A HALF-SPACE STOCHASTIC PROJECTED GRADIENT METHOD FOR GROUP SPARSITY REGULARIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Optimizing with group sparsity is significant in enhancing model interpretability in machining learning applications, *e.g.*, feature selection, compressed sensing and model compression. However, for large-scale stochastic training problems, effective group sparsity exploration are typically hard to achieve. Particularly, the state-of-the-art stochastic optimization algorithms usually generate merely dense solutions. To overcome this shortage, we propose a stochastic method—Half-space Stochastic Projected Gradient (HSPG) method to search solutions of high group sparsity while maintain the convergence. Initialized by a simple Prox-SG Step, the HSPG method relies on a novel Half-Space Step to substantially boost the sparsity level. Numerically, HSPG demonstrates its superiority in deep neural networks, *e.g.*, VGG16, ResNet18 and MobileNetV1, by computing solutions of higher group sparsity, competitive objective values and generalization accuracy.

1 INTRODUCTION

In many recent machine learning optimization tasks, researchers not only focus on finding solutions with small prediction/generalization error but also concentrate on improving the interpretation of model by filtering out redundant parameters and achieving slimmer model architectures. One technique to achieve the above goal is by augmenting the sparsity-inducing regularization terms to the raw objective functions to generate sparse solutions (including numerous zero elements). The popular ℓ_1-regularization promotes the sparsity of solutions by element-wise penalizing the optimization variables. However, in many practical applications, there exist additional constraints on variables such that the zero coefficients are often not randomly distributed but tend to be clustered into varying more sophisticated sparsity structures, *e.g.*, disjoint and overlapping groups and hierarchy (Yuan & Lin, 2006; Huang et al., 2010; 2009). As the most important and natural form of structured sparsity, the disjoint group-sparsity regularization, which assumes the pre-specified disjoint blocks of variables are selected (non-zero variables) or ignored (zero variables) simultaneously (Bach et al., 2012), serves as a momentous role in general structured sparsity learning tasks since other instances such as overlapping group and hierarchical sparsity are typically solved by converting into the equivalent disjoint group versions via introducing latent variables (Bach et al., 2012), and has found numerous applications in computer vision (Elhamifar et al., 2012), signal processing (Chen & Selesnick, 2014), medical imaging (Liu et al., 2018), and deep learning (Scardapane et al., 2017), especially on the model compression of deep neural networks, where the group sparsity[1] is leveraged to remove redundant entire hidden structures directly.

Problem Setting. We study the disjoint group sparsity regularization problem which can be typically formulated as the mixed ℓ_1/ℓ_p-regularization problem, and pay special attention to the most popular and widely used instance p as 2 (Bach et al., 2012; Halabi et al., 2018),

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \left\{ \Psi(x) \overset{\text{def}}{=} f(x) + \lambda\Omega(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x) + \lambda \sum_{g\in\mathcal{G}} \|[x]_g\| \right\}, \tag{1}$$

where $\lambda > 0$ is a weighting factor, $\|\cdot\|$ denotes ℓ_2-norm, $f(x)$ is the average of numerous N continuously differentiable instance functions $f_i : \mathbb{R}^n \to \mathbb{R}$, such as the loss functions measuring the deviation from the observations in various data fitting problems, $\Omega(x)$ is the so-called mixed ℓ_1/ℓ_2

[1] Group sparsity is defined as # of zero groups, where a zero group means all its variables are exact zeros.

norm, $\mathcal{G}$ is a prescribed fixed partition of index set $\mathcal{I} = \{1, 2, \cdots, n\}$, wherein each component $g \in \mathcal{G}$ indexes a group of variables upon the perspective of applications. Theoretically, a larger λ typically results in a higher group sparsity while sacrifices more on the bias of model estimation, hence λ needs to be carefully fine-tuned to achieve both low f and high group-sparse solutions.

Literature Review. Problem (1) has been well studied in deterministic optimization with various algorithms that are capable of returning solutions with both low objective value and high group sparsity under proper λ (Yuan & Lin, 2006; Roth & Fischer, 2008; Huang et al., 2011; Ndiaye et al., 2017). Proximal methods are classical approaches to solve the structured non-smooth optimization (1), including the popular proximal gradient method (Prox-FG) which only uses the first-order derivative information. When N is huge, stochastic methods become ubiquitous to operate on a small subset to avoid the costly evaluation over all instances in deterministic methods for large-scale problems. Proximal stochastic gradient method (Prox-SG) (Duchi & Singer, 2009) is the natural stochastic extension of Prox-FG. Regularized dual-averaging method (RDA) (Xiao, 2010; Yang et al., 2010) is proposed by extending the dual averaging scheme in (Nesterov, 2009). To improve the convergence rate, there exists a set of incremental gradient methods inspired by SAG (Roux et al., 2012) to utilizes the average of accumulated past gradients. For example, proximal stochastic variance-reduced gradient method (Prox-SVRG) (Xiao & Zhang, 2014) and proximal spider (Prox-Spider) (Zhang & Xiao, 2019) are developed to adopt multi-stage schemes based on the well-known variance reduction technique SVRG proposed in (Johnson & Zhang, 2013) and Spider developed in (Fang et al., 2018) respectively. SAGA (Defazio et al., 2014) stands as the midpoint between SAG and Prox-SVRG.

Compared to deterministic methods, the studies of mixed ℓ_1/ℓ_2-regularization (1) in stochastic field become somewhat rare and limited. Prox-SG, RDA, Prox-SVRG, Prox-Spider and SAGA are valuable state-of-the-art stochastic algorithms for solving problem (1) but with apparent weakness. Particularly, these existing stochastic algorithms typically meet difficulties to achieve both decent convergence and effective group sparsity identification simultaneously (e.g., small function values but merely dense solutions), because of the randomness and the limited sparsity-promotion mechanisms. In depth, Prox-SG, RDA, Prox-SVRG, Prox-Spider and SAGA derive from proximal gradient method to utilize the proximal operator to produce group of zero variables. Such operator is generic to extensive non-smooth problems, consequently perhaps not sufficiently insightful if the target problems possess certain properties, *e.g.*, the group sparsity structure as problem (1). In fact, in convex setting, the proximal operator suffers from variance of gradient estimate; and in non-convex setting, especially deep learning, the discreet step size (learning rate) further deteriorates its effectiveness on the group sparsity promotion, as will show in Section 2 that the projection region vanishes rapidly except RDA. Besides, the variance reduction techniques are typically required to measure over a huge mini-batch data points in both theory and practice which is probably prohibitive for large-scale problems, and have been observed as sometimes noneffective for deep learning applications (Defazio & Bottou, 2019). On the other hand, to introduce sparsity, there exist heuristic weight pruning methods (Li et al., 2016; Luo et al., 2017), whereas they commonly do not equip with theoretical guarantee, so that easily diverge and hurt generalization accuracy.

Our Contributions. Half-Space Stochastic Projected Gradient (HSPG) method overcomes the limitations of the existing stochastic algorithms on the group sparsity identification, while maintains comparable convergence characteristics. While the main-stream works on (group) sparsity have focused on using proximal operators of regularization, our method is unique and fresh in enforcing group sparsity more effectively by leveraging half-space structure and is well supported by the theoretical analysis and empirical evaluations. We now summarize our contributions as follows.

- *Algorithmic Design:* We propose the HSPG to solve the disjoint group sparsity regularized problem as (1). Initialized with a Prox-SG Step for seeking a close-enough but perhaps dense solution estimate, the algorithmic framework relies on a novel Half-Space Step to exploit group sparse patterns. We delicately design the Half-Space Step with the following main features: *(i)* it utilizes previous iterate as the normal direction to construct a reduced space consisting of a set of half-spaces and the origin; *(ii)* a new group projection operator maps groups of variables onto zero if they fall out of the constructed reduced space to identify group sparsity considerably more effectively than the proximal operator; and *(iii)* with proper step size, the Half-Space Step enjoys the sufficient decrease property, and achieves progress to optimum in both theory and practice.

- *Theoretical Guarantee:* We provide the convergence guarantees of HSPG. Moreover, we prove HSPG has looser requirements to identify the sparsity pattern than Prox-SG, revealing its superiority

on the group sparsity exploration. Particularly, for the sparsity pattern identification, the required distance to the optimal solution x^* of HSPG is better than the distance required by Prox-SG.

- *Numerical Experiments:* Experimentally, HSPG outperforms the state-of-the-art methods in the aspect of the group sparsity exploration, and achieves competitive objective value convergence and runtime in both convex and non-convex problems. In the popular deep learning tasks, HSPG usually computes the solutions with multiple times higher group sparsity and similar generalization performance on unseen testing data than those generated by the competitors, which may be further used to construct smaller and more efficient network architectures.

2 THE HSPG METHOD

We state the Half-Space Stochastic Projected Gradient (HSPG) method in Algorithm 1. In general, it contains two stages: Initialization Stage and Group-Sparsity Stage. The first Initialization Stage employs Prox-SG Step (Algorithm 2) to search for a close-enough but usually non-sparse solution estimate. Then the second and fundamental stage proceeds Half-Space Step (Algorithm 3) started with the non-sparse solution estimate to effectively exploit the group sparsity within a sequence of reduced spaces, and converges to the group-sparse solutions with theoretical convergence property.

Algorithm 1 Outline of HSPG for solving (1).

1: **Input:** $x_0 \in \mathbb{R}^n$, $\alpha_0 \in (0,1)$, $\epsilon \in [0,1)$, and $N_{\mathcal{P}} \in \mathbb{Z}^+$.
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: **if** $k < N_{\mathcal{P}}$ **then**
4: Compute $x_{k+1} \leftarrow$ Prox-SG(x_k, α_k) by Algorithm 2.
5: **else**
6: Compute $x_{k+1} \leftarrow$ Half-Space$(x_k, \alpha_k, \epsilon)$ by Algorithm 3.
7: Update α_{k+1}.

Algorithm 2 Prox-SG Step.

1: **Input:** Current iterate x_k, and step size α_k.
2: Compute the stochastic gradient of f on mini-batch $\mathcal{B}_k$

$$\nabla f_{\mathcal{B}_k}(x_k) \leftarrow \frac{1}{|\mathcal{B}_k|} \sum_{i \in \mathcal{B}_k} \nabla f_i(x_k). \tag{2}$$

3: **Return** $x_{k+1} \leftarrow \text{Prox}_{\alpha_k \lambda \Omega(\cdot)}\left(x_k - \alpha_k \nabla f_{\mathcal{B}_k}(x_k)\right)$.

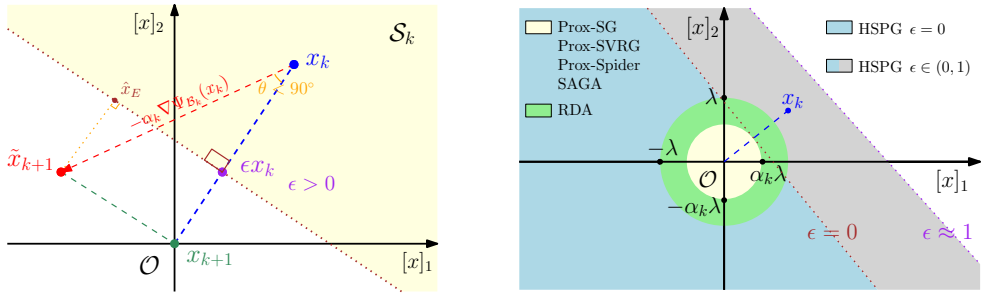
Initialization Stage. The Initialization Stage performs the vanilla proximal stochastic gradient method (Prox-SG, Algorithm 2) to approach the solution of (1). At kth iteration, a mini-batch $\mathcal{B}_k$ is sampled to generate an unbiased estimator of the full gradient of f (line 2, Algorithm 2) to compute a trial iterate $\widehat{x}_{k+1} := x_k - \alpha_k \nabla f_{\mathcal{B}_k}(x_k)$, where α_k is the step size, and $f_{\mathcal{B}_k}$ is the average of the instance functions f_i cross $\mathcal{B}_k$. The next iterate x_{k+1} is then updated based on the proximal mapping

$$x_{k+1} = \text{Prox}_{\alpha_k \lambda \Omega(\cdot)}(\hat{x}_{k+1}) = \arg\min_{x \in \mathbb{R}^n} \frac{1}{2\alpha_k} \|x - \hat{x}_{k+1}\|^2 + \lambda \Omega(x), \tag{3}$$

where the regularization term $\Omega(x)$ is defined in (1). Notice that the above subproblem (3) has a closed-form solution, where for each $g \in \mathcal{G}$, we have

$$[x_{k+1}]_g = \max\left\{0, 1 - \alpha_k \lambda / \|[\widehat{x}_{k+1}]_g\|\right\} \cdot [\widehat{x}_{k+1}]_g. \tag{4}$$

In HSPG, the Initialization Stage proceeds Prox-SG Step $N_{\mathcal{P}}$ times as a localization mechanism to seek an estimation which is close enough to a solution of problem (1) as drawn later in Theorem 1 of Section 3. In practice, although the close-enough requirement is perhaps hard to be verified, we empirically suggest to keep running the Prox-SG Step until observing some stage-switch signal by testing on the stationarity of objective values, norm of (sub)gradient or validation accuracy similarly to (Zhang et al., 2020). However, the Initialization Stage alone is *insufficient* to exploit the group sparsity structure, *i.e.*, the computed solution estimate is typically dense, due to the randomness and



(a) Half-Space Projection

(b) Projection Region

Figure 1: Illustration of Half-Space Step with projection in (9), where $\mathcal{G} = \{\{1, 2\}\}$.

the moderate truncation mechanism of proximal operator constrained in its projection region, *i.e.*, the trial iterate $[\widehat{x}_{k+1}]_g$ is projected to zero only if it falls into an ℓ_2-ball centered at the origin with radius $\alpha_k \lambda$ by (4). Our remedy is to incorporate it with the following Half-Space Step, which exhibits an effective sparsity promotion mechanism while still remains the convergent property.

Algorithm 3 Half-Space Step

1: **Input:** Current iterate x_k, step size α_k, and ϵ.
2: Compute the stochastic gradient of Ψ on $\mathcal{I}^{\neq 0}(x_k)$ by mini-batch $\mathcal{B}_k$

$$[\nabla \Psi_{\mathcal{B}_k}(x_k)]_{\mathcal{I}^{\neq 0}(x_k)} \leftarrow \frac{1}{|\mathcal{B}_k|} \sum_{i \in \mathcal{B}_k} [\nabla \Psi_i(x_k)]_{\mathcal{I}^{\neq 0}(x_k)} \tag{5}$$

3: Compute $[\tilde{x}_{k+1}]_{\mathcal{I}^{\neq 0}(x_k)} \leftarrow [x_k - \alpha_k \nabla \Psi_{\mathcal{B}_k}(x_k)]_{\mathcal{I}^{\neq 0}(x_k)}$ and $[\tilde{x}_{k+1}]_{\mathcal{I}^{0}(x_k)} \leftarrow 0$.
4: **for** each group g in $\mathcal{I}^{\neq 0}(x_k)$ **do**
5: **if** $[\tilde{x}_{k+1}]_g^\top [x_k]_g < \epsilon \, \|[x_k]_g\|^2$ **then**
6: $[\tilde{x}_{k+1}]_g \leftarrow 0$.
7: **Return** $x_{k+1} \leftarrow \tilde{x}_{k+1}$.

Group-Sparsity Stage. The Group-Sparsity Stage is designed to effectively determine the groups of zero variables and capitalize convergence characteristic, which is in sharp contrast to other heuristic aggressive weight pruning methods but typically lacking theoretical guarantee (Li et al., 2016; Luo et al., 2017). The underlying intuition of its atomic Half-Space Step (Algorithm 3) is to project $[x_k]_g$ to zero only if $-[x_k]_g$ serves as a descent step to $\Psi(x_k)$, *i.e.*, $-[x_k]_g^\top [\nabla \Psi(x_k))]_g < 0$, hence updating $[x_{k+1}]_g \leftarrow [x_k]_g - [x_k]_g = 0$ still results in some progress to the optimality. Before introducing that, we first define the following index sets for any $x \in \mathbb{R}^n$:

$$\mathcal{I}^0(x) := \{g : g \in \mathcal{G}, [x]_g = 0\} \text{ and } \mathcal{I}^{\neq 0}(x) := \{g : g \in \mathcal{G}, [x]_g \neq 0\}, \tag{6}$$

where $\mathcal{I}^0(x)$ represents the indices of groups of zero variables at x, and $\mathcal{I}^{\neq 0}(x)$ indexes the groups of nonzero variables at x. To proceed, we further define an artificial set that x lies in:

$$\mathcal{S}(x) := \left\{ z \in \mathbb{R}^n : [z]_g = 0 \text{ if } g \in \mathcal{I}^0(x), \text{and } [z]_g^T [x]_g \geq \epsilon \, \|[x]_g\|^2 \text{ if } g \in \mathcal{I}^{\neq 0}(x) \right\} \bigcup \{0\}, \tag{7}$$

which consists of half-spaces and the origin. Here the parameter $\epsilon > 0$ controls the grey region presented in Figure 1b, and the exact way to set ϵ will be discussed in Section 4 and Appendix. Hence, x inhabits $\mathcal{S}(x_k)$, *i.e.*, $x \in \mathcal{S}(x_k)$, only if: *(i)* $[x]_g$ lies in the upper half-space for all $g \in \mathcal{I}^{\neq 0}(x_k)$ for some prescribed $\epsilon \in [0, 1)$ as shown in Figure 1a; and *(ii)* $[x]_g$ equals to zero for all $g \in \mathcal{I}^0(x_k)$. The fundamental assumption for Half-Space Step to success is that: the Initialization Stage has produced a (possibly *non-sparse*) solution estimate x_k nearby a group sparse solution x^* of problem (1), *i.e.*, the optimal distance $\|x_k - x^*\|$ is sufficiently small. As seen in Appendix, it further indicates that the group sparse optimal solution x^* inhabits $\mathcal{S}_k := \mathcal{S}(x_k)$, which implies that $\mathcal{S}_k$ has already covered the group-support of x^*, *i.e.*, $\mathcal{I}^{\neq 0}(x^*) \subseteq \mathcal{I}^{\neq 0}(x_k)$. Our goal now becomes minimizing $\Psi(x)$ over

$\mathcal{S}_k$ to identify the remaining groups of zero variables, *i.e.*, $\mathcal{I}^0(x^*)/\mathcal{I}^0(x_k)$, which is formulated as the following smooth optimization problem:

$$x_{k+1} = \arg\min_{x \in \mathcal{S}_k} \ \Psi(x) = f(x) + \lambda \Omega(x). \tag{8}$$

By the definition of $\mathcal{S}_k$, $[x]_{\mathcal{I}^0(x_k)} \equiv 0$ are constrained as fixed during Algorithm 3 proceeding, and only the entries in $\mathcal{I}^{\neq 0}(x_k)$ are allowed to move. Hence $\Psi(x)$ is smooth on $\mathcal{S}_k$, and (8) is a reduced space optimization problem. A standard way to solve problem (8) would be the stochastic gradient descent equipped with Euclidean projection (Nocedal & Wright, 2006). However, such a projected method rarely produces zero (group) variables as the dense $\hat{x}_E$ illustrated in Figure 1a. To address it, we introduce a novel projection operator to effectively conduct group projection as follows.

As stated in Algorithm 3, we first approximate the gradient of Ψ on the free variables in $\mathcal{I}^{\neq 0}(x_k)$ by $[\nabla \Psi_{\mathcal{B}_k}(x_k)]_{\mathcal{I}^{\neq 0}(x_k)}$ (line 2, Algorithm 3), then employ SGD to compute a trial point $\widetilde{x}_{k+1}$ (line 3, Algorithm 3) which is passed into a new projection operator $\mathrm{Proj}_{\mathcal{S}_k}(\cdot)$ defined as

$$\left[\mathrm{Proj}_{\mathcal{S}_k}(z)\right]_g := \begin{cases} [z]_g & \text{if } [z]_g^T [x_k]_g \geq \epsilon \left\| [x_k]_g \right\|^2, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

The above projector of form (9) is not the standard Euclidean projection operator in most cases[2], but still satisfies the following two advantages: *(i)* the actual search direction $d_k := (\mathrm{Proj}_{\mathcal{S}_k}(\tilde{x}_{k+1}) - x_k)/\alpha_k$ performs as a descent direction to $\Psi_{\mathcal{B}_k}(x_k) := f_{\mathcal{B}_k}(x_k) + \lambda \Omega(x_k)$, *i.e.*, $[d_k]_g^\top [\nabla \Psi_{\mathcal{B}_k}(x_k))]_g < 0$ as $\theta < 90^\circ$ in Figure 1a, then the progress to the optimum is made via the sufficient decrease property as drawn in Lemma 1; and *(ii)* effectively project groups of variables to zero simultaneously if the inner product of corresponding entries is sufficiently small. In contrast, the Euclidean projection operator is far away effective to promote group sparsity, as the Euclidean projected point $\hat{x}_E \neq 0$ versus $x_{k+1} = \mathrm{Proj}_{\mathcal{S}_k}(\tilde{x}_{k+1}) = 0$ shown in Figure 1a.

Lemma 1. *Algorithm 3 yields the next iterate x_{k+1} as $\mathrm{Proj}_{\mathcal{S}_k}(x_k - \alpha_k \nabla \Psi_{\mathcal{B}_k}(x_k))$, then the search direction $d_k := (x_{k+1} - x_k)/\alpha_k$ is a descent direction for $\Psi_{\mathcal{B}_k}(x_k)$, i.e., $d_k^\top \nabla \Psi_{\mathcal{B}_k}(x_k) < 0$. Moreover, letting L be the Lipschitz constant for $\nabla \Psi_{\mathcal{B}_k}$ on the feasible domain, and $\hat{\mathcal{G}}_k := \mathcal{I}^{\neq 0}(x_k) \bigcap \mathcal{I}^0(x_{k+1})$ and $\tilde{\mathcal{G}}_k := \mathcal{I}^{\neq 0}(x_k) \bigcap \mathcal{I}^{\neq 0}(x_{k+1})$ be the sets of groups which projects or not onto zero, we have*

$$\Psi_{\mathcal{B}_k}(x_{k+1}) \leq \Psi_{\mathcal{B}_k}(x_k) - \left(\alpha_k - \frac{\alpha_k^2 L}{2}\right) \sum_{g \in \tilde{\mathcal{G}}_k} \left\| [\nabla \Psi_{\mathcal{B}_k}(x_k)]_g \right\|^2 - \left(\frac{1-\epsilon}{\alpha_k} - \frac{L}{2}\right) \sum_{g \in \hat{\mathcal{G}}_k} \left\| [x_k]_g \right\|^2. \tag{10}$$

We then intuitively illustrate the strength of HSPG on group sparsity exploration. In fact, the half-space projection (9) is a more effective sparsity promotion mechanism compared to the existing methods. Particularly, it benefits from a much larger projection region to map a reference point $\hat{x}_{k+1} := x_k - \alpha_k \nabla f_{\mathcal{B}_k}(x_k)$ or its variants to zero. As the 2D case described in Figure 1b, the projection regions of Prox-SG, Prox-SVRG, Prox-Spider and SAGA are ℓ_2-balls with radius as $\alpha_k \lambda$. In stochastic learning, especially deep learning tasks, the step size α_k is usually selected around 10^{-3} to 10^{-4} or even smaller for convergence. Together with the common setting of $\lambda \ll 1$, their projection regions would vanish rapidly, resulting in the difficulties to produce group sparsity. As a sharp contrast, even though $\alpha_k \lambda$ is near zero, the projection region of HSPG $\{x : x_k^T x < (\alpha_k \lambda + \epsilon \|x_k\|) \|x_k\|\}$ (seen in Appendix) is still an open half-space which contains those ℓ_2 balls as well as RDA's if ϵ is large enough. Moreover, the positive control parameter ϵ adjusts the level of aggressiveness of group sparsity promotion (9), *i.e.*, the larger the more aggressive, and meanwhile maintains the progress to the optimality by Lemma 1. In practice, proper fine tuning ϵ is sometimes required to achieve both group sparsity enhancement and sufficient decrease on objective value as will see in Section 4.

Intuition of Two-Stage Method: To end this section, we discuss the advantage of designing such two stage schema rather than an adaptive switch back and forth between the Prox-SG Step and Half-Space Step based on some evaluation switching criteria, as many multi-step deterministic optimization algorithms (Chen et al., 2017). In fact, we numerically observed that switching back to the Prox-SG Step consistently deteriorate the progress of group sparsity exploration by Half-Space Step while without obvious gain on convergence. Such regression on group sparsity by the Prox-SG Step is less attractive in realistic applications, *e.g.*, model compression, where people usually possess

[2]Unless $\Omega(x)$ is $\|x\|_1$ where each $g \in \mathcal{G}$ is singleton, then $\mathcal{S}_k$ becomes an orthant face (Chen et al., 2020).

heavy models of high generalization accuracy ahead and want to filter out the redundancy effectively. Therefore, in term of the ease of application, we end at organizing Prox-SG Step and Half-Space Step as such a two-stage schema, controlled by a switching hypermeter $N_{\mathcal{P}}$. In theory, we require $N_{\mathcal{P}}$ sufficiently large to let the initial iterate of Half-Space Step be close enough to the local minimizer as shown in Section 3. In practice, HSPG is sensitive to the choice of $N_{\mathcal{P}}$ at early iterations, *i.e.*, switching to Half-Space Step too early may result accuracy loss. But such sensitivity vanishes rapidly if switching to Half-Space Step after some acceptable evaluation switching criteria.

3 Convergence Analysis

In this section, we give the convergence guarantee of our HSPG. Towards that end, we make the following widely used assumption in optimization literature (Xiao & Zhang, 2014; Yang et al., 2019) and active set identification analysis of regularization problem (Nutini et al., 2019; Chen et al., 2018).

Assumption 1. *Each $f_i : \mathbb{R}^n \to \mathbb{R}$, for $i = 1, 2, \cdots, N$, is differentiable and bounded below. Their gradients $\nabla f_i(x)$ are Lipschitz continuous, and let L be the shared Lipschitz constant.*

Assumption 2. *The least and the largest ℓ_2-norm of non-zero groups in x^* are lower and upper bounded by some constants, i.e., $0 < 2\delta_1 := \min_{g \in \mathcal{I}^{\neq 0}(x^*)} \|[x^*]_g\|$ and $0 < 2\delta_2 := \max_{g \in \mathcal{I}^{\neq 0}(x^*)} \|[x^*]_g\|$. Moreover, we request a common strict complementarity on any $\mathcal{B}$, i.e., $0 < 2\delta_3 := \min_{g \in \mathcal{I}^0(x^*)} (\lambda - \|[\nabla f_{\mathcal{B}}(x^*)]_g\|)$ for regularization optimization.*

Notations: Let x^* be a local minimizer of problem (1) with group sparsity property, Ψ^* be the local minimum value corresponding to x^*, and $\{x_k\}_{k=0}^{\infty}$ be the iterates generated from Algorithm 1. Denote the gradient mapping of $\Psi(x)$ and its estimator on mini-batch $\mathcal{B}$ as $\xi_\eta(x) := \frac{1}{\eta}\left(x - \text{Prox}_{\eta\lambda\Omega(\cdot)}(x - \eta\nabla f(x))\right)$ and $\xi_{\eta,\mathcal{B}}(x) := \frac{1}{\eta}\left(x - \text{Prox}_{\eta\lambda\Omega(\cdot)}(x - \eta\nabla f_{\mathcal{B}}(x))\right)$ respectively. We say $\tilde{x}$ a stationary point of $\Psi(x)$ if $\xi_\eta(\tilde{x}) = 0$. The noise $e(x) := \xi_{\eta,\mathcal{B}}(x) - \xi_\eta(x)$ is zero-mean because of the random sampling of mini-batch $\mathcal{B}$, *i.e.*, $\mathbb{E}_{\mathcal{B}}[e(x)] = 0$. To be simple, let $\widetilde{\mathcal{X}}$ be a neighbor of x^* as $\widetilde{\mathcal{X}} := \{x : \|x - x^*\| \leq R\}$ with R as a positive constant related to δ_1, δ_2 and ϵ (see Appendix), and M be the supremum of $\|\partial\Psi(x)\|$ on the compact set $\widetilde{\mathcal{X}}$.

Remark: Assumption 1 implies that $\nabla f_{\mathcal{B}}(x)$ measured on mini-batch $\mathcal{B}$ is Lipschitz continuous on $\mathbb{R}^n$ with the same Lipschitz constant L, while $\nabla\Psi_{\mathcal{B}}(x)$ is not as shown in Appendix. However, the Lipschitz continuity of $\nabla\Psi_{\mathcal{B}}(x)$ still holds on $\mathcal{X} = \{x : \|[x]_g\| \geq \delta_1 \text{ for each } g \in \mathcal{G}\}$ by excluding a ℓ_2-ball centered at the origin with radius δ_1 from $\mathbb{R}^n$. For simplicity, let $\nabla\Psi_{\mathcal{B}}(x)$ share the same Lipschitz constant L on $\mathcal{X}$ with $\nabla f_{\mathcal{B}}(x)$, since we can always select the bigger value as their shared Lipschitz constant. Now, we state the first main theorem of HSPG.

Theorem 1. *Suppose f is convex on $\widetilde{\mathcal{X}}$, $\epsilon \in \left[0, \min\left\{\frac{\delta_1^2}{\delta_2}, \frac{2\delta_1 - R}{2\delta_2 + R}\right\}\right)$, $\|x_K - x^*\| \leq \frac{R}{2}$ for $K \geq N_{\mathcal{P}}$. Set $k := K + t$, $(t \in \mathbb{Z}^+)$, step size $\alpha_k = \mathcal{O}(\frac{1}{\sqrt{N}t}) \in \left(0, \min\left\{\frac{2(1-\epsilon)}{L}, \frac{1}{L}, \frac{2\delta_1 - R - \epsilon(2\delta_2 + R)}{M}\right\}\right)$, and mini-batch size $|\mathcal{B}_k| = \mathcal{O}(t) \leq N - \frac{N}{2M}$. Then for any $\tau \in (0, 1)$, $\{x_k\}$ converges to some stationary point in expectation with probability at least $1 - \tau$, i.e., $\mathbb{P}(\lim_{k\to\infty} \mathbb{E}\left[\|\xi_{\alpha_k,\mathcal{B}_k}(x_k)\|\right] = 0) \geq 1 - \tau$.*

Remark: Theorem 1 only requires local convexity of f on a neighborhood $\widetilde{\mathcal{X}}$ of x^* while itself can be non-convex in general. This local convexity assumption appears in many non-convex analysis, such as: tensor decomposition (Ge et al., 2015) and shallow neural networks (Zhong et al., 2017). Theorem 1 implies that if the Kth iterate locates close enough to x^*, the step size α_k and mini-batch size $|\mathcal{B}_k|$ is set as above, (it further indicates x^* inhabits the $\{\mathcal{S}_k\}_{k \geq K}$ of all subsequent iterates updated by Half-Space Step with high probability in Appendix), then the Half-Space Step in Algorithm 3 guarantees the convergence to the stationary point. The $\mathcal{O}(t)$ mini-batch size is commonly used in the analysis of stochastic algorithms, *e.g.*, Adam and Yogi (Zaheer et al., 2018). Later based on numerical results in Section 4, we observe that a much weaker increasing or even constant mini-batch size is sufficient. In fact, experiments show that practically, a reasonably large mini-batch size can work well if the variance is not large. Although the assumption $\|x_K - x^*\| < R/2$ is hard to be verified in practice, setting $N_{\mathcal{P}}$ large enough usually performs quite well.

We then reveal the sparsity identification guarantee of HSPG as stated in Theorem 2.

Theorem 2. *If $k \geq N_{\mathcal{P}}$ and $\|x_k - x^*\| \leq \frac{2\alpha_k\delta_3}{1 - \epsilon + \alpha_k L}$, then HSPG yields $\mathcal{I}^0(x^*) \subseteq \mathcal{I}^0(x_{k+1})$.*

Remark: Theorem 2 shows that when x_k is in the ℓ_2-ball centered at x^* with radius $\frac{2\alpha_k\delta_3}{1-\epsilon+\alpha_k L}$, HSPG identifies the optimal sparsity pattern, *i.e.*, $\mathcal{I}^0(x^*) \subseteq \mathcal{I}^0(x_{k+1})$. In contrast, to identify the sparsity pattern, Prox-SG requires the iterates to fall into the ℓ_2-ball centered at x^* with radius $\alpha_k\delta_3$ (Nutini et al., 2019). Since $\alpha_k \leq 1/L$ and $\epsilon \in [0, 1)$, then $\frac{2\alpha_k\delta_3}{1-\epsilon+\alpha_k L} \geq \alpha_k\delta_3$ implies that the ℓ_2-ball of HSPG contains the ℓ_2-ball of Prox-SG, *i.e.*, HSPG has a stronger performance in sparsity pattern identification. Therefore, Theorem 2 reveals a better sparsity identification property of HSPG than Prox-SG, and no similar results exist for other methods to our knowledge.

The Initialization Stage Selection: To satisfy the pre-requirement of convergence of Half-Space Step as Theorem 1, *i.e.*, initial iterate close enough to x^*, there exists several proper candidates *e.g.*, Prox-SG, Prox-SVRG and SAGA to form as the Initialization Stage. Considering the tradeoff between computational efficiency and theoritical convergence, our default setting is to select Prox-SG. Although Prox-SVRG/SAGA may have better theoretical convergence property than Prox-SG, they require higher time and space complexity to compute or estimate full gradient on a huge mini-batch or store previous gradient, which may be prohibitive for large-scale training especially when the memory is often limited. Besides, it is well noticed that SVRG does not work as desired on the popular non-convex deep learning applications (Defazio & Bottou, 2019; Chen et al., 2020). In contrast, Prox-SG is efficient and can also achieves the good initialization assumption in Theorem 1. We now provide a sufficient condition that employing Prox-SG Step sufficiently many times can compute such an iterate $x_{N_\mathcal{P}}$ sufficiently close to x^* with high probability as Proposition 1, which is established under an additional but popular Polyak-Lojasiewicz (PL) condition for non-smooth problem (Li & Li, 2018), *i.e.*, there exists a $\mu > 0$ such that for any $x \in \mathbb{R}^n$ and $\eta > 0$, $\|\xi_\eta(x)\|^2 \geq 2\mu(\Psi(x) - \Psi^*)$.

Proposition 1. *Suppose f is convex and Ψ satisfies the PL condition. There exist constants $C > 0, \gamma \in (0, 1/2L)$ such that for any fixed $\tau \in (0, 1)$, if $\alpha_k \equiv \alpha < \min\left\{\frac{2\gamma\mu\tau R^2}{(2L\gamma-1)C}, \frac{1}{2\mu}, \frac{1}{L}\right\}$, and mini-batch size $|\mathcal{B}_k| \equiv |\mathcal{B}| > \frac{32\gamma\mu M^2}{2\gamma\mu\tau R^2-(2L\gamma-1)C\alpha}$ for any $k < N_\mathcal{P}$, then $\|x_{N_\mathcal{P}} - x^*\|_2 \leq R/2$ holds with probability at least $1-\tau$, i.e., $\mathbb{P}(\|x_{N_\mathcal{P}} - x^*\| \leq R/2) \geq 1-\tau$ for any $N_\mathcal{P} \geq K$ with $K := \left\lceil \frac{\log\left(poly(\tau R^2, 1/|\mathcal{B}|, \alpha)/(\Psi(x_0)-\Psi^*)\right)}{\log(1-2\mu\alpha)} \right\rceil$, where poly$(\cdot)$ is some polynomial of assembled variables.*

Remark: PL condition is sufficient but not necessary for ensuring iterate close enough to the optima. Proposition 1 implies that after sufficient number of iterations, Prox-SG produces an iterate $x_{N_\mathcal{P}}$ that is $R/2$-close to x^* with high probability. However, one should notice that Prox-SG does not guarantee any group sparsity property of $x_{N_\mathcal{P}}$ due to the limited projection region and randomness.

4 Numerical Experiments

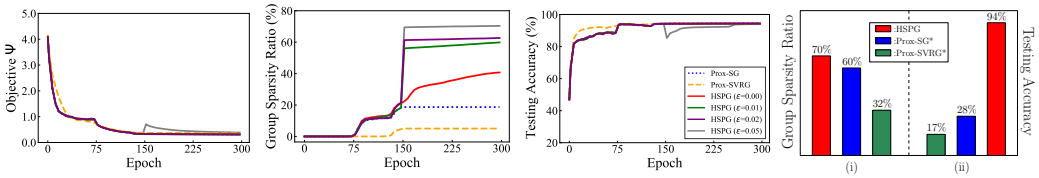
In this section, we present results of several benchmark numerical experiments in deep neural networks to illustrate the superiority of HSPG than other related algorithms on group sparsity exploration and the comparable convergence. Besides, two extensible convex experiments are conducted in Appendix to empirically demonstrate the validness and superiority of the group sparsity identification by HSPG.

Image Classification: We now consider the popular Deep Convolutional Neural Networks (DCNNs) for image classification tasks. Specifically, we select several popular and benchmark DCNN architectures, *i.e.*, VGG16 (Simonyan & Zisserman, 2014), ResNet18 (He et al., 2016) and MobileNetV1 (Howard et al., 2017) on two benchmark datasets CIFAR10 (Krizhevsky & Hinton, 2009) and Fashion-MNIST (Xiao et al., 2017). We conduct all experiments for 300 epochs with a mini-batch size of 128 and λ as 10^{-3}, since it returns competitive testing accuracy to the models trained without regularization. The step size α_k is initialized as 0.1, and decayed by a factor 0.1 periodically. We set each filter in the convolution layers as a group variable.

In these experiments, we proceed a test on the objective value stationarity similarly to (Zhang et al., 2020, Section 2.1) and switch to Half-Space Step roughly on 150 epochs with $N_\mathcal{P}$ as $150N/|\mathcal{B}|$. The control parameter ϵ in the half-space projection (9) controls the aggressiveness level of group sparsity promotion, which is first set as 0, then fined tuned to be around 0.02 to favor the sparsity level whereas does not hurt the target objective Ψ; the detailed procedure is in Appendix. We exclude RDA because of no acceptable experimental results attained during our tests with the step size parameter γ setting throughout all powers of 10 from 10^{-3} to 10^3, and skip Prox-Spider and SAGA since Prox-SVRG has been a superb representative to the proximal incremental gradient methods.

Table 1: Final Ψ/group sparsity ratio/testing accuracy for tested algorithms on non-convex problems.

Backbone	Dataset	Prox-SG	Prox-SVRG	HSPG	
				ϵ as 0	fine tuned ϵ
VGG16	CIFAR10	**0.59** / 53.95% / 90.57%	0.82 / 14.73% / 89.42%	**0.59** / 74.60% / **91.10%**	**0.59** / **75.61%** / 90.92%
	Fashion-MNIST	0.54 / 15.63% / **92.99%**	2.66 / 0.45% / 92.69%	0.54 / 22.18% / 92.98%	**0.53** / **60.77%** / 92.87%
ResNet18	CIFAR10	**0.31** / 19.50% / 94.09%	0.36 / 2.79% / 94.17%	**0.31** / 41.58% / 94.39%	**0.31** / **62.97%** / **94.53%**
	Fashion-MNIST	0.14 / 0.00% / 94.82%	0.19 / 0.00% / 94.64%	**0.13** / 6.60% / **94.93%**	**0.13** / **63.93%** / 94.86%
MobileNetV1	CIFAR10	**0.40** / 57.81% / 91.60%	0.65 / 32.22% / 90.08%	**0.40** / 65.04% / **91.86%**	0.41 / **71.66%** / 91.54%
	Fashion-MNIST	**0.22** / 65.80% / 94.36%	0.48 / 38.76% / 93.95%	0.23 / 74.52% / 94.43%	0.24 / **83.71%** / **94.44%**



(a) Objective Ψ (b) Group Sparsity Ratio (c) Testing Accuracy (d) HSPG VS Truncation

Figure 2: On ResNet18 with CIFAR10, (a)-(c): Evolution of Ψ, group sparsity ratio and testing accuracy, (d): HSPG versus Prox-SG* and Prox-SVRG* (Prox-SG and Prox-SVRG with simple truncation mechanism).

Table 1 demonstrates the effectiveness and superiority of HSPG, where we mark the best values as bold, and the group sparsity ratio is defined as the percentage of zero groups. In particular, *(i)* HSPG computes remarkably higher group sparsity than other methods on all tests under both $\epsilon = 0$ and fine tuned ϵ, of which the solutions are typically multiple times sparser in the manner of group than those of Prox-SG, while Prox-SVRG performs not comparably since the variance reduction techniques may not work as desired for deep learning applications (Defazio & Bottou, 2019); *(ii)* HSPG performs competitively with respect to the final objective values Ψ and f (see f in Appendix). In addition, all the methods reach a comparable generalization performance on unseen test data. On the other hand, sparse regularization methods may yield solutions with entries that are not exactly zero but are very small. Sometimes all entries below certain threshold ($\mathcal{T}$) are set to zero (Jenatton et al., 2010; Halabi et al., 2018). However, such simple truncation mechanism is heuristic-rule based, hence may hurt convergence and accuracy. To illustrate this, we set the groups of the solutions of Prox-SG and Prox-SVRG to zero if the magnitudes of the group variables are less than some $\mathcal{T}$, and denote the corresponding solutions as Prox-SG* and Prox-SVRG*. As shown in Figure 2d(i), under the $\mathcal{T}$ with no accuracy regression, Prox-SG* and Prox-SVRG* reach higher group sparsity ratio as 60% and 32% compared to Table 1, but still significantly lower than the 70% of HSPG under $\epsilon = 0.05$ without simple truncation. Under the $\mathcal{T}$ to reach the same group sparsity ratio as HSPG, the testing accuracy of Prox-SG* and Prox-SVRG* regresses drastically to 28% and 17% in Figure 2d(ii) respectively.

Finally, we investigate the group sparsity evolution under different ϵ's. As shown in Figure 2b, HSPG produces the highest group-sparse solutions compared with other methods. Notably, at the early $N_{\mathcal{P}}$ iterations, HSPG performs merely the same as Prox-SG. However, after switching to Half-Space Step at the 150th epoch, HSPG outperforms all the other methods dramatically, and larger ϵ results in higher sparsity level. It is a strong evidence that our half-space based technique is much more successful than the proximal mechanism and its variants in terms of the group sparsity identification. Besides, the evolutions of Ψ and testing accuracy confirm the comparability on convergence among the tested algorithms. Particularly, the objective Ψ generally monotonically decreases for small $\epsilon = 0$ to 0.02, and experiences a mild pulse after switch to Half-Space Step for larger ϵ, *e.g.*, 0.05, which matches Lemma 1. As a result, with the similar generalization accuracy, HSPG allows dropping entire hidden units of networks, which may further achieve automatic dimension reduction and construct smaller model architectures for efficient inference.

5 Conclusions

We proposed a new Half-Space Stochastic Projected Gradient (HSPG) method for disjoint group-sparsity induced regularized problem, which makes use of proximal stochastic gradient method to seek a near-optimal solution estimate, followed by a novel half-space group projection to effectively exploit the group sparsity structure. In theory, we provided the convergence guarantee, and showed its better sparsity identification performance. Experiments on both convex and non-convex problems demonstrated that HSPG usually achieves solutions with competitive objective values and significantly higher group sparsity compared with state-of-the-arts stochastic solvers.

REFERENCES

Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2012.

Po-Yu Chen and Ivan W Selesnick. Group-sparse signal denoising: non-convex regularization, convex optimization. *IEEE Transactions on Signal Processing*, 62(13):3464–3478, 2014.

Tianyi Chen, Frank E Curtis, and Daniel P Robinson. A reduced-space algorithm for minimizing ℓ_1-regularized convex functions. *SIAM Journal on Optimization*, 27(3):1583–1610, 2017.

Tianyi Chen, Frank E Curtis, and Daniel P Robinson. Farsa for ℓ_1-regularized convex optimization: local convergence and numerical experience. *Optimization Methods and Software*, 2018.

Tianyi Chen, Tianyu Ding, Bo Ji, Guanyi Wang, Yixin Shi, Sheng Yi, Xiao Tu, and Zhihui Zhu. Orthant based proximal stochastic gradient method for ℓ_1-regularized optimization. *arXiv preprint arXiv:2004.03639*, 2020.

Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. In *Advances in Neural Information Processing Systems*, 2019.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pp. 1646–1654, 2014.

John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10(Dec):2899–2934, 2009.

Ehsan Elhamifar, Guillermo Sapiro, and Rene Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1600–1607. IEEE, 2012.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pp. 689–699, 2018.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pp. 797–842, 2015.

Marwa El Halabi, Francis Bach, and Volkan Cevher. Combinatorial penalties: Which structures are preserved by convex relaxations? In *International Conference on Artificial Intelligence and Statistics*, pp. 1551–1560. PMLR, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Junzhou Huang, Xiaolei Huang, and Dimitris Metaxas. Learning with dynamic group sparsity. In *2009 IEEE 12th International Conference on Computer Vision*, pp. 64–71. IEEE, 2009.

Junzhou Huang, Tong Zhang, et al. The benefit of group sparsity. *The Annals of Statistics*, 38(4): 1978–2004, 2010.

Junzhou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. *Journal of Machine Learning Research*, 12(Nov):3371–3412, 2011.

Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. Structured sparse principal component analysis. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 366–373, 2010.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pp. 315–323, 2013.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

Zhize Li and Jian Li. A simple proximal stochastic gradient method for nonsmooth nonconvex optimization. In *Advances in neural information processing systems*, 2018.

Shujun Liu, Jianxin Cao, Hongqing Liu, Xichuan Zhou, Kui Zhang, and Zhengzhou Li. Mri reconstruction via enhanced group sparsity and nonconvex regularization. *Neurocomputing*, 272: 108–121, 2018.

Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision*, pp. 5058–5066, 2017.

Eugene Ndiaye, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Gap safe screening rules for sparsity enforcing penalties. *The Journal of Machine Learning Research*, 18(1):4671–4703, 2017.

Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 2009.

Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

Julie Nutini, Mark Schmidt, and Warren Hare. "active-set complexity" of proximal gradient: How long does it take to find the sparsity pattern? *Optimization Letters*, 13(4):645–655, 2019.

Volker Roth and Bernd Fischer. The group-lasso for generalized linear models: uniqueness of solutions and efficient algorithms. In *Proceedings of the 25th international conference on Machine learning*, pp. 848–855, 2008.

Nicolas L Roux, Mark Schmidt, and Francis R Bach. A stochastic gradient method with an exponential convergence _rate for finite training sets. In *Advances in neural information processing systems*, pp. 2663–2671, 2012.

Simone Scardapane, Danilo Comminiello, Amir Hussain, and Aurelio Uncini. Group sparse regularization for deep neural networks. *Neurocomputing*, 241:81–89, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.

Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Haiqin Yang, Zenglin Xu, Irwin King, and Michael R Lyu. Online learning for group lasso. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 1191–1198, 2010.

Minghan Yang, Andre Milzarek, Zaiwen Wen, and Tong Zhang. A stochastic extra-step quasi-newton method for nonsmooth nonconvex optimization. *arXiv preprint arXiv:1910.09373*, 2019.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In *Advances in neural information processing systems*, pp. 9793–9803, 2018.

Junyu Zhang and Lin Xiao. Multi-level composite stochastic optimization via nested variance reduction. *arXiv preprint arXiv:1908.11468*, 2019.

Pengchuan Zhang, Hunter Lang, Qiang Liu, and Lin Xiao. Statistical adaptive stochastic gradient methods. *arXiv preprint arXiv:2002.10597*, 2020.

Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon. Recovery guarantees for one-hidden-layer neural networks. In *International Conference on Machine Learning*, 2017.