcal{F}}$, $\mu\nabla_z\widetilde{\mathcal{P}}$ and $\mu\nabla_z\widetilde{\mathcal{F}}$ are piecewise quadratic functions with respect to μ.

By the proof of Theorem 2.2, the set $\Omega_\theta \cap \mathcal{Z}$ is bounded and $\|z\|_\infty \leq \max\{\alpha, \eta\}$ holds for any $z \in \Omega_\theta \cap \mathcal{Z}$, where $\eta := \max\{\sqrt{\frac{N_1 N_0 \theta}{\lambda_2}}, \frac{\theta}{\lambda_1}\}$. Let $L_{\widetilde{\mathcal{H}}}$ and $L_{\nabla\widetilde{\mathcal{H}}}$ be Lipschitz modulus of $\mu\widetilde{\mathcal{H}}$ over $\Omega_\theta \cap \mathcal{Z} \times (0,1)$, and $\mu\nabla_z\widetilde{\mathcal{H}}$ over $\{z : \|z\|_\infty \leq \max\{\alpha, 2\eta\}\} \times (0,1)$, respectively.

Our smoothing proximal gradient algorithm is presented in Algorithm 3.1.

Algorithm 3.1 A smoothing proximal gradient algorithm (SPG)

```
Initialization: choose $z^{(0)} \in \mathcal{Z}$, $0 < \mu^{(0)} < 1$, $0 < \tau_1 < 1$, $\tau_2 > 0$, $\tau_3 \geq 1$, and
    $L^{(0)} \geq 1$. Set $k := 0$.
while a termination criterion is not met, do
    Set $z^{(k+1)}$ be the unique minimizer of the strongly convex quadratic program
        (3.3)    $\min_{z \in \mathcal{Z}} \left\langle \nabla_z \widetilde{\mathcal{H}}(z^{(k)}, \mu^{(k)}), z - z^{(k)} \right\rangle + \mathcal{R}(z) + \frac{L^{(k)}}{2}\|z - z^{(k)}\|_2^2.$
    Update the smoothing and proximal parameters $\mu^{(k+1)}$ and $L^{(k+1)}$ by
    (3.4)
    $\begin{cases} (\mu^{(k+1)}, L^{(k+1)}) := (\mu^{(k)}, L^{(k)}), \text{ if } \widetilde{\mathcal{O}}(z^{(k+1)}, \mu^{(k)}) - \widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)}) < -\tau_2 \frac{\mu^{(k)}}{L^{(k)}}, \\ (\mu^{(k+1)}, L^{(k+1)}) := (\tau_1 \mu^{(k)}, \tau_3 L^{(k)}), \text{ otherwise.} \end{cases}$
    Set $k := k + 1$.
end while
```

3.2. Convergence Analysis. The following lemma will be used for the convergence results of SPG.

LEMMA 3.2. *Let $\{z^{(k)}\}$ and $\{\mu^{(k)}\}$ be the sequences generated by Algorithm 3.1 with $\mathcal{O}(z^{(0)}) < \theta$, $\tau_1\tau_3 \geq 1$ and $\mu^{(0)}L^{(0)}$ satisfying*

$$\mu^{(0)}L^{(0)} \geq \max\left\{6\lambda_2 N_1 N_0 + \frac{2}{\eta}(N_2 L_{\widetilde{\mathcal{H}}} + \lambda_1 N_1 N),\ 8\lambda_2 + L_{\nabla\widetilde{\mathcal{H}}}\right\}. \tag{3.5}$$

Then, the following statements hold.

(a) The sequence $\left\{\widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)})\right\}$ is non-increasing, and $\{z^{(k)}\} \subset \Omega_\theta \cap \mathcal{Z}$;

(b) If $\widetilde{\mathcal{O}}(z^{(k+1)}, \mu^{(k)}) - \widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)}) \geq -\tau_2\frac{\mu^{(k)}}{L^{(k)}}$, then there exists a nonnegative vector $\gamma^{(k+1)} \in \mathbb{R}^\nu$ such that

$$\begin{aligned}
&\left\|\nabla_z\widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)}) + A^\top\gamma^{(k+1)}\right\|_2 \leq 2\sqrt{\tau_2}(\mu^{(k)})^{1/2}, \text{ and}\\
&Az^{(k)} \leq c,\ \ -\tau_2\frac{(\mu^{(k)})^2}{8\lambda_2 + L_{\nabla\widetilde{\mathcal{H}}}} \leq (\gamma^{(k+1)})^\top(Az^{(k)} - c) \leq 0.
\end{aligned} \tag{3.6}$$

The proof is given in Appendix A. It is worth mentioning that $z^{(0)} = 0$ satisfies the condition on the initial guess in Lemma 3.2. Now we present our main convergence theorem as follows.

THEOREM 3.3. *Under assumptions of Lemma 3.2, the following statements hold.*
(a) $\lim_{k\to\infty}\mu^{(k)} = 0$;
(b) $\{\mathcal{O}(z^{(k)})\}$ and $\{\widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)})\}$ are convergent. Moreover, we have

$$\lim_{k\to\infty}\widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)}) = \lim_{k\to\infty}\mathcal{O}(z^{(k)}). \tag{3.7}$$

Proof. (a) Assume on contradiction that there exists $k_0 > 0$ such that whenever $k \geq k_0$, $\widetilde{\mathcal{O}}(z^{(k+1)}, \mu^{(k)}) - \widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)}) < -\tau_2\frac{\mu^{(k)}}{L^{(k)}}$ holds. Hence, it follows from the updating formula (3.4) of μ and L that $\mu^{(k)} = \mu^{(k_0)}$ and $L^{(k)} = L^{(k_0)}$ for all $k \geq k_0$. Together with the inequality $\widetilde{\mathcal{O}}(z^{(k+1)}, \mu^{(k)}) \geq \widetilde{\mathcal{O}}(z^{(k+1)}, \mu^{(k+1)})$, we have

$$\widetilde{\mathcal{O}}(z^{(k+1)}, \mu^{(k+1)}) - \widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)}) < -\tau_2\frac{\mu^{(k)}}{L^{(k)}} = -\tau_2\frac{\mu^{(k_0)}}{L^{(k_0)}} \tag{3.8}$$

for all $k \geq k_0$.

Denote $l = \lceil\frac{L^{(k_0)}}{\tau_2\mu^{(k_0)}}\widetilde{\mathcal{O}}(z^{(k_0)}, \mu^{(k_0)})\rceil$. It then follows from (3.2) and (3.8) that

$$\begin{aligned}
&-\widetilde{\mathcal{O}}(z^{(k_0)}, \mu^{(k_0)}) \leq \widetilde{\mathcal{O}}(z^{(k_0+l)}, \mu^{(k_0+l)}) - \widetilde{\mathcal{O}}(z^{(k_0)}, \mu^{(k_0)})\\
=&\sum_{k=k_0}^{k_0+l-1}\left(\widetilde{\mathcal{O}}(z^{(k+1)}, \mu^{(k+1)}) - \widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)})\right) < -\tau_2\frac{\mu^{(k_0)}}{L^{(k_0)}}l \leq -\widetilde{\mathcal{O}}(z^{(k_0)}, \mu^{(k_0)}),
\end{aligned}$$

which leads to a contradiction. Hence, the second situation in (3.4) happens infinite times and we prove the assertion (a).

(b) Recall the nonnegativity of $\widetilde{\mathcal{O}}(z, \mu)$ and the monotonical non-increasing of $\left\{\widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)})\right\}$, we can conclude that $\{\widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)})\}$ is convergent. Together with $\lim_{k\to\infty}\mu^{(k)} = 0$, we have $\lim_{k\to\infty}\widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)}) = \lim_{k\to\infty}\mathcal{O}(z^{(k)})$, which completes the proof. ∎

THEOREM 3.4. *Suppose the assumptions of Lemma 3.2 hold. Let $\mathcal{K} = \{k : \mu^{(k+1)} = \tau_1\mu^{(k)}, k \geq 0\}$. Then $\{z^{(k)} : k \in \mathcal{K}\}$ is bounded and any accumulation point z^* of $\{z^{(k)} : k \in \mathcal{K}\}$ is a generalized d-stationary point of (LRP). Moreover, if $\mathcal{O}$ is regular at z^*, then z^* is a d-stationary point of (LRP).*

Proof. Let $\{\gamma^{(k+1)}\} \subset \mathbb{R}^{\nu}_{+}$ satisfy (3.6) for $k \in \mathcal{K}$. By the structure of the matrix A and Lipschitz continuity of $\mathcal{O}$, $\{\gamma^{(k+1)}\}$ is bounded. Due to the fact that $\lim_{k\to\infty}\mu^{(k)} = 0$, it holds that $\mathcal{K}$ has infinitely many elements. Since z^* is an accumulation point of $\{z^{(k)} : k \in \mathcal{K}\}$, there exist subsequences $\{z^{(j_k)}\}$ of $\{z^{(k)} : k \in \mathcal{K}\}$ and $\{\gamma^{(j_k+1)}\}$ of $\{\gamma^{(j_k+1)} : k \in \mathcal{K}\}$ such that $\lim_{k\to\infty} z^{(j_k)} = z^*$ and $\lim_{k\to\infty}\gamma^{(j_k+1)} = \gamma^*$. By taking k from the both sides of (3.6) to infinity, we obtain that

$$0 = \liminf_{k\in\mathcal{K},k\to\infty}\left\|\nabla_z\widetilde{\mathcal{O}}(z^{(k)},\mu^{(k)}) + A^\top\gamma^{(k+1)}\right\|, \quad (Az^* - c)^\top\gamma^* = 0, \quad Az^* - c \leq 0. \tag{3.9}$$

Since $\mathcal{O}(z)$ is a finite-sum composite max function, we have the gradient sub-consistency [4, 6] that

$$\text{co}\left\{\lim_{k\to\infty}\nabla_z\widetilde{\mathcal{O}}(z^{(j_k)},\mu^{(j_k)})\right\} \subset \partial\mathcal{O}(z^*). \tag{3.10}$$

Hence, we have $0 \in \partial Q(z^*) + A^\top\gamma^*$ and hence z^* is a generalized KKT point of (LRP).

It follows Theorem 2.6 that z^* is a generalized d-stationary point of (LRP).

If $\mathcal{O}$ is regular at z^*, then $\mathcal{O}^\circ(z^*;d) = \mathcal{O}'(z^*;d)$ for all $d \in \mathcal{T}_{\mathcal{Z}}$. In this case z^* is a d-stationary point of (LRP). □

4. Numerical Experiments.

In this section, we evaluate the numerical performance of our proposed SPG method for training the autoencoders. We first introduce the implementation details including the default settings for the parameters and the test problems. Then we report the numerical comparison among our SPG with a few state-of-the-art SGD based approaches, including the Adam [20], the Adamax [20], the Adadelata [38], the Adagrad [12], the AdagradDecay [12], and the Vanilla SGD [9]. Our numerical experiments will use both synthetic datasets and real dataset. All the numerical experiments in this section are performed on a workstation with one Intel (R) Xeon (R) Silver 4110CPU (at 2.10GHz × 32) and 384GB of RAM running MATLAB R2018b under Ubuntu 18.10.

4.1. Implementation Details.

We first introduce the model parameters involved in (LRP). We set the regularization parameter $\lambda_2 = 0.1$. If the sparsity of V is pursued, we further set $\lambda_1 = 0.0001$ in (LRP). The penalty parameter β takes the constant value $\frac{1}{N}$.

We explain how to choose the algorithm parameters of SPG. The constants τ_1 and τ_2 take the values 0.5 and 0.001, respectively. The initial value of the smoothing parameter is set as $\mu^{(0)} = 0.001$. Although the global convergence of Algorithm 3.1 requires sufficiently large parameters τ_3 and $L^{(0)}$, they can take much smaller values for better performance in practice. Empirically, we set $\tau_3 = 1.1$ and $L^{(0)} = L_* := \max\{1, \sqrt{N_0N_1/N}, \beta, N_0/30\}$, unless otherwise stated.

Recall (3.6), we set $\mu^{(k)} \leq \varepsilon$ as the stopping criterion of Algorithm 3.1 and the tolerance ε takes 10^{-7} in the experiments unless otherwise stated. Besides, the maximum number of iterations in SPG is set as 4000. Next, we describe the default initial guess in the following. The matrix $W^{(0)}$ are randomly generated by $W^{(0)} = \text{randn}(N_1, N_0)/N$, where $\text{randn}(n,p)$ stands for an $n \times p$ randomly generated matrix under the standard Gaussian distribution. Then, we set $b^{(0)} = 0$ and $v_n^{(0)} = (W^{(0)}x_n)_+$ for all $n = 1, 2, \ldots, N$.

For solving the quadratic programming subproblem (3.3), we propose a new splitting and alternating method, abbreviated as SAMQP, to significantly increase the efficiency by exploiting the special structure. The details descriptions, as well as the comparison with exiting QP solvers, are put in Appendix B.

The MATLAB codes of the SGD based approaches including Adam, Adamax, Adadelata, Adagrad, AdagradDecay, and Vanilla SGD are downloaded from the Library [19]. These approaches directly solve problem (1.3). All of these algorithms are run under their defaulting settings. The batch-size is set to $\max\{N/100, 10\}$.

There are two classes of test problems. The first class of data sets are generated randomly. Let N and N_{test} be the numbers of training and test samplings, respectively. We use parameter $\epsilon_0 > 0$ to control the noise level. We construct the data sets by the following two ways, where $\text{rand}(n, p)$ stands for an $n \times p$ randomly generated matrix under uniform distribution in $[0, 1]$.

- Data type 1: we generate the data matrix $X_{\text{all}} = (x_1, x_2, \ldots, x_{N+N_{\text{test}}})$ by setting $x_i \sim \mathcal{N}(\vartheta, \Sigma_0^\top \Sigma) + \epsilon_0 \mathcal{N}(0, 1)$ for all $i = 1, 2, \ldots, N + N_{\text{test}}$, where $\vartheta = 0.5 + \text{randn}(N_0, 1)$ and $\Sigma_0 = \text{randn}(N_0, 1)$. We then set all negative elements of X_{all} to be zero. The first N and the last N_{test} columns of X_{all} are selected to be the training and test sets, respectively.
- Data type 2: we generate the data matrix X_{all} by $X_{\text{all}} = \text{rand}(N + N_{\text{test}}, N_0) + \epsilon_0 \text{randn}(N + N_{\text{test}}, N_0)$. We then set all negative elements of X_{all} to be zero. The first N and the last N_{test} columns of X_{all} are selected to be the training and test sets, respectively.

In the numerical experiments, we will frequently use the following nine combinations of (N, N_0, N_1) to determine the size of the randomly generated data sets. For convenience, we simply call these combinations "E.g. 1 to 9" as follows. Other combinations will be stated otherwise.

(1) E.g. 1: $N = 50$, $N_1 = 50$, $N_0 = 25$; (2) E.g. 2: $N = 50$, $N_1 = 100$, $N_0 = 25$;
(3) E.g. 3: $N = 50$, $N_1 = 100$, $N_0 = 40$; (4) E.g. 4: $N = 50$, $N_1 = 10$, $N_0 = 5$;
(5) E.g. 5: $N = 75$, $N_1 = 10$, $N_0 = 5$; (6) E.g. 6: $N = 100$, $N_1 = 10$, $N_0 = 5$;
(7) E.g. 7: $N = 100$, $N_1 = 100$, $N_0 = 25$; (8) E.g. 8: $N = 150$, $N_1 = 10$, $N_0 = 5$;
(9) E.g. 9: $N = 150$, $N_1 = 20$, $N_0 = 10$.

The second class of test problems are selected from the MNIST datasets [25] consisting of 10-classes handwritten digits with the size 28×28, namely, $N_0 = 784$. In practice, we randomly pick up data entries from each class of MNIST under uniform distribution.

We record the following four measurements, the function value of (LRP) ("FVal"), the average feasibility violation ("FeasVi"), the training error ("TrainErr") and the test error ("TestErr") of (R), which are denoted by $\mathcal{O}(z)$, $\frac{1}{NN_1}\sum_{n=1}^{N} \|v_n - (Wx_n + b_1)_+\|_1$, $\mathcal{F}(z)$, and $\frac{1}{N_{\text{test}}}\sum_{n=N_{\text{test}}+1}^{N+N_{\text{test}}} \left\|(W^\top v_n + b_2)_+ - x_n\right\|_2^2$, respectively. We use "Noise" and "Time" to represent the value of ϵ_0 and the CPU Time in seconds, respectively.

4.2. Properties of SPG. In this subsection, we investigate the numerical performance of SPG in solving problems with randomly generated data sets. We first study the convergence properties of SPG. The test problem is generated by data type 1 with parameter combination E.g. 6. and $\epsilon_0 = 0.05$. The penalty parameter β takes its default setting. The numerical results of SPG with randomly initial guess is present in Figure 2. We can learn from Figure 2 that (i) all of the training error, the test error, the function value of (LRP) and the smoothing function value decrease in a same order, in particular the decreasing curves of the last two values merge together very soon; (ii) the feasibility reduces to its tolerance rapidly; (iii) the smoothing parameter sequence $\{\mu^{(k)}\}$ converges almost linearly to zero.

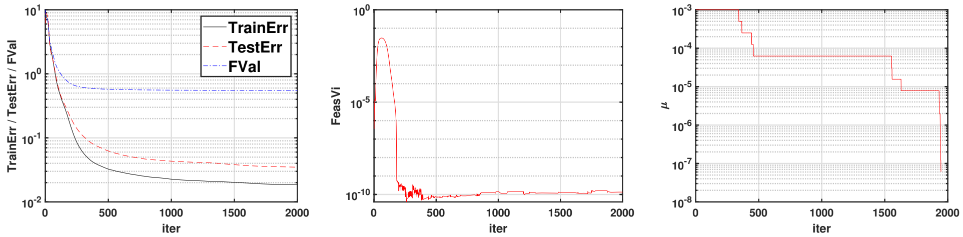


Fig. 2: Algorithm performance of SPG

Secondly, we compare SPG with different choices of $L^{(0)}$ on a group of randomly generated data sets with data type 1, $N_{\text{test}} = 0$ and $\epsilon_0 = 0.05$. We select three different β and four parameter combinations. The numerical results are shown in Figure 3. We can learn from Figure 3 that SPG may diverge if $L^{(0)}$ is not sufficiently large, particularly if β is large. When β is small, the performance of SPG is not very sensitive to the choice of $L^{(0)}$. We also find that bigger $L^{(0)}$ usually leads to slow convergence. Hence, we can conclude that a suitably selected $L^{(0)}$, such as our default setting, is important to SPG.

Finally, we compare SPG with different choices of β on a group of randomly generated data sets with $N_{\text{test}} = 0$, $\epsilon_0 = 0.05$ and data type 1. We choose β from the set $\{\frac{1}{N}, \frac{10}{N}, \frac{1}{10N}, 1, 10\}$. We record how TrainErr, the FVal and the FeasVi decrease through the iteration. The numerical results with parameter combinations E.g. 4 and E.g. 8 are illustrated in Figures 4 and 5, respectively. We can learn from these two figures that the bigger β always leads to slower convergence.

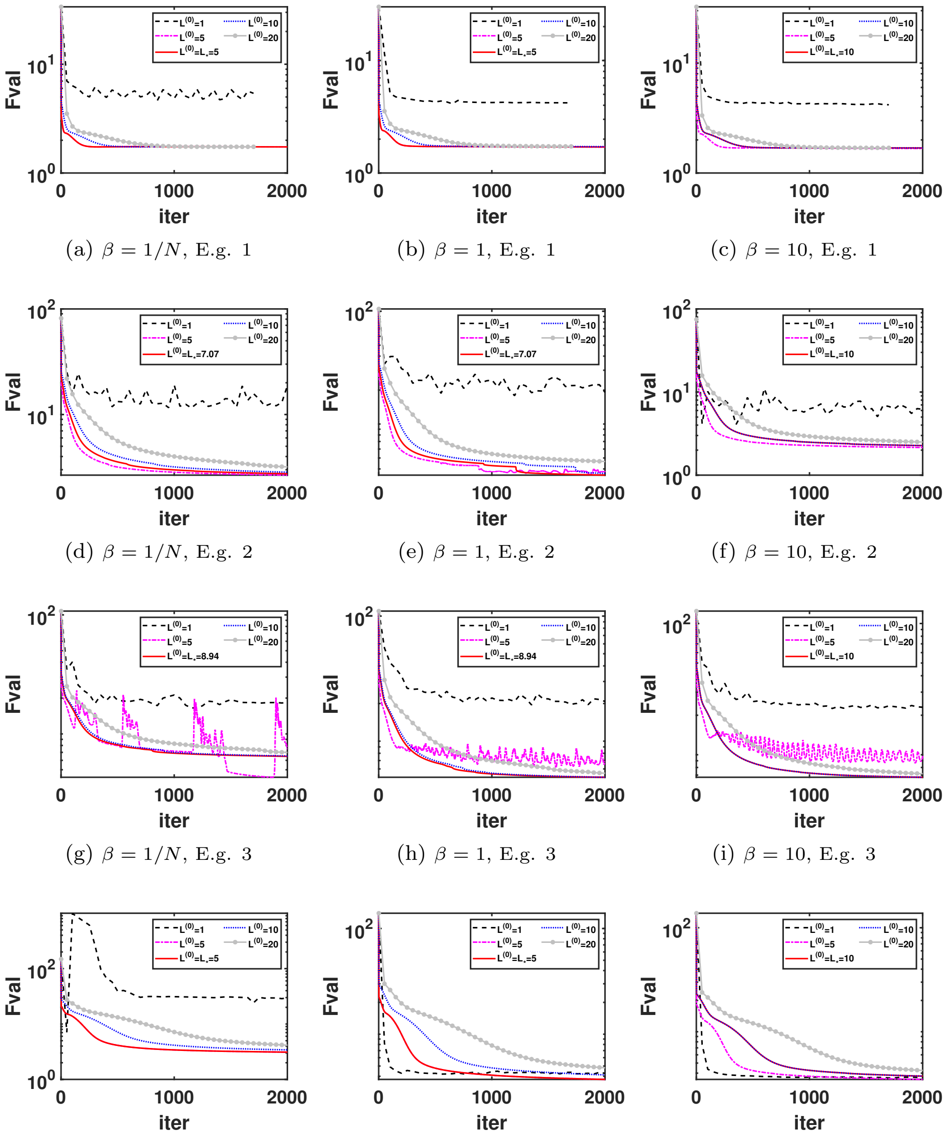
4.3. Comparison with Other Methods. In this subsection, we compare SPG with the existing SGD-based approaches.

We choose two groups of data sets randomly generated by the two data types described in Subsection 4.1, and the numerical results are demonstrated in Figures 6 and 7, respectively. Here, all algorithms start from the same random initial guess. We set $\epsilon_0 = 0.05$ and $N_{\text{test}} = 30$. The chosen parameter combinations are given in the subtitles of these two figures.

We can learn from Figures 6 and 7 that SPG reduces the training and test errors slower than some other algorithms at very beginning. It can reach a lower residual than the others finally.

It can also be observed from these two figures that Adadelta outperforms the other SGD-based approaches in the aspects of efficiency and solution quality. Therefore, we consider to use Adadelta as a pre-process to accelerate SPG. More specifically, we first run Adadelta for 1000 epochs and then switch to SPG. We call the consequent hybrid algorithm SPG-ADA. In the following tests, such pre-processing will be the default setting of SPG.

We select a new group of data sets randomly generated by data type 1 with $N = 1000$, $N_{\text{test}} = 300$, $\epsilon_0 = 0.05$, and different combinations of N_0 and N_1. The iteration number of Adadelta is set as 10000. We run SPG-ADA and Adadelta 100 times and record the average output values in Table 1. We can learn from Table 1 that SPG-ADA can obtain better training and test errors than Adadelta in comparable



(a) $\beta = 1/N$, E.g. 1 (b) $\beta = 1$, E.g. 1 (c) $\beta = 10$, E.g. 1

(d) $\beta = 1/N$, E.g. 2 (e) $\beta = 1$, E.g. 2 (f) $\beta = 10$, E.g. 2

(g) $\beta = 1/N$, E.g. 3 (h) $\beta = 1$, E.g. 3 (i) $\beta = 10$, E.g. 3

(j) $\beta = 1/N$, E.g. 7 (k) $\beta = 1$, E.g. 7 (l) $\beta = 10$, E.g. 7

Fig. 3: Comparison of SPG with varying $L^{(0)}$.

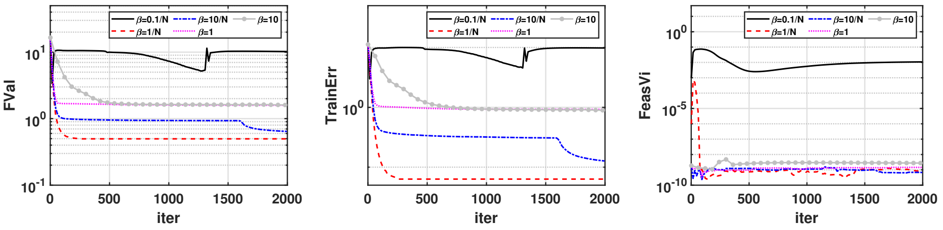


Fig. 4: Comparison of SPG with varying β, $L^{(0)} = L_\star$.

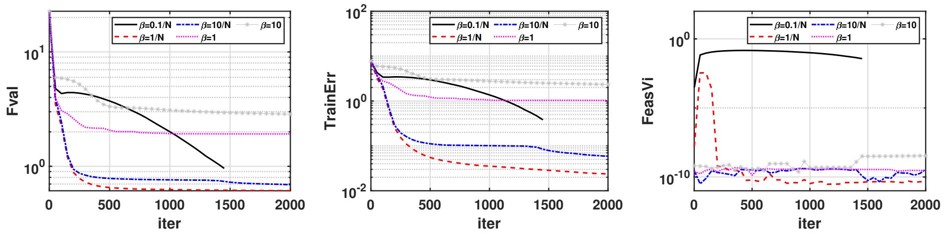
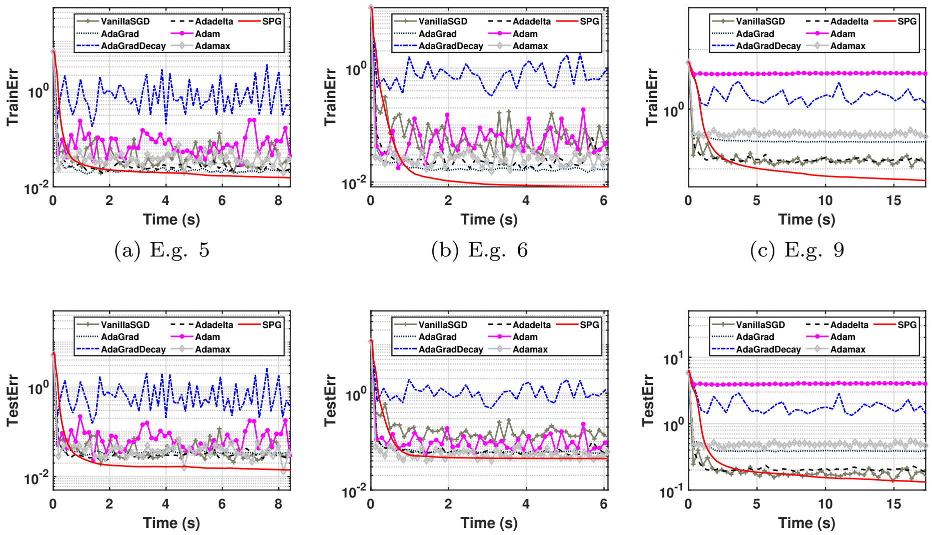


Fig. 5: Comparison of SPG with varying β, $L^{(0)} = L_{\star}$.



(a) E.g. 5

(b) E.g. 6

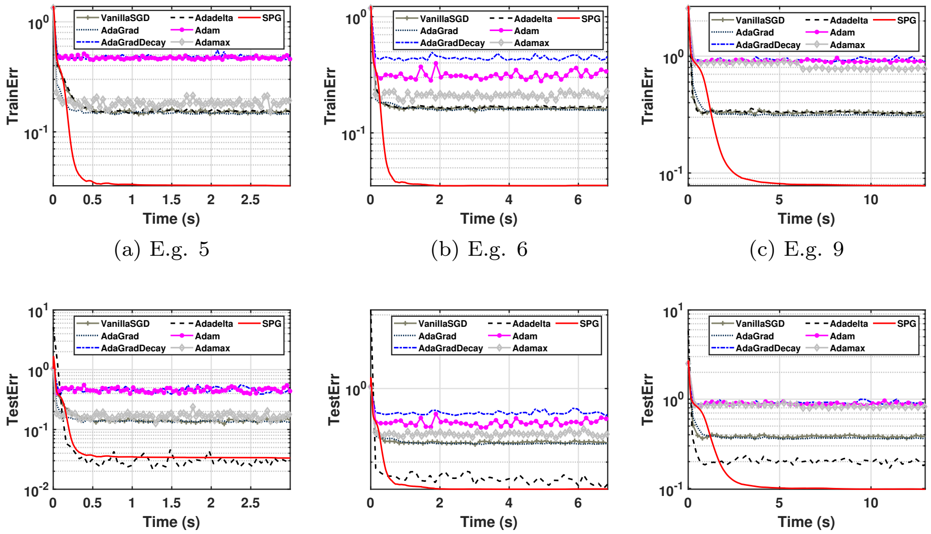
(c) E.g. 9

(d) E.g. 5

(e) E.g. 6

(f) E.g. 9

Fig. 6: Comparison among SPG and SGD-based approaches with data type 1.



(a) E.g. 5

(b) E.g. 6

(c) E.g. 9

(d) E.g. 5

(e) E.g. 6

(f) E.g. 9

Fig. 7: Comparison among SPG and SGD-based approaches with data type 2.

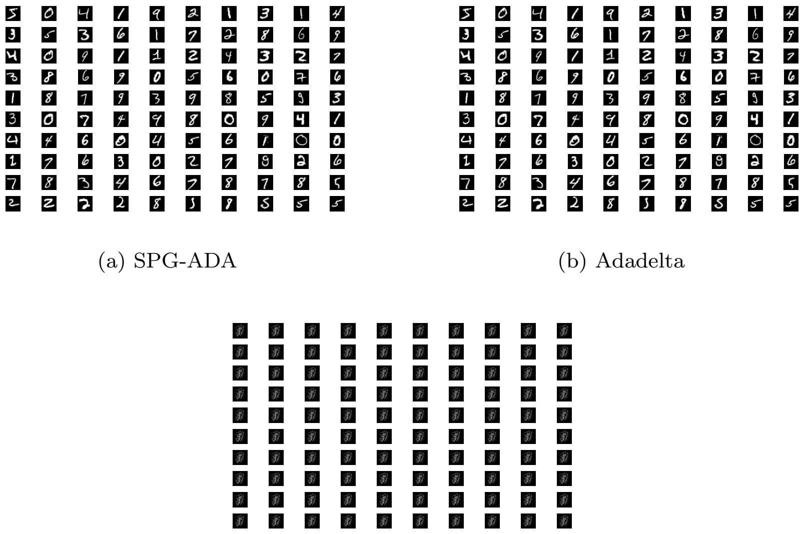
Table 1: Comparison between SPG-ADA and Adadelta with $N = 1000$.

		SPG-ADA				Adadelta		
N_0	N_1	TrainErr	TestErr	FeaErr	Time	TrainErr	TestErr	Time
5	20	3.297e-02	3.636e-02	1.234e-11	8.758	5.518e-02	5.847e-02	3.044
5	30	2.974e-02	3.103e-02	5.599e-12	10.592	5.470e-02	5.566e-02	3.623
5	40	2.960e-02	3.200e-02	7.238e-12	15.206	5.474e-02	5.632e-02	3.786
10	40	6.708e-02	7.727e-02	1.140e-11	19.184	1.257e-01	1.341e-01	5.590
10	60	6.867e-02	7.863e-02	5.138e-11	22.599	1.348e-01	1.436e-01	6.149
10	80	8.105e-02	9.057e-02	8.814e-11	25.701	1.364e-01	1.441e-01	7.169
20	80	1.824e-01	2.200e-01	3.020e-12	33.962	3.766e-01	4.265e-01	8.992
20	120	1.135e-01	2.611e-01	3.634e-12	38.191	4.051e-01	4.566e-01	12.275
20	160	1.946e-01	2.380e-01	2.181e-12	72.942	3.746e-01	4.240e-01	20.268

CPU time.

4.4. Tests on MNIST. In this subsection, we investigate the numerical comparison among SPG-ADA and Adadelta in solving problems arising from the real data set MNIST.

Firstly, we set $N = 100$ and $N_1 = 500$. We can find the reconstruction results corresponding to the autoencoder solutions obtained by SPG-ADA and Adadelta in Figure 8 (a)-(b), respectively. We can conclude that SPG-ADA can reach the comparable reconstruction quality as Adadelta. In addition, we also present the reconstruction result derived by Adam, as a failure case. Therefore, we exclude Adam in the last numerical experiment.



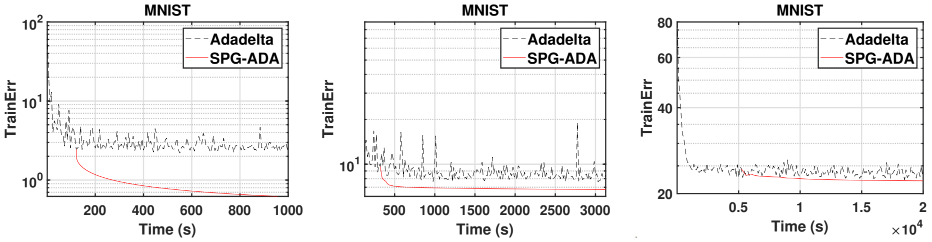
(a) SPG-ADA

(b) Adadelta

(c) Adam

Fig. 8: Reconstruction of MNIST by SPG-ADA, Adadelta and Adam.

Finally, we demonstrate how the training and test errors decrease through the iterations of SPG-ADA and Adadelta. We select different combinations of N and N_1. The results are illustrated in Figure 9. We can learn that SPG-ADA is much more robust and can always find better solutions.



(a) $N = 100$, $N_1 = 500$ (b) $N = 1000$, $N_1 = 1000$ (c) $N = 10000$, $N_1 = 2000$

Fig. 9: Comparison between SPG-ADA and Adadelta on MNIST.

5. Conclusion.

The regularized minimization model (R) using the ReLU activation function has been extensively applied for the autoencoders. However, the set of global minimizers of the model is generally unbounded. Existing algorithms cannot guarantee to generate bounded sequences with decreasing objective function values. In this paper, we propose the regularized minimization model with l_1-norm penalty (RP) that has same global minimizers, local minimizer and d-stationary points with the regularized minimization model (R). Moreover, we develop the linearly constrained regularized minimization model with l_1-norm penalty (LRP) which has a bounded solution set contained in the solution sets of (R) and (RP). We develop a smoothing proximal gradient (SPG) algorithm to solve (LRP). We prove the sequence generated by the SPG algorithm is bounded and has a subsequence converging to a generalized d-stationary point of (LRP). We conduct comprehensive numerical experiments to verify the effectiveness, efficiency and robustness of the SPG algorithm.

Finally, we mention that our results on the relationships among (R), (RP), (LRP) can be extended to the following three corresponding problems for training an L-layer DNN with ReLU activation functions, given input data $\{x_n\}_{n=1}^N$ and output data $\{y_n\}_{n=1}^N$.

$$\begin{aligned}&\min_z \mathcal{F}(z)+\mathcal{R}(z)\\ &\text{s.t. } u_{n,\ell}=(W_\ell u_{n,\ell-1}+b_\ell)_+, \quad n=1,\dots,N,\ \ell=1,\dots,L,\end{aligned} \tag{5.1}$$

$$\begin{aligned}&\min_z \mathcal{F}(z)+\mathcal{R}(z)+\beta\sum_{n=1}^N\sum_{\ell=1}^L e_{N_\ell}^\top(u_{n,\ell}-(W_\ell u_{n,\ell-1}+b_\ell)_+)\\ &\text{s.t. } u_{n,\ell}\geq(W_\ell u_{n,\ell-1}+b_\ell)_+, \quad n=1,\dots,N,\ \ell=1,\dots,L,\end{aligned} \tag{5.2}$$

and

$$\begin{aligned}&\min_z \mathcal{F}(z)+\mathcal{R}(z)+\beta\sum_{n=1}^N\sum_{\ell=1}^L e_{N_\ell}^\top(u_{n,\ell}-(W_\ell u_{n,\ell-1}+b_\ell)_+)\\ &\text{s.t. } u_{n,\ell}\geq(W_\ell u_{n,\ell-1}+b_\ell)_+,\ \|b_\ell\|_\infty\leq\alpha_\ell, \quad n=1,\dots,N,\ \ell=1,\dots,L,\end{aligned} \tag{5.3}$$

respectively, where α_ℓ is a given constant, $u_{n,0}=x_n$, $\mathcal{F}(z)=\frac{1}{N}\sum_{n=1}^N\|(W_L u_{n,L}+b_L)_+-y_n\|_2^2$, $\mathcal{R}(z)=\lambda_1\sum_{n=1}^N\sum_{\ell=1}^L e_{N_\ell}^T u_{n,l}+\lambda_2\sum_{\ell=1}^L\|W_\ell\|_F^2$,

$$z=(\mathrm{vec}(W_1)^\top,\dots,\mathrm{vec}(W_L)^\top,u_{1,1}^\top,u_{2,1}^\top,\dots,u_{N,1}^\top,u_{1,2}^\top,\dots,u_{N,L}^\top,b_1^\top,\dots,b_L^\top)^\top,$$

for all $\ell = 1, 2, \ldots, L$ and $n = 1, 2, \ldots, N$. However, the increasing number of layers results in more rapidly increasing number of variables which requires further development on the algorithm to maintain the numerical comparability to SGD-based approaches.

Acknowledgements. The authors would like to thank the reviewers for their insightful comments and efforts towards improving our manuscript.

REFERENCES

[1] A. F. Agarap, *Deep learning using rectified linear units (ReLU)*, preprint, arxiv:1803.08375, (2018).

[2] D. Boley, *Local linear convergence of the alternating direction method of multipliers on quadratic or linear programs*, SIAM J. Optim., 23 (2013), pp. 2183–2207.

[3] H. Bourlard and Y. Kamp, *Auto-association by multilayer perceptrons and singular value decomposition*, Biol. Cybern., 59 (1988), pp. 291–294.

[4] J. Burke, X. Chen, and H. Sun, *The subdifferential of measurable composite max integrands and smoothing approximation*, Math. Program., 181 (2020), pp. 229–264.

[5] M. Carreira-Perpinan and W. Wang, *Distributed optimization of deeply nested systems*, in Proceedings of the 17th International Conference on Artificial Intelligence and Statistics, 2014, pp. 10–19.

[6] X. Chen, *Smoothing methods for nonsmooth, nonconvex minimization*, Math. Program., 134 (2012), pp. 71–99.

[7] X. Chen, Z. Lu, and T. K. Pong, *Penalty methods for a class of non-Lipschitz optimization problems*, SIAM J. Optim., 26 (2016), pp. 1465–1492.

[8] F. H. Clarke, *Optimization and Nonsmooth Analysis*, vol. 5, SIAM, Philadelphia, 1990.

[9] H. Cramir, *Mathematical methods of statistics*, Princeton U. Press, Princeton, (1946), p. 500.

[10] Y. Cui, Z. He, and J.-S. Pang, *Multicomposite nonconvex optimization for training deep neural networks*, SIAM J. Optim., 30 (2020), pp. 1693–1723.

[11] G. E. Dahl, T. N. Sainath, and G. E. Hinton, *Improving deep neural networks for LVCSR using rectified linear units and dropout*, in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8609–8613.

[12] J. Duchi, E. Hazan, and Y. Singer, *Adaptive subgradient methods for online learning and stochastic optimization*, J. Mach. Learn. Res., 12 (2011), pp. 2121–2159.

[13] X. Glorot, A. Bordes, and Y. Bengio, *Deep sparse rectifier neural networks*, in Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, 2011, pp. 315–323.

[14] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1, MIT press Cambridge, 2016.

[15] M. Grant and S. Boyd, *CVX: Matlab software for disciplined convex programming, version 2.1*, 2014.

[16] G. E. Hinton and R. R. Salakhutdinov, *Reducing the dimensionality of data with neural networks*, Science, 313 (2006), pp. 504–507.

[17] A. J. Hoffman, *On approximate solutions of systems of linear inequalities*, in Selected Papers of Alan J Hoffman: With Commentary, World Scientific, 2003, pp. 174–176.

[18] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, *What is the best multi-stage architecture for object recognition?*, in 2009 IEEE 12th international conference on computer vision, 2009, pp. 2146–2153.

[19] H. Kasai, *SGDLibrary: A MATLAB library for stochastic optimization algorithms*, J. Mach. Learn. Res., 18 (2018), pp. 1–5.

[20] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, preprint, arxiv:1412.6980, (2014).

[21] A. Krogh and J. A. Hertz, *A simple weight decay can improve generalization*, in Advances in Neural Information Processing Systems, 1992, pp. 950–957.

[22] T. T.-K. Lau, J. Zeng, B. Wu, and Y. Yao, *A proximal block coordinate descent algorithm for deep neural network training*, preprint, arXiv:1803.09082, (2018).

[23] Y. Le Cun and F. Fogelman-Soulié, *Modèles connexionnistes de l'apprentissage*, Intellectica, 2 (1987), pp. 114–143.

[24] Y. LeCun, *The MNIST database of handwritten digits*, (1998).

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86, pp. 2278–2324.

[26] P. Li and P.-M. Nguyen, *On random deep weight-tied autoencoders: Exact asymptotic analysis, phase transitions, and implications to training*, in Proceedings of the International Conference on Learning Representations, 2019.

[27] A. L. Maas, A. Y. Hannun, and A. Y. Ng, *Rectifier nonlinearities improve neural network acoustic models*, in Proceedings of the 30-th International Conference on Machine Learning, vol. 30, 2013, p. 3.

[28] W. S. McCulloch and W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, The Bulletin of Mathematical Biophysics, 5 (1943), pp. 115–133.

[29] R. Mifflin, *Semismooth and semiconvex functions in constrained optimization*, SIAM J. Control Optim., 15 (1977), pp. 959–972.

[30] V. Nair and G. E. Hinton, *Rectified linear units improve restricted boltzmann machines*, in Proceedings of the 27-th International Conference on Machine Learning, 2010.

[31] A. Ng, *Sparse autoencoder*, CS294A Lecture Notes, 72 (2011), pp. 1–19.

[32] L. Pasa and A. Sperduti, *Pre-training of recurrent neural networks via linear autoencoders*, in Advances in Neural Information Processing Systems, 2014, pp. 3572–3580.

[33] R. Sun, *Optimization for deep learning: theory and algorithms*, preprint, arxiv:1912.08957, (2019).

[34] G. Taylor, R. Burmeister, Z. Xu, B. Singh, A. Patel, and T. Goldstein, *Training neural networks without gradients: A scalable admm approach*, in Proceedings of the 33rd International Conference on Machine Learning, 2016, pp. 2722–2731.

[35] M. A. Weingessel, *The quadprog package*, (2007).

[36] L. Wen, L. Gao, and X. Li, *A new deep transfer learning based on sparse auto-encoder for fault diagnosis*, IEEE Transactions on systems, man, and cybernetics: systems, 49 (2017), pp. 136–144.

[37] Y. Xu, X. Liu, X. Cao, and et al., *Artificial intelligence: A powerful paradigm for scientific research*, The Innovation, 2 (2021), p. 100179.

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

[39] J. Zeng, T. T.-K. Lau, S. Lin, and Y. Yao, *Global convergence of block coordinate descent in deep learning*, in Proceedings of the 36th International Conference on Machine Learning, 2019, pp. 7313–7323.

Appendix A. Proof of Lemma 3.2.

Proof. (a) It follows from the updating formula (3.4) of μ and L, the required relations (3.5) and $\tau_1\tau_3 \geq 1$ that

$$\mu^{(k)}L^{(k)} \geq \mu^{(0)}L^{(0)} \geq \max\left\{6\lambda_2 N_1 N_0 + \frac{2}{\eta}(N_2 L_{\widetilde{\mathcal{H}}} + \lambda_1 N_1 N), 8\lambda_2 + L_{\nabla\widetilde{\mathcal{H}}}\right\} \tag{A.1}$$

holds for all $k = 0, 1, \ldots$.

Next, we use the mathematical induction to prove the facts that $\{z^{(k)}\} \subset \Omega_\theta \cap \mathcal{Z}$ and $\{\widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)})\}$ is non-increasing. Naturally, we have $z^{(0)} \in \Omega_\theta \cap \mathcal{Z}$. Then we suppose that $z^{(k)} \in \Omega_\theta \cap \mathcal{Z}$ and $\widetilde{\mathcal{O}}(z^{(l)}, \mu^{(l)}) \leq \widetilde{\mathcal{O}}(z^{(l-1)}, \mu^{(l-1)})$ hold for all $l = 1, 2, \ldots, k$.

We deduce from $z^{(k)} \in \Omega_\theta \cap \mathcal{Z}$ and the proof of Theorem 2.2 that $\|z^{(k)}\|_\infty \leq \max\{\alpha, \eta\}$. If $\|z^{(k+1)}\|_\infty > \max\{\alpha, 2\eta\}$, we immediately have $\bar{\eta} := \|z^{(k+1)} - z^{(k)}\|_\infty > \eta$. Then, it holds that

$$\begin{aligned}
&\left\langle \nabla_z \widetilde{\mathcal{H}}(z^{(k)}, \mu^{(k)}), \mu^{(k)}), z^{(k+1)} - z^{(k)} \right\rangle + \mathcal{R}(z^{(k+1)}) - \mathcal{R}(z^{(k)}) + \frac{L^{(k)}}{2}\|z^{(k+1)} - z^{(k)}\|_2^2 \\
\geq& - N_2 \frac{L_{\widetilde{\mathcal{H}}}}{\mu^{(k)}}\bar{\eta} + \frac{L^{(k)}}{2}\bar{\eta}^2 + \mathcal{R}(z^{(k+1)}) - \mathcal{R}(z^{(k)}) \\
\geq& - N_2 \frac{L_{\widetilde{\mathcal{H}}}}{\mu^{(k)}}\bar{\eta} + \frac{L^{(k)}}{2}\bar{\eta}^2 - \lambda_1 N_1 N \bar{\eta} - \lambda_2 N_0 N_1 \bar{\eta} \max_{j \in \{1,2,\ldots,N_1\}} \|W_{\cdot,j}^{(k+1)} + W_{\cdot,j}^{(k)}\|_\infty \\
\geq& - N_2 \frac{L_{\widetilde{\mathcal{H}}}}{\mu^{(k)}}\bar{\eta} + \frac{L^{(k)}}{2}\bar{\eta}^2 - \lambda_1 N_1 N \bar{\eta} - \lambda_2 N_0 N_1 \bar{\eta}\,(\bar{\eta} + 2\eta) > 0,
\end{aligned}$$

where the second inequality comes from the definition of $\mathcal{R}$, the third inequality results from the relations $\|W_{\cdot,j}^{(k+1)} + W_{\cdot,j}^{(k)}\|_\infty \leq \|W_{\cdot,j}^{(k+1)} - W_{\cdot,j}^{(k)}\|_\infty + 2\|W_{\cdot,j}^{(k)}\|_\infty \leq \bar{\eta} + 2\eta$ for all $j = 1, 2, \ldots, N_1$, and the last inequality comes from $0 < \mu^{(k)} < 1$ and (A.1). This leads to a contradiction, since $z^{(k+1)}$ is a solution of subproblem (3.3). Hence, we have $\|z^{(k+1)}\|_\infty \le \max\{\alpha, 2\eta\}$.

By the KKT condition of (3.3), there exists a nonnegative vector $\gamma^{(k+1)} \in \mathbb{R}^\nu$ such that

$$\begin{cases} \nabla_z \widetilde{\mathcal{H}}(z^{(k)}, \mu) + \nabla \mathcal{R}(z^{(k+1)}) + A^\top \gamma^{(k+1)} + L^{(k)}(z^{(k+1)} - z^{(k)}) = 0, \\ Az^{(k+1)} \le c, (\gamma^{(k+1)})^\top (Az^{(k+1)} - c) = 0. \end{cases} \tag{A.2}$$

It follows from the inequality (A.1), the relations (A.2), $(\gamma^{(k+1)})^\top(Az^{(k)} - c) \le 0$, $0 < \mu^{(k)} \le 1$ and the definition of $L_{\nabla\widetilde{\mathcal{H}}}$ that

(A.3)

$$\begin{aligned} &\widetilde{\mathcal{O}}(z^{(k+1)}, \mu^{(k)}) - \widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)}) \\ \le& \left\langle \nabla_z \widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)}), z^{(k+1)} - z^{(k)} \right\rangle + \frac{2\lambda_2 + L_{\nabla\widetilde{\mathcal{H}}}}{2\mu^{(k)}} \|z^{(k+1)} - z^{(k)}\|_2^2 \\ =& \left\langle \nabla \mathcal{R}(z^{(k)}) - \nabla \mathcal{R}(z^{(k+1)}) - A^\top \gamma^{(k+1)} - L^{(k)}(z^{(k+1)} - z^{(k)}), z^{(k+1)} - z^{(k)} \right\rangle \\ &+ \frac{2\lambda_2 + L_{\nabla\widetilde{\mathcal{H}}}}{2\mu^{(k)}} \|z^{(k+1)} - z^{(k)}\|_2^2 \\ \le& \frac{2\lambda_2 + L_{\nabla\widetilde{\mathcal{H}}} - 2\mu^{(k)}L^{(k)}}{2\mu^{(k)}} \|z^{(k+1)} - z^{(k)}\|_2^2 - \left\langle A^\top \gamma^{(k+1)}, z^{(k+1)} - z^{(k)} \right\rangle + 2\lambda_2 \|z^{(k+1)} - z^{(k)}\|_2^2 \\ \le& \frac{6\lambda_2 + L_{\nabla\widetilde{\mathcal{H}}} - 2\mu^{(k)}L^{(k)}}{2\mu^{(k)}} \|z^{(k+1)} - z^{(k)}\|_2^2 + (\gamma^{(k+1)})^\top (Az^{(k)} - c) \le 0. \end{aligned}$$

Due to the nondecreasing property of $\{\widetilde{\mathcal{O}}(z^{(k+1)}, \mu)\}$ with respect to smoothing parameter μ, we have $\widetilde{\mathcal{O}}(z^{(k+1)}, \mu^{(k+1)}) \le \widetilde{\mathcal{O}}(z^{(k+1)}, \mu^{(k)})$ with $\mu^{(k+1)} \le \mu^{(k)}$. Together with the relations (3.2) and (A.3), we arrive at $\widetilde{\mathcal{O}}(z^{(k+1)}, \mu^{(k+1)}) \le \widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)})$ and $\mathcal{O}(z^{(k+1)}) < \theta$. Besides, it follows from the definition of $z^{(k+1)}$ that $z^{(k+1)} \in \mathcal{Z}$. Hence, we have $z^{(k+1)} \in \Omega_\theta \cap \mathcal{Z}$ and $\widetilde{\mathcal{O}}(z^{(l)}, \mu^{(l)}) \le \widetilde{\mathcal{O}}(z^{(l-1)}, \mu^{(l-1)})$ hold for all $l = 1, 2, \ldots, k+1$. This completes the part (a) by mathematical induction.

(b) By what we have proved in (a) and the inequality $\widetilde{\mathcal{O}}(z^{(k+1)}, \mu^{(k)}) - \widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)}) \ge -\tau_2 \frac{\mu^{(k)}}{L^{(k)}}$, we can obtain that

$$-\tau_2 \frac{\mu^{(k)}}{L^{(k)}} \le \frac{6\lambda_2 + L_{\nabla\widetilde{\mathcal{H}}} - 2\mu^{(k)}L^{(k)}}{2\mu^{(k)}} \|z^{(k+1)} - z^{(k)}\|_2^2 + (\gamma^{(k+1)})^\top (Az^{(k)} - c).$$

which together with the relations $(\gamma^{(k+1)})^\top(Az^{(k)} - c) \le 0$ and $\mu^{(k)}L^{(k)} > 8\lambda_2 + L_{\nabla\widetilde{\mathcal{H}}}$ further implies that

$$\|z^{(k+1)} - z^{(k)}\|_2^2 \le \frac{2\tau_2 (\mu^{(k)})^2}{L^{(k)}(2\mu^{(k)}L^{(k)} - 6\lambda_2 - L_{\nabla\widetilde{\mathcal{H}}})}, \tag{A.4}$$

$$-\tau_2 \frac{(\mu^{(k)})^2}{8\lambda_2 + L_{\nabla\widetilde{\mathcal{H}}}} \le (\gamma^{(k+1)})^\top (Az^{(k)} - c) \le 0. \tag{A.5}$$

It follows from the inequality (A.4) and the KKT condition (A.2) that

$$\begin{aligned} &\left\| \nabla \widetilde{\mathcal{O}}(z^{(k)}, \mu^{(k)}) + A^\top \gamma^{(k+1)} \right\|_2 = \left\| \nabla \mathcal{R}(z^{(k)}) - \nabla \mathcal{R}(z^{(k+1)}) - L^{(k)}(z^{(k+1)} - z^{(k)}) \right\|_2 \\ \le& \left\| \nabla \mathcal{R}(z^{(k)}) - \nabla \mathcal{R}(z^{(k+1)}) \right\|_2 + L^{(k)} \left\| z^{(k+1)} - z^{(k)} \right\|_2 \le (2\lambda_2 + L^{(k)}) \left\| z^{(k+1)} - z^{(k)} \right\|_2 \\ \le& \sqrt{\frac{2\tau_2}{L^{(k)}(2\mu^{(k)}L^{(k)} - 6\lambda_2 - L_{\nabla\widetilde{\mathcal{H}}})}} \mu^{(k)}(2\lambda_2 + L^{(k)}) \le 2\sqrt{\tau_2} (\mu^{(k)})^{1/2}, \end{aligned}$$

where the last inequality results from the inequalities $0 < \mu^{(k)} < 1$, $\mu^{(k)}L^{(k)} > 8\lambda_2 + L_{\nabla\widetilde{\mathcal{H}}}$ and $L^{(k)} > 2\lambda_2$. Together with (A.5), we can conclude the proof. ∎

Appendix B. A Structured Algorithm for Solving (3.3).

We notice that the subproblem (3.3) is a convex quadratic programming (QP), which can be solved by any QP solvers such as 'quadprog' [35], the default QP solver in MATLAB, and 'CVX' [15]. Since the subproblem of our SPG to be solved in autoencoder scenario is usually large-scale but structured, the existing solvers are not efficient enough. Therefore, in this subsection we propose a special algorithm for subproblem (3.3) to take the structure into account. We focus on this subproblem at the k-th iteration of SPG for any $k = 0, 1, ...$ For brevity, we will drop the superscript (k) and let $(\bar{W}, \bar{b}, \bar{V})$ to denote the current iterate $(W^{(k)}, b^{(k)}, V^{(k)})$ (and similarly for $\mu^{(k)}$ and $L^{(k)}$) in this section. In addition, we introduce a new group of variables $U = (u_1, u_2, \dots, u_N)$ subject to $u_n = Wx + b_1$ for all $n = 1, ..., N$. Hence the quadratic programming (3.3) can be reformulated as

(B.1)
$$\begin{aligned} \min_{W,b,V,U,\rho} \quad & \langle g_W, W - \bar{W} \rangle + \langle g_b, b - \bar{b} \rangle + \langle g_V, V - \bar{V} \rangle + \lambda_1 \|W\|_F^2 + \lambda_2 \sum_{n=1}^{N} e^\top v_n + \frac{L}{2} \|z - \bar{z}\|_2^2 \\ \text{s.t.} \quad & b \in \Omega_3, v_n \geq u_n,\ v_n \geq 0,\ u_n = Wx + b_1, \quad \text{for all} \ n = 1, 2, \dots, N, \end{aligned}$$

where $g_W = \nabla_W \widetilde{\mathcal{H}}(\bar{W}, \bar{b}, \bar{V}, \mu)$, $g_V = \nabla_V \widetilde{\mathcal{H}}(\bar{W}, \bar{b}, \bar{V}, \mu)$, and $g_b = \nabla_b \widetilde{\mathcal{H}}(\bar{W}, \bar{b}, \bar{V}, \mu)$ are preset constants in this subproblem.

The variables of problem (B.1) can be divided into two parts (W, b) and (V, U). We then apply the alternating direction method of multipliers (ADMM) to solve (B.1). By penalizing the equality constraints, we obtain the augmented Lagrange penalty function

$$\begin{aligned} \mathcal{G}(W, b, V, U, \rho) :=& \lambda_1 \|W\|_F^2 + \lambda_2 \sum_{n=1}^{N} e^\top v_n + \langle g_W, W - \bar{W} \rangle + \langle g_b, b - \bar{b} \rangle + \langle g_V, V - \bar{V} \rangle \\ &+ \sum_{n=1}^{N} \langle \rho_n, u_n - (Wx_n + b_1) \rangle + \frac{1}{2} \sum_{n=1}^{N} \|u_n - (Wx_n + b_1)\|_2^2 + \frac{L}{2} \|z - \bar{z}\|_2^2, \end{aligned}$$

where $\rho = (\rho_1, \rho_2, \dots, \rho_N)$ with $\rho_n \in \mathbb{R}^{N_1}$, for all $n = 1, ..., N$, are the Lagrangian multipliers associated with the equality constraints.

At the l-th iteration, we first fix $W = W^{(l)}$, $b = b^{(l)}$, $\rho = \rho^{(l)}$, and the (V, U) subproblem can be formulated as

$$\begin{aligned} &\min_{U,V} \ \mathcal{G}(W^{(l)}, b^{(l)}, V, U, \rho^{(l)}) \\ &\text{s.t. } v_n \geq u_n,\ v_n \geq 0,\ \text{for all } n = 1, 2, \dots, N. \end{aligned}$$

Due to the separability of v_n and u_n for all $n = 1, ..., N$, the (V, U) subproblem has also a closed-form solution, which is illustrated as follows,

$$\text{(B.2)} \qquad \begin{cases} V_{j,n}^{(l+1)} = -\xi_{j,n}^{1;l}, U_{j,n}^{(l+1)} = -\xi_{j,n}^{2;l}, & \text{if } \xi_{j,n}^{2;l} \geq \xi_{j,n}^{1;l} \text{ and } \xi_{j,n}^{1;l} \leq 0, \\ V_{j,n}^{(l+1)} = 0, U_{j,n}^{(l+1)} = -\xi_{j,n}^{2;l}, & \text{if } \xi_{j,n}^{2;l} \geq 0, \xi_{j,n}^{1;l} > 0, \\ V_{j,n}^{(l+1)} = U_{j,n}^{(l+1)} = 0, & \text{if } \xi_{j,n}^{2;l} < 0, \text{ and } L\xi_{j,n}^{1;l} + \xi_{j,n}^{2;l} > 0, \\ V_{j,n}^{(l+1)} = U_{j,n}^{(l+1)} = -\dfrac{L\xi_{j,n}^{1;l} + \xi_{j,n}^{2;l}}{L+1}, & \text{otherwise} \end{cases}$$

for all $j = 1, 2, \dots, N_1$ and $n = 1, 2, \dots, N$. Here, $\xi_n^{1;l} = g_{v_n}/L - \bar{v}_n + \lambda_2 e/L$, $\xi_n^{2;l} = \rho_n^{(l)} - (W^{(l)} x_n + b_1^{(l)})$, g_{v_n} is the n-th column of g_V, and $\xi_{j,n}^{1;l}$ and $\xi_{j,n}^{2;l}$ are the j-th elements of $\xi_n^{1;l}$ and $\xi_n^{2;l}$, respectively, for all $j = 1, 2, \dots, N_1$ and $n = 1, 2, \dots, N$.

Secondly, we fix $V = V^{(l+1)}$, $U = U^{(l+1)}$, $\rho = \rho^{(l)}$, and then the (W, b) subproblem can be written as

$$\min_{W, b \in \Omega_3} \mathcal{G}(W, b, V^{(l+1)}, U^{(l+1)}, \rho^{(l)}).$$

By simply calculation, we can obtain its closed-form solution as follows.

$$W^{(l+1)} = \widehat{W}^{(l+1)} \widetilde{I}^\top, \text{ and } \quad b^{(l+1)} = \mathrm{Proj}_{\Omega_3}(\widehat{W}^{(l+1)} s_{N_0+1}, \bar{b}_2 - g_{b_2}/L), \tag{B.3}$$

where

$$\widehat{W}^{(l+1)} = \left(-[g_W, g_{b_1}] + L[\bar{W}, \bar{b}_1] + \rho^{(l)} \widehat{X}^\top + U^{(l+1)} \widehat{X}^\top \right) \left(L I_{N_0+1} + 2\lambda_1 \widetilde{I}^\top \widetilde{I} + \widehat{X} \widehat{X}^\top \right)^{-1},$$

$s_{N_0+1} = (0, 0, \ldots, 0, 1) \in \mathbb{R}^{N_0+1}$, $[g_{b_1}^\top, g_{b_2}^\top] = g_b^\top$, $\widehat{X} := (X^\top, 1_N)^\top$ and $\widetilde{I} = [I_{N_0}, 0]$.

Finally, we present the framework of ADMM for solving the subproblem (3.3).

Algorithm B.1 A Splitting and Alternating Method for the Quadratic Programming (3.3) (SAMQP)

```
Initialization: set $(W^{(l)}, b^{(l)}, V^{(l)}) = (\bar{W}, \bar{b}, \bar{V})$, $\rho^{(l)} = 0$, $u_n^{(l)} = W^{(l)} x_n + b_1^{(l)}$ for all $n = 1, ..., N$, and $l := 0$.
while the stop criterion is not met do
    Calculate $V^{(l+1)}, U^{(l+1)}$ by (B.2);
    Calculate $W^{(l+1)}, b^{(l+1)}$ by (B.3);
    Calculate $\rho_n^{(l+1)} = \rho_n^{(l)} + (u_n^{(l+1)} - (W^{(l+1)} x_n + b_1^{(l+1)}))$, for all $n = 1, 2, \ldots, N$;
    Set $l := l + 1$.
end while
```

Since the subproblem (3.3) is strongly convex, any sequence generated by SAMQP, a two block ADMM, converges to a global solution of (3.3). Furthermore, the local R-linear rate convergence of SAMQP can be guaranteed by Boley [2].

To test the efficiency of SAMQP, we construct the following randomly generated test problems. We set $X = \mathrm{rand}(N_0, N)$, $g_W = \mathrm{rand}(N_1, N_0)$, $g_b = \mathrm{rand}(N_1 + N_0, 1)$, $g_V = \mathrm{rand}(N_1, N)$, $\bar{W} = \mathrm{randn}(N_1, N_0)/N$, $\bar{V}_n = (\bar{W} x_n)_+$ for all $n = 1, 2, \ldots, N$ and $\bar{b} = 0$. The problem parameters μ and L are set as 0.001 and 1, respectively. In addition, the stopping criterion is set as

$$\max \left\{ \|\rho^{(l+1)} - \rho^{(l)}\|_F^2, \|U^{(l+1)} - U^{(l)}\|_F^2 \right\} \leq 10^{-6}.$$

We compare SAMQP with some existing QP solvers including the 'quadprog' solver from MATLAB, the 'fmincon' solver from MATLAB and the 'CVX' solver [15] for solving (3.3). We choose seven test problems with different sizes. We record the CPU time in seconds required by these solvers. The results are displayed in Table 2, in which "–" stands for the cases that the solver runs out of memory during the iteration or terminates abnormally. It can be easily observed that SAMQP is the most efficient and robust one among these four solvers.

Table 2: A comparison of CPU time for several solvers and SAMQP.

				CPU time (s)			
N	N_1	N_0	N_2	'fmincon'	'quadprog'	'CVX'	SAMQP
100	5	5	535	3.502	0.707	2.031	0.099
100	10	10	1120	33.990	4.546	1.172	0.105
100	20	20	2440	674.163	39.303	1.781	0.189
100	40	40	5680	–	359.555	6.672	0.419
100	100	10	11110	–	–	7.453	0.838
1000	100	10	101110	–	–	50.781	6.056
10000	784	1000	8625784	–	–	–	189.868