Training Structured Neural Networks Through Manifold Identification and Variance Reduction

Anonymous authors
Paper under double-blind review

Abstract

This paper proposes an algorithm, RMDA, for training neural networks (NNs) with a regularization term for promoting desired structures. RMDA does not incur computation additional to proximal SGD with momentum, and achieves variance reduction without requiring the objective function to be of the finite-sum form. Through the tool of manifold identification from nonlinear optimization, we prove that after a finite number of iterations, all iterates of RMDA possess a desired structure identical to that induced by the regularizer at the stationary point of asymptotic convergence, even in the presence of engineering tricks like data augmentation and dropout that complicate the training process. Experiments on training NNs with structured sparsity confirm that variance reduction is necessary for such an identification, and show that RMDA thus significantly outperforms existing methods for this task. For unstructured sparsity, RMDA also outperforms a state-of-the-art pruning method, validating the benefits of training structured NNs through regularization.

1 Introduction

Training neural networks (NNs) with regularization to obtain a certain desired structure such as structured sparsity or discrete-valued parameters is a problem of increasing interest. Existing approaches either use stochastic subgradients of the regularized objective (Wen et al., 2016; 2018) or combine popular stochastic gradient algorithms for NNs, like SGD with momentum (MSGD) or Adam (Kingma & Ba, 2015), with the proximal operator associated with the regularizer to conduct proximal stochastic gradient updates to obtain a model with preferred structures (Bai et al., 2018; Yang et al., 2019; Yun et al., 2020; Deleu & Bengio, 2021). Such methods come with proven convergence for certain measures of first-order optimality and have shown some empirical success in applications. However, we notice that an essential theoretical support lacking in existing methods is the guarantee for the output iterate to possess the same structure as that at the point of convergence. More specifically, often the imposed regularization is only known to induce a desired structure exactly at optimal or stationary points of the underlying optimization problem (see for example, Zhao & Yu, 2006), but training algorithms are only able to generate iterates asymptotically converging to a stationary point. Without further theoretical guarantees, it is unknown whether the output iterate, which is just an approximation of the stationary point, still has the same structure. For example, let us assume that sparsity is desired, the point of convergence is $x^* = (1, 0, 0)$, and two algorithms respectively produce iterates $\{y^t = (1, t^{-1}, t^{-1})\}$ and $\{z^t = (1 + t^{-1}, 0, 0)\}$. Clearly, both iterate sequences converge to x^*, but only z^t has the same desired structure as its limit point x^*, while y^t is not useful for sparsity despite that the point of convergence is. This work aims at filling this gap to propose an algorithm for training structured NNs that can provably make all its iterates after a finite number of iterations possess the desired structure of the stationary point to which the iterates converge. We term the structure at a stationary point a stationary structure, and it should be understood that for multiple stationary points, each might correspond to a different stationary structure, and we aim at identifying the one at the limit point

of the iterates of an algorithm, instead of selecting the optimal one among all stationary structures. Although finding the structure at an inferior stationary point might seem not very meaningful, another reason for studying this identification property is that for the same point of convergence, the structure at the limit point is the most preferable one. Consider the same example above, we note that for any sequence $\{x^t\}$ converging to x^*, $x_1^t \neq 0$ for all t large enough, for otherwise x^t does not converge to x^*. Therefore, x^t cannot be sparser than x^* if $x^t \to x^*$.[1] Identifying the structure of the point of convergence thus also amounts to finding the locally most ideal structure under the same convergence premise.

It is well-known in the literature of nonlinear optimization that generating iterates consistently possessing the structure at the stationary point of convergence is possible if all points with the same structure near the stationary point can be presented locally as a manifold along which the regularizer is smooth. This manifold is often termed as the active manifold (relative to the given stationary point), and the task of generating iterates staying in the active manifold relative to the point of convergence after finite iterations is called manifold identification (Lewis, 2002; Hare & Lewis, 2004; Lewis & Zhang, 2013). To identify the active manifold of a stationary point, we need the regularizer to be partly smooth (Lewis, 2002; Hare & Lewis, 2004) at that point, roughly meaning that the regularizer is smooth along the active manifold around the point, while the change in its value is drastic along directions leaving the manifold. A more technical definition will be given in Section 3. Fortunately, most regularizers used in machine learning are partly smooth, so stationary structure identification is possible, and various deterministic algorithms are known to achieve so (Hare & Lewis, 2007; Hare, 2011; Wright, 2012; Liang et al., 2017a;b; Li et al., 2020; Lee, 2020; Bareilles et al., 2020).

On the other hand, for stochastic gradient-related methods to identify a stationary structure, existing theory suggests that the variance of the gradient estimation needs to vanish as the iterates approach a stationary point (Poon et al., 2018), and indeed, it is observed empirically that proximal stochastic gradient descent (SGD) is incapable of manifold identification due to the presence of the variance in the gradient estimation (Lee & Wright, 2012; Sun et al., 2019).[2] Poon et al. (2018) showed that variance-reduction methods such as SVRG (Johnson & Zhang, 2013; Xiao & Zhang, 2014) and SAGA (Defazio et al., 2014) that utilize the finite-sum structure of empirical risk minimization to drive the variance of their gradient estimators to zero are suitable for this task. Unfortunately, with the standard practice of data augmentation and dropout in deep learning, training of deep learning models with a regularizer should be treated as the following stochastic optimization problem that minimizes the expected loss over a distribution, instead of the commonly seen finite-sum form:

$$\min_{W \in \mathcal{E}} \quad F(W) \coloneqq \mathbb{E}_{\xi \sim \mathcal{D}} \left[f_\xi(W) \right] + \psi(W), \tag{1}$$

where $\mathcal{E}$ is a Euclidean space with inner product $\langle \cdot, \cdot \rangle$ and the associated norm $\|\cdot\|$, $\mathcal{D}$ is a distribution over a space Ω, f_ξ is differentiable almost everywhere for all $\xi \in \Omega$, and $\psi(W)$ is a regularizer that might be nondifferentiable. We will also use the notation $f(W) \coloneqq \mathbb{E}_{\xi \sim \mathcal{D}}[f_\xi(W)]$. Without a finite-sum structure in (1), Defazio & Bottou (2019) pointed out that classical variance-reduction methods are ineffective for deep learning, and one major reason is the necessity of periodically evaluating $\nabla f(W)$ (or at least using a large batch from $\mathcal{D}$ to get a precise approximation of it) in variance-reduction methods is intractable, hence manifold identification and therefore finding the stationary structure becomes an extremely tough task for deep learning. Although recently there are efforts in developing variance-reduction methods for (1) inspired by online problems (Wang et al., 2019; Nguyen et al., 2021; Pham et al., 2020; Cutkosky & Orabona, 2019; Tran-Dinh et al., 2019), these methods all have multiple hyperparameters to tune and incur computational cost at least twice or thrice to that of (proximal) SGD. As the training of deep learning models is time- and resource-consuming, these drawbacks make such methods less ideal for deep learning.

To tackle these difficulties, we extend the recently proposed modernized dual averaging framework (Jelassi & Defazio, 2020) to the regularized setting by incorporating proximal

[1] See a more detailed discussion in Appendix C.1.

[2] An exception is the interpolation case, in which the variance of plain SGD vanishes asymptotically. But data augmentation and dropout often fail this interpolation condition.

operations, and obtain a new algorithm RMDA (Regularized Modernized Dual Averaging) for (1). The proposed algorithm provably achieves variance reduction beyond finite-sum problems without any cost or hard-to-tune hyperparameters additional to those of proximal momentum SGD (proxMSGD), and we provide theoretical guarantees for its convergence and ability for manifold identification. The key difference between RMDA and the original regularized dual averaging (RDA) of Xiao (2010) is that RMDA incorporates momentum and can achieve better performance for deep learning in terms of the generalization ability, and the new algorithm requires nontrivial proofs for its guarantees. We further conduct experiments on training deep learning models with a regularizer for structured-sparsity to demonstrate the ability of RMDA to identify the stationary structure without sacrificing the prediction accuracy.

When the desired structure is (unstructured) sparsity, a popular approach is pruning that trims a given dense model to a specified level, and works like (Gale et al., 2019; Blalock et al., 2020; Evci et al., 2020; Verma & Pesquet, 2021) have shown promising results. Among them, it is worth noting that Verma & Pesquet (2021) also utilized proximal operations like structured training methods including ours. However, as a post-processing approach, pruning is essentially different from structured training considered in this work, as pruning is mainly used when a model is available, while structured training combines training and structure inducing in one procedure to potentially reduce the computational cost and memory footprint when resources are scarce. We will also show in our experiment that RMDA can achieve better performance than a state-of-the-art pruning method, suggesting that structured training indeed has its merits for obtaining sparse NNs.

The main contributions of this work are summarized as follows.

- ***Principled analysis***: We use the theory of manifold identification from nonlinear optimization to provide a unified way towards better understanding of algorithms for training structured neural networks.
- ***Variance reduction beyond finite-sum with low cost***: RMDA achieves variance reduction for problems that consist of an infinite-sum term plus a regularizer (see Lemma 2) while incorporating momentum to improve the generalization performance. Its spatial and computational cost is almost the same as proxMSGD, and there is no additional hyperparameters to tune, making RMDA suitable for large-scale deep learning.
- ***Structure identification***: With the help of variance reduction, our theory shows that under suitable conditions, after a finite number of iterations, iterates of RMDA stay in the active manifold of its limit point.
- ***Superior empirical performance***: Experiments on neural networks with structured sparsity exemplify that RMDA can identify a stationary structure without reducing the validation accuracy, thus outperforming existing methods by achieving higher group sparsity. Another experiment on unstructured sparsity also shows RMDA outperforms a state-of-the-art pruning method.

2 Algorithm

Details of the proposed RMDA are in Algorithm 1. At the t-th iteration with the iterate W^{t-1}, we draw an independent sample $\xi_t \sim \mathcal{D}$ to compute the stochastic gradient $\nabla f_{\xi_t}(W^{t-1})$, decide a learning rate η_t, and update the weighted sum V_t of previous stochastic gradients using η_t and the scaling factor $\beta_t \coloneqq \sqrt{t}$:

$$V_0 \coloneqq 0, \quad V_t \coloneqq \sum\nolimits_{k=1}^{t} \eta_k \beta_k \nabla f_{\xi_k}(W^{k-1}) = V_{t-1} + \eta_t \beta_t \nabla f_{\xi_t}(W^{t-1}), \quad \forall t > 0.$$

The tentative iterate $\tilde{W}^t$ is then obtained by the proximal operation associated with ψ:

$$\tilde{W}^t = \operatorname{prox}_{\alpha_t \beta_t^{-1} \psi}\left(W^0 - \beta_t^{-1} V^t\right), \quad \alpha_t \coloneqq \sum\nolimits_{k=1}^{t} \beta_k \eta_k, \tag{2}$$

where for any function g, $\operatorname{prox}_g(x) \coloneqq \arg\min_y \|y - x\|^2/2 + g(y)$ is its proximal operator. The iterate is then updated along the direction $\tilde{W}^t - W^{t-1}$ with a factor of $c_t \in [0, 1]$:

$$W^t = (1 - c_t) W^{t-1} + c_t \tilde{W}^t = W^{t-1} + c_t \left(\tilde{W}^t - W^{t-1}\right). \tag{3}$$

Algorithm 1: RMDA $(W^0, T, \eta(\cdot), c(\cdot))$

input : Initial point W^0, learning rate schedule $\eta(\cdot)$, momentum schedule $c(\cdot)$, number of epochs T

$V_0 \leftarrow 0, \quad \alpha_0 \leftarrow 0$

for $t = 1, \ldots, T$ **do**

  $\beta_t \leftarrow \sqrt{t}, \quad s_t \leftarrow \eta(t)\beta_t, \quad \alpha_t \leftarrow \alpha_{t-1} + s_t$

  Sample $\xi_t \sim \mathcal{D}$ and compute $V^t \leftarrow V^{t-1} + s_t \nabla f_{\xi_t}(W^{t-1})$

  $\tilde{W}^t \leftarrow \arg\min_W \langle V^t, W \rangle + \frac{\beta_t}{2}\left\|W - W^0\right\|^2 + \alpha_t \psi(W)$ // (2)

  $W^t \leftarrow (1 - c(t))W^{t-1} + c(t)\tilde{W}^t$

output: The final model W^T

When $\psi \equiv 0$, RMDA reduces to the modernized dual averaging algorithm of Jelassi & Defazio (2020), in which case it has been shown that mixing W^{t-1} and $\tilde{W}^t$ in (3) equals to introducing momentum (Jelassi & Defazio, 2020; Tao et al., 2018). We found that this introduction of momentum greatly improves the performance of RMDA and is therefore essential for applying it on deep learning problems.

3 Analysis

We provide theoretical analysis of the proposed RMDA in this section. Our analysis shows variance reduction in RMDA and stationarity of the limit point of its iterates, but all of them revolves around our main purpose of identification of a stationary structure within a finite number of iterations. The key tools for this end are partial smoothness and manifold identification (Hare & Lewis, 2004; Lewis, 2002). Our result is the currently missing cornerstone for those proximal algorithms applied to deep learning problems for identifying desired structures. In fact, it is actually well-known in convex optimization that those algorithms based on plain proximal stochastic gradient without variance reduction are *unable to identify the active manifold*, and the structure of the iterates oscillates due to the variance in the gradient estimation; see, for example, experiments and discussions in Lee & Wright (2012); Sun et al. (2019). Our work is therefore the first one to provide justification for solving the regularized optimization problem in deep learning to really identify a desired structure induced by the regularizer. Throughout, ∇f_ξ denotes the gradient of f_ξ, $\partial\psi$ is the (regular) subdifferential of ψ, and $\operatorname{relint}(C)$ means the relative interior of the set C.

We start from introducing the notion of partial smoothness.

Definition 1. *A function ψ is partly smooth at a point W^* relative to a set $\mathcal{M}_{W^*} \ni W^*$ if*

1. *Around W^*, $\mathcal{M}_{W^*}$ is a $\mathcal{C}^2$-manifold and $\psi|_{\mathcal{M}_{W^*}}$ is $\mathcal{C}^2$.*
2. *ψ is regular (finite with the Fréchet subdifferential coincides with the limiting Fréchet subdifferential) at all points $W \in \mathcal{M}_{W^*}$ around W^* with $\partial\psi(W) \neq \emptyset$.*
3. *The affine span of $\partial\psi(W^*)$ is a translate of the normal space to $\mathcal{M}_{W^*}$ at W^*.*
4. *$\partial\psi$ is continuous at W^* relative to $\mathcal{M}_{W^*}$.*

We often call $\mathcal{M}_{W^*}$ the active manifold at W^*. Another concept required for manifold identification is prox-regularity (Poliquin & Rockafellar, 1996).

Definition 2. *A function ψ is prox-regular at W^* for $V^* \in \partial\psi(W^*)$ if ψ is finite at W^*, locally lower semi-continuous around W^*, and there is $\rho > 0$ such that $\psi(W_1) \geq \psi(W_2) + \langle V, W_1 - W_2 \rangle - \frac{\rho}{2}\|W_1 - W_2\|^2$ whenever W_1, W_2 are close to W^* with $\psi(W_2)$ near $\psi(W^*)$ and $V \in \partial\psi(W_2)$ near V^*. ψ is prox-regular at W^* if it is so for all $V \in \partial\psi(W^*)$.*

To broaden the applicable range, a function ψ prox-regular at some W^* is often also assumed to be subdifferentially continuous (Poliquin & Rockafellar, 1996) there, meaning that if $W^t \to W^*$, $\psi(W^t) \to \psi(W^*)$ holds when there are $V^* \in \partial\psi(W^*)$ and a sequence $\{V^t\}$ such that $V^t \in \partial\psi(W^t)$ and $V^t \to V^*$. Notably, all convex and weakly-convex (Nurminskii, 1973) functions are regular, prox-regular, and subdifferentially continuous in their domain.

3.1 Theoretical Results

By using the techniques in Nesterov & Shikhman (2015), firstly we show that under convexity, the expected objective of RMDA with a certain parameter scheduling converges to the optimum. We also note that when $c_t \equiv 1$, RMDA reduces to the classical RDA, and convergence to a global optimum (of $W^t = \tilde{W}^t$ in this case) on convex problems has been proven by Lee & Wright (2012); Duchi & Ruan (2021).

Theorem 1. *Assume f and ψ are convex, then given any W^0, for $c_t = \alpha_{t+1}^{-1}$ and for any global optimum W^* of (1) with objective F^*, the following convergence guarantee holds.*

$$2\mathbb{E}\left[F\left(W^k\right) - F^*\right] \leq \frac{\beta_t}{\alpha_t}\left\|W^* - W^0\right\|^2 + \alpha_t^{-1}\sum_{k=1}^{t}\eta_k^2\beta_k\mathbb{E}\left\|\nabla f_{\xi_k}\left(W^{k-1}\right)\right\|^2 + \alpha_t^{-1}\textit{constant}. \quad (4)$$

In particular, if $\alpha_t = t$ and $\mathbb{E}\left\|\nabla f_{\xi_t}\left(W^{t-1}\right)\right\|^2 \leq C$ for all t for some $C \geq 0$, we have that $\mathbb{E}[F(W^t)] - F^* \leq O(t^{-1/2})$.

Next, we show that if $\{\tilde{W}^t\}$ converges to a point W^* (which could be a non-stationary one), $\{W^t\}$ also converges to W^*.

Lemma 1. *Consider Algorithm 1 with $\{c_t\}$ satisfying $\sum c_t = \infty$. If $\{\tilde{W}^t\}$ converges to a point W^*, $\{W^t\}$ also converges to W^*.*

We then show that if $\{\tilde{W}^t\}$ converges to a point, almost surely this point of convergence is stationary. This requires the following lemma for variance reduction of RMDA, meaning that the variance of using V_t to estimate $\nabla f(W^{t-1})$ reduces to zero, as $\alpha_t^{-1}V_t$ converges to $\nabla f(W^{t-1})$ almost surely, and this result could be of its own interest. The first claim below uses a classical result in stochastic optimization that can be found at, for example, (Gupal, 1979, Theorem 4.1, Chapter 2.4), but the second one is, to our knowledge, new.

Lemma 2. *Consider Algorithm 1. Assume for any $\xi \sim \mathcal{D}$, f_ξ is L-Lipschitz-continuously-differentiable almost surely for some L, so f is also L-Lipschitz-continuously-differentiable, and there is $C \geq 0$ such that $\mathbb{E}_{\xi_t\sim\mathcal{D}}\left\|\nabla f_{\xi_t}\left(W^{t-1}\right)\right\|^2 \leq C$ for all t. If $\{\eta_t\}$ satisfies*

$$\sum \beta_t\eta_t\alpha_t^{-1} = \infty, \quad \sum\left(\beta_t\eta_t\alpha_t^{-1}\right)^2 < \infty, \quad \left\|W^{t+1} - W^t\right\|\left(\beta_t\eta_t\alpha_t^{-1}\right)^{-1} \xrightarrow{a.s.} 0, \quad (5)$$

then $\alpha_t^{-1}V^t \longrightarrow \nabla f(W^{t-1})$ with probability one. Moreover, if $\{W^t\}$ lies in a bounded set, we get $\mathbb{E}\left\|\alpha_t^{-1}V^t - \nabla f\left(W^{t-1}\right)\right\|^2 \to 0$ even if the second condition in (5) is replaced by a weaker condition of $\beta_t\eta_t\alpha_t^{-1} \to 0$.

In general, the last condition in (5) requires some regularity conditions in F to control the change speed of W^t. One possibility is when ψ is the indicator function of a convex set, $\beta_t\eta_t \propto t^p$ for $t \in (1/2, 1)$ will satisfy this condition. However, in other settings for η_t, even when F and ψ are both convex, existing analyses for the classical RDA such that $c_t \equiv 1$ in Algorithm 1 still need an additional local error bound assumption to control the change of $W^{t+1} - W^t$. Hence, to stay focused on our main message, we take this assumption for granted, and leave finding suitable sufficient conditions for it as future work.

With the help of Lemmas 1 and 2, we can now show the stationarity result for the limit point of the iterates. The assumption of $\beta_t\alpha_t^{-1}$ approaching 0 below is classical in analyses of dual averaging in order to gradually remove the influence of the term $\left\|W - W^0\right\|^2$.

Theorem 2. *Consider Algorithm 1 with the conditions in Lemmas 1 and 2 hold, and assume the set of stationary points $\mathcal{Z} := \{W \mid 0 \in \partial F(W)\}$ is nonempty and $\beta_t\alpha_t^{-1} \to 0$. For any given W^0, consider the event that $\{\tilde{W}^t\}$ converges to a point W^* (each event corresponds to a different W^*), then if $\partial\psi$ is outer semicontinuous at W^*, and this event has a nonzero probability, $W^* \in \mathcal{Z}$, or equivalently, W^* is a stationary point, with probability one conditional on this event.*

Finally, with Lemmas 1 and 2 and Theorem 2, we prove the main result that the active manifold of the limit point is identified in finite iterations of RMDA under nondegeneracy.

Theorem 3. *Consider Algorithm* 1 *with the conditions in Theorem* 2 *satisfied. Consider the event of* $\{\tilde{W}^t\}$ *converging to a certain point* W^* *as in Theorem* 2*, if the probability of this event is nonzero;* ψ *is prox-regular and subdifferentially continuous at* W^* *and partly smooth at* W^* *relative to the active* $\mathcal{C}^2$ *manifold* $\mathcal{M}$*;* $\partial\psi$ *is outer semicontinuous at* W^**; and the nondegeneracy condition*

$$-\nabla f\left(W^*\right) \in \operatorname{relint} \partial\psi\left(W^*\right) \tag{6}$$

holds at W^**, then conditional on this event, almost surely there is* $T_0 \geq 0$ *such that*

$$\tilde{W}^t \in \mathcal{M}, \quad \forall t \geq T_0. \tag{7}$$

In other words, the active manifold at W^* *is identified by the iterates of Algorithm* 1 *after a finite number of iterations almost surely.*

As mentioned in Section 1, an important reason for studying manifold identification is to get the lowest-dimensional manifold representing the structure of the limit point, which often corresponds to a preferred property for the application, like the highest sparsity, lowest rank, or lowest VC dimension locally. See an illustrated example in Appendix C.1.

4 Applications in Deep Learning

We discuss two popular schemes of training structured deep learning models achieved through regularization to demonstrate the applications of RMDA. More technical details for applying our theory to the regularizers in these applications are in Appendix C.

4.1 Structured Sparsity

As modern deep NN models are often gigantic, it is sometimes desirable to trim the model to a smaller one when only limited resources are available. In this case, zeroing out redundant parameters during training at the group level is shown to be useful (Zhou et al., 2016), and one can utilize regularizers promoting structured sparsity for this purpose. The most famous regularizer of this kind is the group-LASSO norm (Yuan & Lin, 2006; Friedman et al., 2010). Given $\lambda \geq 0$ and a collection $\mathcal{G}$ of index sets $\{\mathcal{I}_g\}$ of the variable W, this convex regularizer is defined as

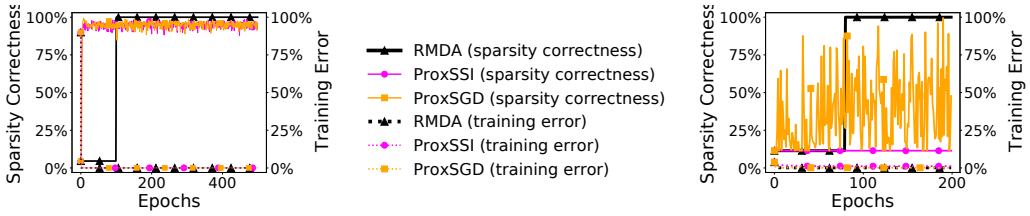
$$\psi(W) \coloneqq \lambda \sum\nolimits_{g=1}^{|\mathcal{G}|} w_g \left\| W_{\mathcal{I}_g} \right\|, \tag{8}$$

with $w_g > 0$ being the pre-specified weight for $\mathcal{I}_g$. For any W^*, let $\mathcal{G}_{W^*} \subseteq \mathcal{G}$ be the index set such that $W^*_{\mathcal{I}_j} = 0$ for all $j \in \mathcal{G}_{W^*}$, the group-LASSO norm is partly smooth around W^* relative to the manifold $\mathcal{M}_{W^*} \coloneqq \{W \mid W_{\mathcal{I}_i} = 0, \forall i \in \mathcal{G}_{W^*}\}$, so our theory applies.

In order to promote structured sparsity, we need to carefully design the grouping. Fortunately, in NNs, the parameters can be grouped naturally (Wen et al., 2016). For any fully-connected layer, let $W \in \mathbb{R}^{\text{out}\times\text{in}}$ be the matrix representation of the associated parameters, where out is the number of output neurons and in is that of input neurons, we can consider the column-wise groups, defined as $W_{:,j}$ for all j, and the row-wise groups of the form $W_{i,:}$. For a convolutional layer with $W \in \mathbb{R}^{\text{filter}\times\text{channel}\times\text{height}\times\text{width}}$ being the tensor form of the corresponding parameters, we can consider channel-wise, filter-wise, and kernel-wise groups, defined respectively as $W_{:,j,:,:}$, $W_{i,:,:,:}$ and $W_{i,j,:,:}$.

4.2 Binary Neural Networks

Making the parameters of an NN binary integers is another way to obtain a more compact model during training and deployment (Hubara et al., 2016), but discrete optimization is hard to scale-up. Using a vector representation $w \in \mathbb{R}^m$ of the variables, Hou et al. (2017) thus proposed to use the indicator function of $\left\{w \mid w_{\mathcal{I}_i} = \alpha_i b_{\mathcal{I}_i}, \alpha_i > 0, b_{\mathcal{I}_i} \in \{\pm 1\}^{|\mathcal{I}_i|}\right\}$ to induce the entries of w to be binary without resorting to discrete optimization tools, where each $\mathcal{I}_i$ enumerates all parameters in the i-th layer. Yang et al. (2019) later proposed to use $\min_{\alpha\in[0,1]^m} \quad \sum_{i=1}^m \left(\alpha_i (w_i+1)^2 + (1-\alpha_i)(w_i-1)^2\right)$ as the regularizer and to include α as



(a) Logistic regression (b) Legend (c) Convolutional network

Figure 1: Group sparsity pattern correctness and training error rates on synthetic data.

a variable to train. At any α^* with $I_0 \coloneqq \{i \mid \alpha_i^* = 0\}$ and $I_1 \coloneqq \{i \mid \alpha_i^* = 1\}$, the objective is partly smooth relative to the manifold $\{(W, \alpha) \mid \alpha_{I_0} = 0, \alpha_{I_1} = 1\}$. Extension to discrete NNs beyond the binary ones is possible, and Bai et al. (2018) have proposed regularizers with closed-form proximal operators for it.

5 Experiments

We use the structured sparsity application in Section 4.1 to empirically exemplify the ability of RMDA to find desired structures in the trained NNs. RMDA and the following methods for structured sparsity in deep learning are compared using PyTorch (Paszke et al., 2019).

- ProxSGD (Yang et al., 2019): A simple proxMSGD algorithm. To obtain group sparsity, we skip the interpolating step in Yang et al. (2019).
- ProxSSI (Deleu & Bengio, 2021): This is a special case of the adaptive proximal SGD framework of Yun et al. (2020) that uses the Newton-Raphson algorithm to approximately solve the subproblem. We directly use the package released by the authors.

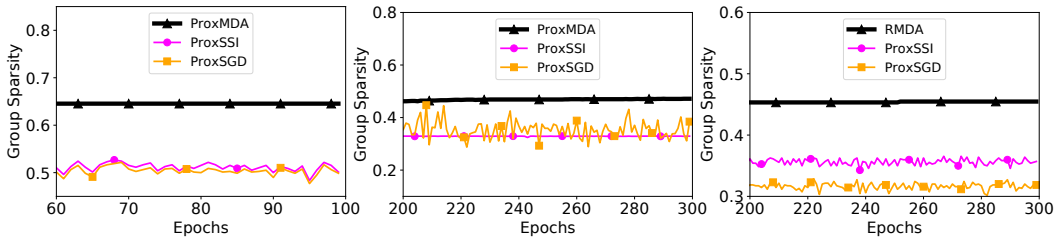
We exclude the algorithm of Wen et al. (2016) because their method is shown to be worse than ProxSSI by Deleu & Bengio (2021).

To compare these algorithms, we examine both the validation accuracy and the group sparsity level of their trained models. We compute the group sparsity as the percentage of groups whose elements are all zero, so the reported group sparsity is zero when there is no group with a zero norm, and is one when the whole model is zero. For all methods above, we use (8) with column-wise and channel-wise groupings in the regularization for training, but adopt the kernel-wise grouping in their group sparsity evaluation. Throughout the experiments, we always use multi-step learning rate scheduling that decays the learning rate by a constant factor every time the epoch count reaches a pre-specified threshold. For all methods, we conduct grid searches to find the best hyperparameters. All results shown in tables are the mean and standard deviation of three independent runs with the same hyperparameters, while figures use one representative run for better visualization.

In convex optimization, a popular way to improve the practical convergence behavior for momentum-based methods is restarting that periodically reset the momentum to zero (O'donoghue & Candes, 2015). Following this idea, we introduce a restart heuristic to RMDA. At each round, we use the output of Algorithm 1 from the previous round as the new input to the same algorithm, and continue using the scheduling η and c without resetting them. For $\psi \equiv 0$, Jelassi & Defazio (2020) suggested to increase c_t proportional to the decrease of η_t until reaching $c_t = 1$. We adopt the same setting for c_t and η_t and restart RMDA whenever η_t changes. As shown in Section 3 that $\tilde{W}^t$ finds the active manifold, increasing c_t to 1 also accords with our interest in identifying the stationary structure.

5.1 Correctness of Identified Structure Using Synthetic Data

Our first step is to numerically verify that RMDA can indeed identify the stationary structure desired. To exactly find a stationary point and its structure a priori, we consider synthetic problems. We first decide a ground truth model W that is structured sparse, generate random data points that can be well separated by W, and then decide their labels using



(a) Logistic regression (b) Fully-connected NN (c) LeNet5

Figure 2: Group Sparsity v.s. epochs of different algorithms of a single run.

W. The generated data are then taken as our training data. We consider a linear logistic regression model and a small NN that has one fully-connected layer and one convolutional layer. To ensure convergence to the ground truth, for logistic regression we generate more data points than the problem dimension to ensure the problem is strongly convex so that there is only one stationary/optimal point, and for the small NN, we initialize all algorithms close enough to the ground truth. We report in Fig. 1 training error rates (as an indicator for the proximity to the ground truth) and percentages of the optimal group sparsity pattern of the ground truth identified. Clearly, although all methods converge to the ground truth, only RMDA identifies the correct structure of it, and other methods without guarantees for manifold identification fail.

5.2 Linear Model and Fully-Connected Neural Network

We turn to real-world data and begin with two relatively simple problems to illustrate the difference between RMDA and other methods. The first is logistic regression with the MNIST dataset (LeCun et al., 1998) and the second is training a fully-connected NN on the FashionMNIST data (Xiao et al., 2017). From Figs. 2a and 2b, we see that similar to the previous experiment, the group sparsity level of RMDA is stable in the last epochs, while that of ProxSGD and ProxSSI oscillates below. This suggests that RMDA is the only method that identifies the structured sparsity at its limit point, and other methods with no variance reduction fail. Moreover, Table 1a shows that manifold identification of RMDA is achieved with no sacrifice of the validation accuracy.

5.3 Convolutional Neural Network

We now move to CNNs and start with training a version of LeNet5 on MNIST, and then consider a modified VGG16 (Simonyan & Zisserman, 2015) with the CIFAR10 dataset (Krizhevsky, 2009). For VGG16, we follow the practice in modern vision tasks to apply data augmentation through random cropping and horizontal flipping so that the training problem is no longer a finite-sum one. The group sparsity results are in Figs. 2c and 3. Notably, for VGG16, RMDA succeeds in finding the structured sparsity pattern despite the presence of data augmentation, showing that RMDA can indeed overcome the difficulty from the infinite-sum setting of modern deep learning tasks, but other methods all have group sparsity levels fluctuating way below that of RMDA. For this more complicated task, we include a baseline of MSGD with no sparsity-inducing regularizer in our comparison in Table 1b. Remarkably, RMDA beats ProxSGD and ProxSSI in both structured sparsity and accuracy, and its accuracy is close to that of MSGD. We also report that in the VGG16 task, on our NVIDIA RTX 8000 GPU, MSGD, ProxSGD, and RMDA all have an average per-epoch cost of around 39 seconds, while ProxSSI needs 189 seconds per epoch. RMDA is thus also more suitable for large-scale structured deep learning in terms of practical efficiency.

To test RMDA on more models, we further train a residual NN (ResNet50) with CIFAR100 and the same data augmentation above. This is a more difficult task and ProxSSI crashes after $1,170$ epochs, so we report its best result before crashing. Table 1c shows that RMDA still outperforms others in both metrics, and even attains the same accuracy as MSGD.

Table 1: Group sparsity and validation accuracy of different methods. We report mean and standard deviation of three runs. MSGD is the baseline with no group-sparsity regularizer.

(a) Simple models

	Logistic regression		Fully-connected NN	
Algorithm	Validation accuracy	Group sparsity	Validation accuracy	Group sparsity
ProxSGD	89.95 ± 0.00%	49.87 ± 0.00%	86.12 ± 0.36%	37.41 ± 1.47%
ProxSSI	89.80 ± 0.01%	50.13 ± 0.00%	**88.89 ± 0.07%**	31.12 ± 1.30%
RMDA	**89.96 ± 0.02%**	**64.58 ± 0.06%**	88.09 ± 0.21%	**47.53 ± 0.32%**

(b) LeNet5 and VGG16

	LeNet5		VGG16	
Algorithm	Validation accuracy	Group sparsity	Validation accuracy	Group sparsity
MSGD	99.28 ± 0.05%	-	93.05 ± 0.22%	-
ProxSGD	99.10 ± 0.03%	33.66 ± 2.43%	90.64 ± 0.10%	73.33 ± 1.29%
ProxSSI	99.01 ± 0.11%	34.14 ± 2.20%	92.19 ± 0.08%	85.29 ± 0.34%
RMDA	**99.11 ± 0.06%**	**45.33 ± 0.99%**	**92.52 ± 0.13%**	**88.10 ± 0.09%**

(c) ResNet50 (He et al., 2016)

Algorithm	Validation accuracy	Group sparsity
MSGD	76.69 ± 0.38%	
ProxSGD	74.83 ± 0.19%	53.26 ± 0.67%
ProxSSI	74.36 ± 0.62%	54.14 ± 2.91%
RMDA	**76.69 ± 0.14%**	**60.13 ± 0.24%**

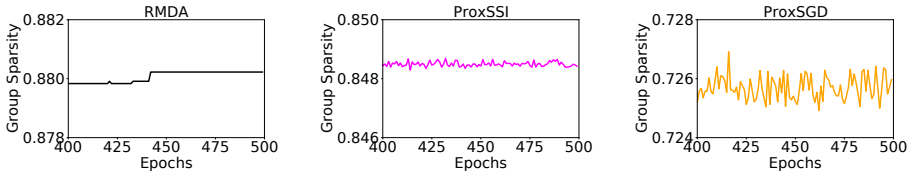


Figure 3: Group Sparsity v.s epochs of different algorithms on VGG16 of a single run.

5.4 Comparison with Pruning

Finally, we compare RMDA with a state-of-the-art pruning method RigL (Evci et al., 2020) using the PyTorch implementation of Sundar & Dwaraknath (2021). As pruning methods focus on unstructured sparsity (without groups), we use RMDA with $\psi(W) = \lambda\|W\|_1$ to have a fair comparison, and tune λ to achieve the same pre-specified sparsity level. Results of RigL and RMDA for 98% sparsity are in Table 2. For RMDA, we report results of 500 epochs, and for RigL, as its performance did not stabilize at 500 epochs, we run it with 1,000 epochs, but RMDA still consistently outperforms RigL. Our result thus shows that regularized training could be a promising alternative to pruning.

6 Conclusions

In this work, we proposed and analyzed a new algorithm, RMDA, for efficiently training structured neural networks with state-of-the-art performance. Even in the presence of data augmentation and dropout, RMDA can still achieve variance reduction and provably identify the desired structure at a stationary point using the tools of manifold identification. Experiments show that existing algorithms for the same purpose fail to find a stable stationary structure, while RMDA achieves so with no accuracy drop nor additional time cost.

Table 2: Comparison between RMDA and RigL for unstructured sparsity.

	ResNet50 with CIFAR10			ResNet50 with CIFAR100		
Algorithm	Sparsity	Accuracy	Epochs	Sparsity	Accuracy	Epochs
Dense baseline		94.81%			74.61%	
RMDA	**99.12%**	**93.43%**	500	**98.30%**	**74.69%**	500
RigL_{2X}	98.00%	93.41%	1,000	98.00%	70.88%	1,000

Reproducibility Statement

The assumptions and the theoretical results are stated in Section 3, with the proofs in Appendix B. For the experiments, our source code is available in the supplementary materials, and the detailed experiment settings are described in Appendix D.

References

Yu Bai, Yu-Xiang Wang, and Edo Liberty. ProxQuant: Quantized neural networks via proximal operators. Technical report, 2018. 1, 7

Gilles Bareilles, Franck Iutzeler, and Jérôme Malick. Newton acceleration on manifolds identified by proximal-gradient methods. Technical report, 2020. arXiv:2012.12936. 2

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? In *Machine Learning and Systems*, 2020. 3

Patrick Breheny and Jian Huang. Penalized methods for bi-level variable selection. *Statistics and its interface*, 2(3):369, 2009. 22

James V. Burke and Michael C. Ferris. Weak sharp minima in mathematical programming. 31(5):1340–1359, 1993. 21

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex SGD. In *Advances in Neural Information Processing Systems*, volume 32, pp. 15236–15245, 2019. 2

Aaron Defazio and Leon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. In *Advances in Neural Information Processing Systems*, volume 32, pp. 1755–1765, 2019. 2

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. 2

Tristan Deleu and Yoshua Bengio. Structured sparsity inducing adaptive optimizers for deep learning. Technical report, 2021. arXiv:2102.03869. 1, 7, 22, 23

John C. Duchi and Feng Ruan. Asymptotic optimality in stochastic optimization. *The Annals of Statistics*, 49(1):21–48, 2021. 5

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *International Conference on Machine Learning*, pp. 2943–2952, 2020. 3, 9

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A note on the group lasso and a sparse group lasso. Technical report, 2010. 6

Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. Technical report, 2019. arXiv:1902.09574. 3

A.M. Gupal. *Stochastic methods for solving nonsmooth extremal problems*. Naukova Dumka, 1979. 5, 16

Warren L. Hare. Identifying active manifolds in regularization problems. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pp. 261–271. Springer, 2011. 2

Warren L. Hare and Adrian S. Lewis. Identifying active constraints via partial smoothness and prox-regularity. *Journal of Convex Analysis*, 11(2):251–266, 2004. 2, 4, 19

Warren L. Hare and Adrian S. Lewis. Identifying active manifolds. *Algorithmic Operations Research*, 2(2):75–75, 2007. 2

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 9, 22

Lu Hou, Quanming Yao, and James T Kwok. Loss-aware binarization of deep networks. In *International Conference on Learning Representations*, 2017. 6

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017. 22

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. *Advances in neural information processing systems*, 29, 2016. 6

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456, 2015. 22

Samy Jelassi and Aaron Defazio. Dual averaging is surprisingly effective for deep learning optimization. Technical report, 2020. arXiv:2010.10502. 2, 4, 7

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26:315–323, 2013. 2

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. 1

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 8

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 8

Ching-pei Lee. Accelerating inexact successive quadratic approximation for regularized optimization through manifold identification. Technical report, 2020. arXiv:2012.02522. 2

Sangkyun Lee and Stephen J. Wright. Manifold identification in dual averaging for regularized stochastic online learning. *Journal of Machine Learning Research*, 13:1705–1744, 2012. 2, 4, 5

Adrian S. Lewis. Active sets, nonsmoothness, and sensitivity. *SIAM Journal on Optimization*, 13(3):702–725, 2002. 2, 4

Adrian S. Lewis and Shanshan Zhang. Partial smoothness, tilt stability, and generalized hessians. 23(1):74–94, 2013. 2, 19

Yu-Sheng Li, Wei-Lin Chiang, and Ching-pei Lee. Manifold identification for ultimately communication-efficient distributed optimization. In *International Conference on Machine Learning*, 2020. 2

Jingwei Liang, Jalal Fadili, and Gabriel Peyré. Activity identification and local linear convergence of forward–backward-type methods. 27(1):408–437, 2017a. 2

Jingwei Liang, Jalal Fadili, and Gabriel Peyré. Local convergence properties of douglas–rachford and alternating direction method of multipliers. *Journal of Optimization Theory and Applications*, 172(3):874–913, 2017b. 2

Yurii Nesterov and Vladimir Shikhman. Quasi-monotone subgradient methods for nonsmooth convex minimization. *Journal of Optimization Theory and Applications*, 165(3): 917–940, 2015. 5

Lam M. Nguyen, Katya Scheinberg, and Martin Takáč. Inexact SARAH algorithm for stochastic optimization. *Optimization Methods and Software*, 36(1):237–258, 2021. 2

Evgeni Alekseevich Nurminskii. The quasigradient method for the solving of the nonlinear programming problems. *Cybernetics*, 9(1):145–150, 1973. 4

Brendan O'donoghue and Emmanuel Candes. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15(3):715–732, 2015. 7

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. 7

Nhan H. Pham, Lam M. Nguyen, Dzung T. Phan, and Quoc Tran-Dinh. ProxSARAH: An efficient algorithmic framework for stochastic composite nonconvex optimization. *Journal of Machine Learning Research*, 21:1–48, 2020. 2

René Poliquin and R Rockafellar. Prox-regular functions in variational analysis. *Transactions of the American Mathematical Society*, 348(5):1805–1838, 1996. 4

Clarice Poon, Jingwei Liang, and Carola-Bibiane Schönlieb. Local convergence properties of SAGA/prox-SVRG and acceleration. In *International Conference on Machine Learning*, 2018. 2

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015. 8

Yifan Sun, Halyun Jeong, Julie Nutini, and Mark Schmidt. Are we there yet? manifold identification of gradient-related proximal methods. In *International Conference on Artificial Intelligence and Statistics*, pp. 1110–1119, 2019. 2, 4

Varun Sundar and Rajat Vadiraj Dwaraknath. [reproducibility report] rigging the lottery: Making all tickets winners. In *ML Reproducibility Challenge 2020*, 2021. 9, 22, 28

Wei Tao, Zhisong Pan, Gaowei Wu, and Qing Tao. Primal averaging: A new gradient evaluation step to attain the optimal individual convergence. *IEEE transactions on cybernetics*, 50(2):835–845, 2018. 4

Quoc Tran-Dinh, Nhan H. Pham, Dzung T. Phan, and Lam M. Nguyen. Hybrid stochastic gradient descent algorithms for stochastic nonconvex optimization. Technical report, 2019. 2

Sagar Verma and Jean-Christophe Pesquet. Sparsifying networks via subdifferential inclusion. In *International Conference on Machine Learning*, pp. 10542–10552, 2021. 3

Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. Spiderboost and momentum: Faster variance reduction algorithms. In *Advances in Neural Information Processing Systems*, volume 32, pp. 2406–2416, 2019. 2

Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. *Advances in neural information processing systems*, 29: 2074–2082, 2016. 1, 6, 7

Wei Wen, Yuxiong He, Samyam Rajbhandari, Minjia Zhang, Wenhan Wang, Fang Liu, Bin Hu, Yiran Chen, and Hai Li. Learning intrinsic sparse structures within long short-term memory. In *International Conference on Learning Representations*, 2018. 1

Stephen J. Wright. Accelerated block-coordinate relaxation for regularized optimization. *SIAM Journal on Optimization*, 22(1):159–186, 2012. 2

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. Technical report, 2017. arXiv:1708.07747. 8

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(88):2543–2596, 2010. 3

Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014. 2

Yang Yang, Yaxiong Yuan, Avraam Chatzimichailidis, Ruud JG van Sloun, Lei Lei, and Symeon Chatzinotas. Proxsgd: Training structured neural networks under regularization and constraints. In *International Conference on Learning Representations*, 2019. 1, 6, 7

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. 6

Jihun Yun, Aurelie C. Lozano, and Eunho Yang. A general family of stochastic proximal gradient methods for deep learning. Technical report, 2020. arXiv:2007.07484. 1, 7

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942, 2010. 22

Peng Zhao and Bin Yu. On model selection consistency of Lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006. 1

Hao Zhou, Jose M. Alvarez, and Fatih Porikli. Less is more: Towards compact cnns. In *European Conference on Computer Vision*, pp. 662–677. Springer, 2016. 6

A Organization of the Appendices

The appendices are organized as follows. The detailed proofs are given in Appendix B. Discussions of applications described in Section 4 are extended in more technical details in Appendix C. Appendix D provides experiment details for Section 5. Finally, we discuss the possibility of different regularizers for structured sparsity and conduct additional experiments in Appendix E.

B Proofs

B.1 Proof of Theorem 1

We consider an estimating sequence whose expectation upper bounds the expected objective value. In particular, define

$$B_t \coloneqq \frac{1}{2}\sum_{k=1}^{t}\frac{s_k^2}{\beta_k}\left\|\nabla f_{\xi_k}\left(W^k\right)\right\|^2, \tag{9}$$

$$\begin{aligned} g_t \coloneqq \min_{W}\Bigg(&\sum_{k=1}^{t} s_k\left(f_{\xi_k}\left(W^{k-1}\right)+\langle\nabla f_{\xi_k}\left(W^{k-1}\right), W-W^{k-1}\rangle+\psi\left(W\right)\right)+\frac{\beta_k}{2}\left\|W-W^0\right\|^2\Bigg)\\ &+B_t+\min_{\hat{W}}\left(s_1\langle\nabla f\left(W^0\right), \hat{W}-W^0\rangle+\frac{1}{2}\left\|\hat{W}-W^0\right\|^2+\psi(\hat{W})-\psi(W^0)\right), \end{aligned} \tag{10}$$

we will show through induction that

$$\mathbb{E}[g_t]\geq\alpha_t\mathbb{E}[F\left(W^{t-1}\right)],\quad\forall t>0. \tag{11}$$

First, let us define

$$M_t\left(V\right)\coloneqq\min_{W}\quad\langle V, W\rangle+\frac{\beta_t}{2}\|W-W_0\|^2+\alpha_t\psi(W). \tag{12}$$

Clearly, M_t is the Moreau envelope of $\alpha_t\Psi$, so it is differentiable with the gradient being β_t^{-1}-Lipschitz continuous. Assume that (11) holds for $t=2,\ldots,k-1$, and for $t=k$, we have from (10) and (12) that

$$\begin{aligned}\mathbb{E}_{\xi_k\sim\mathcal{D}}[g_k]-g_{k-1}=s_k\left(f\left(W^{k-1}\right)-\langle\nabla f\left(W^{k-1}\right), W^{k-1}\rangle\right)+\mathbb{E}_{\xi_k\sim\mathcal{D}}\left[M_k\left(V_k\right)\right]-\\ M_{k-1}\left(V_{k-1}\right)+\mathbb{E}_{\xi_k\sim\mathcal{D}}\left[B_k\right]-B_{k-1}+s_kf\left(W^0\right).\end{aligned} \tag{13}$$

We also get that

$$
\begin{aligned}
&\mathbb{E}_{\xi_k\sim\mathcal{D}} M_k\left(V_k\right) - M_{k-1}\left(V_{k-1}\right) \\
&= \mathbb{E}_{\xi_k\sim\mathcal{D}} M_k\left(V_k\right) - M_k\left(V_{k-1}\right) + M_k\left(V_{k-1}\right) - M_{k-1}\left(V_{k-1}\right) \\
&\geq \mathbb{E}_{\xi_k\sim\mathcal{D}} M_k\left(V_k\right) - M_k\left(V_{k-1}\right) + \frac{\beta_k-\beta_{k-1}}{2}\left\|\tilde{W}^{k-1}-W^0\right\|^2 + s_k\psi\left(\tilde{W}^{k-1}\right) && (14)\\
&\geq \mathbb{E}_{\xi_k\sim\mathcal{D}} M_k\left(V_k\right) - M_k\left(V_{k-1}\right) + \frac{\beta_k-\beta_{k-1}}{2}\left\|\tilde{W}^{k-1}-W^0\right\|^2 + \alpha_k\psi\left(W^{k-1}\right) - \alpha_{k-1}\psi\left(W^{k-2}\right) && (15)\\
&\geq \mathbb{E}_{\xi_k\sim\mathcal{D}}\left[s_k\langle\nabla f_{\xi_k}\left(W^{k-1}\right), \tilde{W}^{k-1}\rangle + \frac{s_k^2}{\beta_k}\left\|\nabla f\left(W^{k-1}\right)\right\|^2\right] + && (16)\\
&\qquad \frac{\beta_k-\beta_{k-1}}{2}\left\|\tilde{W}^{k-1}-W^0\right\|^2 + \alpha_k\psi\left(W^{k-1}\right) - \alpha_{k-1}\psi\left(W^{k-2}\right) \\
&= s_k\langle\nabla f\left(W^{k-1}\right), \tilde{W}^{k-1}\rangle + \mathbb{E}_{\xi_k\sim\mathcal{D}}\left[B_k\right] - B_{k-1} + && (17)\\
&\qquad \frac{\beta_k-\beta_{k-1}}{2}\left\|\tilde{W}^{k-1}-W^0\right\|^2 + \alpha_k\psi\left(W^{k-1}\right) - \alpha_{k-1}\psi\left(W^{k-2}\right) \\
&= \alpha_k\langle\nabla f\left(W^{k-1}\right), W^{k-1}\rangle - \alpha_{k-1}\langle\nabla f\left(W^{k-1}\right), W^{k-2}\rangle + \mathbb{E}_{\xi_k\sim\mathcal{D}}\left[B_k\right] - B_{k-1} + && (18)\\
&\qquad \frac{\beta_k-\beta_{k-1}}{2}\left\|\tilde{W}^{k-1}-W^0\right\|^2 + \alpha_k\psi\left(W^{k-1}\right) - \alpha_{k-1}\psi\left(W^{k-2}\right)
\end{aligned}
$$

where we used the definition of $\tilde{W}^{k-1}$ from (2) in (14), convexity of ψ and (3) in (15), the Lipschitz continuity of ∇M_t in (16), (9) in (17), and (3) again in (18).

Substituting (18) back into (13) and noting that $\beta_k - \beta_{k-1} > 0$ and

$$
\begin{aligned}
&\alpha_k f\left(W^{k-1}\right) - \alpha_{k-1} f\left(W^{k-2}\right) \\
&\leq s_k f\left(W^{k-1}\right) + \alpha_{k-1}\langle\nabla f\left(W^{k-1}\right), W^{k-1} - W^{k-2}\rangle,
\end{aligned}
$$

we see

$$
\mathbb{E}_{\xi_k\sim\mathcal{D}}[g_k] - g_{k-1} \geq \alpha_k\mathbb{E}_{\xi_k\sim\mathcal{D}}[F\left(W^{k-1}\right)] - \alpha_{k-1}F\left(W^{k-2}\right), \tag{19}
$$

and by taking expectation over both sides of (19), we get

$$
\mathbb{E}[g_k] - \mathbb{E}[g_{k-1}] \geq \alpha_k\mathbb{E}[F\left(W^{k-1}\right)] - \alpha_{k-1}\mathbb{E}\left[F\left(W^{k-2}\right)\right],
$$

and using (11) with $t = k-1$ from our assumption leads to that

$$
\mathbb{E}[g_k] \geq \alpha_k\mathbb{E}\left[F\left(W^{k-1}\right)\right],
$$

as desired.

It still remains to prove the basic case of $t = 1$. In this case, we have $\alpha_1 = s_1$, $\beta_1 = 1$, so

$$
\begin{aligned}
&\mathbb{E}g_1 - \alpha_1 F(W^0) \\
&= \min_W \left(s_1\langle\nabla f\left(W^0\right), W - W^0\rangle + \frac{1}{2}\left\|W - W^0\right\|^2 + \psi(W) - \psi(W^0)\right) - \\
&\qquad \min_W \left(s_1\langle\nabla f\left(W^0\right), W - W^0\rangle + \frac{1}{2}\left\|W - W^0\right\|^2 + \psi(W) - \psi(W^0)\right) = 0,
\end{aligned}
$$

finishing the induction.

Now from (10), we get that

$$\begin{aligned}
\mathbb{E}[g_t] \leq & \sum_{k=1}^{t} s_k \left(f\left(W^{k-1}\right) + \langle \nabla f_{\xi_k}\left(W^{k-1}\right), W^* - W^{k-1} \rangle + \psi\left(W^*\right) \right) + \frac{\beta_t}{2} \left\| W^* - W^0 \right\|^2 \\
& + \mathbb{E}[B_t] + s_1 \min_{\hat{W}} \quad \left(\langle \nabla f\left(W^0\right), \hat{W} - W^0 \rangle + \frac{1}{2} \left\| \hat{W} - W^0 \right\|^2 + \psi(\hat{W}) - \psi(W^0) \right) \\
\leq & \sum_{k=1}^{t} s_k \left(f\left(W^*\right) + \psi(W^*) \right) + \frac{\beta_t}{2} \left\| W^* - W^0 \right\|^2 + \mathbb{E}[B_t] \\
& + s_1 \min_{\hat{W}} \quad \left(\langle \nabla f\left(W^0\right), \hat{W} - W^0 \rangle + \frac{1}{2} \left\| \hat{W} - W^0 \right\|^2 + \psi(\hat{W}) - \psi(W^0) \right) \\
= & \alpha_k F\left(W^*\right) + \frac{\beta_t}{2} \left\| W^* - W^0 \right\|^2 + \mathbb{E}[B_t] \\
& + s_1 \min_{\hat{W}} \quad \left(\langle \nabla f\left(W^0\right), \hat{W} - W^0 \rangle + \frac{1}{2} \left\| \hat{W} - W^0 \right\|^2 + \psi(\hat{W}) - \psi(W^0) \right),
\end{aligned}$$

whose combination with (10) then proves (4).

Finally, we consider the convergence rate when $\alpha_t = t$, which implies $s_t \equiv 1$. We have from (4) that

$$\begin{aligned}
\mathbb{E}\left[F(W^t) - F^*\right] &\leq t^{-1/2} + t^{-1} C \sum_{k=0}^{t} k^{-1/2} + t^{-1}\text{constant} \\
&= O\left(t^{-1/2}\right) + t^{-1} O\left(\int_{x=0}^{t} x^{-1/2}\right) \\
&= O\left(t^{-1/2}\right) + t^{-1} O\left(t^{1/2}\right) = O\left(t^{-1/2}\right).
\end{aligned}$$

B.2 Proof of Lemma 1

Proof. Using (3), the distance between W^t and W^* can be upper bounded through the triangle inequality:

$$\begin{aligned}
\left\| W^t - W^* \right\| &= \left\| (1 - c_t)\left(W^{t-1} - W^*\right) + c_t\left(\tilde{W}^t - W^*\right) \right\| \\
&\leq c_t \left\| \tilde{W}^t - W^* \right\| + (1 - c_t) \left\| W^{t-1} - W^* \right\|. \qquad (20)
\end{aligned}$$

For any event such that $\tilde{W}^t \to W^*$, for any $\epsilon > 0$, we can find $T_\epsilon \geq 0$ such that

$$\left\| \tilde{W}^t - W^* \right\| \leq \epsilon, \quad \forall t \geq T_\epsilon.$$

Let $\delta_t := \|W^t - W^*\|$, we see from the above inequality and (20) that

$$\delta_t \leq (1 - c_t)\, \delta_{t-1} + c_t \epsilon, \quad \forall t \geq T_\epsilon.$$

By deducting ϵ from both sides, we get that

$$(\delta_t - \epsilon) \leq (1 - c_t)\left(\delta_{t-1} - \epsilon\right), \quad \forall t \geq T_\epsilon.$$

Since $\sum c_t = \infty$, we further deduce that

$$\begin{aligned}
\lim_{t \to \infty} (\delta_t - \epsilon) &\leq \prod_{t=T_\epsilon}^{\infty} (1 - c_t)\left(\delta_{T_\epsilon - 1} - \epsilon\right) \\
&\leq \prod_{t=T_\epsilon}^{\infty} \exp\left(-c_t\right)\left(\delta_{T_\epsilon - 1} - \epsilon\right) \\
&= \exp\left(- \sum_{t=T_\epsilon}^{\infty} c_t \right)\left(\delta_{T_\epsilon - 1} - \epsilon\right) = 0,
\end{aligned}$$

where in the first inequality we used the fact that $\exp(x) \geq 1 + x$ for all real number x. The result above then implies that

$$\lim_{t\to\infty} \delta_t \leq \epsilon.$$

As ϵ is arbitrary and $\delta_t \geq 0$ from the definition, we conclude that $\lim_{t\to\infty} \delta_t = 0$, which is equivalent to that $W^t \to W^*$. □

B.3 Proof of Lemma 2

Proof. We observe that

$$\begin{aligned}
\alpha_t^{-1} V^t &= \sum_{k=1}^{t} \frac{\eta_k \beta_k}{\alpha_t} \nabla f_{\xi_k}\left(W^{k-1}\right) \\
&= \frac{\alpha_{t-1}}{\alpha_t} \alpha_{t-1}^{-1} V^{t-1} + \frac{\alpha_t - \alpha_{t-1}}{\alpha_t} \nabla f_{\xi_t}\left(W^{t-1}\right) \\
&= \left(1 - \frac{\beta_t \eta_t}{\alpha_t}\right) \alpha_{t-1}^{-1} V^{t-1} + \frac{\beta_t \eta_t}{\alpha_t} \nabla f_{\xi_t}\left(W^{t-1}\right).
\end{aligned}$$

From that f is L-Lipschitz-continuously differentiable, we have that

$$\begin{aligned}
\left\|\mathbb{E}_{\xi_{t+1}\sim\mathcal{D}}\left[\nabla f_{\xi_{t+1}}\left(W^t\right)\right] - \mathbb{E}_{\xi_t\sim\mathcal{D}}\left[\nabla f_{\xi_t}\left(W^{t-1}\right)\right]\right\| &= \left\|\nabla f\left(W^t\right) - f\left(W^{t-1}\right)\right\| \\
&\leq L\left\|W^t - W^{t-1}\right\|. \qquad (21)
\end{aligned}$$

Therefore, (5) and (21) imply that

$$0 \leq \frac{\left\|\mathbb{E}_{\xi_{t+1}\sim\mathcal{D}}\left[\nabla f_{\xi_{t+1}}\left(W^t\right)\right] - \mathbb{E}_{\xi_t\sim\mathcal{D}}\left[\nabla f_{\xi_t}\left(W^{t-1}\right)\right]\right\|}{\beta_t \eta_t \alpha_t^{-1}} \leq L \frac{\left\|W^t - W^{t-1}\right\|}{\beta_t \eta_t \alpha_t^{-1}} \xrightarrow{\text{a.s.}} 0,$$

which together with the sandwich lemma shows that

$$\frac{\left\|\mathbb{E}_{\xi_{t+1}\sim\mathcal{D}}\left[\nabla f_{\xi_{t+1}}\left(W^t\right)\right] - \mathbb{E}_{\xi_t\sim\mathcal{D}}\left[\nabla f_{\xi_t}\left(W^{t-1}\right)\right]\right\|}{\beta_t \eta_t \alpha_t^{-1}} \xrightarrow{\text{a.s.}} 0. \qquad (22)$$

Therefore, the first two conditions of (5) together with (22) and the bounded variance assumption satisfy the requirements of (Gupal, 1979, Chapter 2.4, Theorem 4.1), so the conclusion of almost sure convergence hold.

For the convergence in $\mathcal{L}_2$ part, we first define $m^t := \alpha_t^{-1} V_t$ and $\tau_t := \beta_t \eta_t \alpha_t^{-1}$ for notational ease. Consider $\left\|m^{t+1} - \nabla F(W^t)\right\|^2$, we have from the update rule in Algorithm 1 that

$$\begin{aligned}
&\left\|m^{t+1} - \nabla F(W^t)\right\|^2 \\
&= \left\|(1-\tau_t)m^t + \tau_t \nabla f_{\xi_{t+1}}(W^t) - \nabla F(W^t)\right\|^2 \\
&= \left\|(1-\tau_t)\left(m^t - \nabla F(W^t)\right) + \tau_t\left(\nabla f_{\xi_{t+1}}(W^t) - \nabla F(W^t)\right)\right\|^2 \\
&= (1-\tau_t)^2\left\|m^t - \nabla F(W^t)\right\|^2 + \tau_t^2\left\|\nabla f_{\xi_{t+1}}(W^t) - \nabla F(W^t)\right\|^2 \\
&\qquad + 2\tau_t(1-\tau_t)\langle m^t - \nabla F(W^t),\, \nabla f_{\xi_{t+1}}(W^t) - \nabla F(W^t)\rangle \\
&= (1-\tau_t)^2\left\|\left(m^t - \nabla F\left(W^{t-1}\right)\right) + \left(\nabla F\left(W^{t-1}\right) - \nabla F\left(W^t\right)\right)\right\|^2 \qquad (23) \\
&\qquad + \tau_t^2\left\|\nabla f_{\xi_{t+1}}(W^t) - \nabla F(W^t)\right\|^2 + 2\tau_t(1-\tau_t)\langle m^t - \nabla F(W^t),\, \nabla f_{\xi_{t+1}}(W^t) - \nabla F(W^t)\rangle.
\end{aligned}$$

Let $\{\mathcal{F}_t\}_{t\geq 0}$ denote the natural filtration of $\{(m^t, W^t)\}_{t\geq 0}$. Namely, $\mathcal{F}_t$ records the information of W^0, $\{c_i\}_{i=0}^{t-1}$, $\{\eta_i\}_{i=0}^{t-1}$, and $\{\xi_i\}_{i=1}^{t}$. By defining $U_t := \left\|m^t - \nabla F(W^{t-1})\right\|^2$ and taking expectation over (23) conditional on $\mathcal{F}_t$, we obtain from $\mathbb{E}\left[\nabla f_{\xi_{t+1}}(W^t) \mid \mathcal{F}_t\right] = \nabla F(W^t)$ that

$$\begin{aligned}
\mathbb{E}\left[U_{t+1} \mid \mathcal{F}_t\right] = (1-\tau_t)^2&\left\|\left(m^t - \nabla F\left(W^{t-1}\right)\right) + \left(\nabla F\left(W^{t-1}\right) - \nabla F\left(W^t\right)\right)\right\|^2 \\
&+ \tau_t^2 \mathbb{E}\left[\left\|\nabla f_{\xi_t}(W^t) - \nabla F(W^t)\right\|^2 \mid \mathcal{F}_t\right]. \qquad (24)
\end{aligned}$$

From the last condition in (5) and the Lipschitz continuity of ∇F, there are random variables $\{\epsilon_t\}$ and $\{u_t\}$ such that $\|u_t\| = 1$, $\epsilon_t \geq 0$, and $\nabla F(W^{t-1}) - \nabla F(W^t) = \tau_t \epsilon_t u_t$ for all $t > 0$, with $\epsilon_t \downarrow 0$ almost surely. We thus obtain that

$$\begin{aligned}
&\left\|m^t - \nabla F(W^{t-1}) + \nabla F(W^{t-1}) - \nabla F(W^t)\right\|^2 \\
&= \left\|m^t - \nabla F(W^{t-1}) + \tau_t \epsilon_t u_t\right\|^2 \\
&= (1+\tau_t)^2 \left\| \frac{1}{1+\tau_t}\left(m^t - \nabla F\left(W^{t-1}\right)\right) + \frac{\tau_t}{1+\tau_t}\epsilon_t u_t \right\|^2 \\
&\leq (1+\tau_t)^2 \left(\frac{1}{1+\tau_t} U_t + \frac{\tau_t}{1+\tau_t} {\epsilon_t}^2 \right), \qquad (25)
\end{aligned}$$

where we used Jensen's inequality and the convexity of $\|\cdot\|^2$ in the last inequality. By substituting (25) back into (24), we obtain

$$\begin{aligned}
&\mathbb{E}\left[U_{t+1} | \mathcal{F}_t\right] \\
&\leq (1-\tau_t)^2(1+\tau_t)U_t + (1-\tau_t)^2(1+\tau_t)\tau_t {\epsilon_t}^2 + {\tau_t}^2 \mathbb{E}\left[\left\|\nabla f_{\xi_t}(W^t) - \nabla F(W^t)\right\|^2 \mid \mathcal{F}_t\right] \\
&\leq (1-\tau_t)(U_t + \tau_t {\epsilon_t}^2) + {\tau_t}^2 \mathbb{E}\left[\left\|\nabla f_{\xi_t}(W^t) - \nabla F(W^t)\right\|^2 \mid \mathcal{F}_t\right] \\
&\leq (1-\tau_t)U_t + \tau_t {\epsilon_t}^2 + {\tau_t}^2 \mathbb{E}\left[\left\|\nabla f_{\xi_t}(W^t) - \nabla F(W^t)\right\|^2 \mid \mathcal{F}_t\right]. \qquad (26)
\end{aligned}$$

For the last term in (26), we notice that

$$\begin{aligned}
\mathbb{E}\left[\left\|\nabla f_{\xi_t}(W^t) - \nabla F(W^t)\right\|^2 \mid \mathcal{F}_t\right] &\leq 2\left(\mathbb{E}\left[\left\|\nabla f_{\xi_t}(W^t)\right\|^2\right] + \left\|\nabla F(W^t)\right\|^2\right) \\
&\leq 2\left(C + \left\|\nabla F(W^t)\right\|^2\right), \qquad (27)
\end{aligned}$$

where the last inequality is from the bounded variance assumption. Since by assumption the $\{W^t\}$ lies in a bounded set K, we have that for any point $W^* \in K$, $W^t - W^*$ is upper bounded, and thus $\|\nabla F(W^t) - \nabla F(W^*)\|$ is also bounded, implying that $\|\nabla F(W^t)\|^2 \leq C_2$ for some $C_2 \geq 0$. Therefore, (27) further leads to

$$\mathbb{E}\left[\left\|\nabla f_{\xi_t}(W^t) - \nabla F(W^t)\right\|^2 \mid \mathcal{F}_t\right] \leq C_3 \qquad (28)$$

for some $C_3 \geq 0$.

Now we further take expectation on (26) and apply (28) to obtain

$$\mathbb{E}U_{t+1} \leq (1-\tau_t)\mathbb{E}U_t + \tau_t {\epsilon_t}^2 + {\tau_t}^2 C_3 = (1-\tau_t)\,\mathbb{E}U_t + \tau_t\left(\epsilon_t^2 + \tau_t C_3\right). \qquad (29)$$

Note that the third implies $\epsilon_t \downarrow 0$, so this together with the second condition that $\tau_t \downarrow 0$ means $\epsilon_t^2 + \tau_t C_3 \downarrow 0$ as well, and thus for any $\delta > 0$, we can find $T_\delta \geq 0$ such that $\epsilon_t^2 + \tau_t C_3 \leq \delta$ for all $t \geq T_\delta$. Thus, (29) further leads to

$$\mathbb{E}U_{t+1} - \delta \leq (1-\tau_t)\mathbb{E}U_t + \tau_t\delta - \delta = (1-\tau_t)\left(\mathbb{E}U_t - \delta\right), \forall t \geq T_\delta. \qquad (30)$$

This implies that $(\mathbb{E}U_t - \delta)$ becomes a decreasing sequence starting from $t \geq T_\delta$, and since $U_t \geq 0$, this sequence is lower bounded by $-\delta$, and hence it converges to a certain value. By recursion of (30), we have that

$$\mathbb{E}U_t - \delta \leq \prod_{i=T_\delta}^{t}(1-\tau_i)\left(\mathbb{E}U_{T_\delta} - \delta\right),$$

and from the well-known inequality $(1+x) \leq \exp^x$ for all $x \in \mathcal{R}$, the above result leads to

$$\mathbb{E}U_t - \delta \leq \exp\left(-\sum i = {T_\delta}^t \tau_i\right)\left(\mathbb{E}U_{T_\delta} - \delta\right).$$

By letting t approach infinity and noting that the first condition of (5) indicates

$$\sum_{t=k}^{\infty} \tau_t = \infty$$

for any $k \geq 0$, we see that

$$-\delta \leq \lim_{t\to\infty} \mathbb{E}U_t - \delta \leq \exp\left(-\sum_{i=T_\delta}^{\infty} \tau_i\right)(\mathbb{E}U_{T_\delta} - \delta) = 0. \tag{31}$$

As δ is arbitrary, by taking $\delta \downarrow 0$ in (31) and noting the nonnegativity of U_t, we conclude that $\lim \mathbb{E}U_t = 0$, as desired. This proves the last result in Lemma 2. □

B.4 Proof of Theorem 2

Proof. Using Lemma 2, we can view $\alpha_t^{-1}V^t$ as $\nabla f(W^t)$ plus some noise that asymptotically decreases to zero with probability one:

$$\alpha_t^{-1}V_t = \nabla f(W^t) + \epsilon_t, \quad \|\epsilon_t\| \xrightarrow{\text{a.s.}} 0. \tag{32}$$

We use (32) to rewrite the optimality condition of (2) as (also see Line 5 of Algorithm 1)

$$-\left(\nabla f\left(W^t\right) + \epsilon_t + \beta_t\alpha_t^{-1}\left(\tilde{W}^t - W^0\right)\right) \in \partial\psi\left(\tilde{W}^t\right). \tag{33}$$

Now we consider $\partial F(\tilde{W}^t)$. Clearly from (33), we have that

$$\nabla f\left(\tilde{W}^t\right) - \nabla f\left(W^t\right) - \epsilon_t - \beta_t\alpha_t^{-1}\left(\tilde{W}^t - W^0\right) \in \partial\nabla f\left(\tilde{W}^t\right) + \psi\left(\tilde{W}^t\right) = \partial F\left(\tilde{W}^t\right). \tag{34}$$

Now we consider the said event that $\tilde{W}^t \to W^*$ for a certain W^*, and let us define this event as $\mathcal{A} \subseteq \Omega$. From Lemma 1, we know that $W^t \to W^*$ as well under $\mathcal{A}$. Let us define $\mathcal{B} \subseteq \Omega$ as the event of $\epsilon_t \to 0$, then we know that since $P(\mathcal{A}) > 0$ and $P(\mathcal{B}) = 1$, where P is the probability function for events in Ω, $P(\mathcal{A} \cap \mathcal{B}) = P(\mathcal{A})$. Therefore, conditional on the event of $\mathcal{A}$, we have that $\epsilon_t \xrightarrow{\text{a.s.}} 0$ still holds. Now we consider any realization of $\mathcal{A} \cap \mathcal{B}$. For the right-hand side of (34), as $\tilde{W}^t$ is convergent and $\beta_t\alpha_t^{-1}$ decreases to zero, by letting t approach infinity, we have that

$$\lim_{t\to\infty} \epsilon_t + \beta_t\alpha_t^{-1}\left(\tilde{W}^t - W^0\right) = 0 + 0\left(W^* - W^0\right) = 0.$$

By the Lipschitz continuity of ∇f, we have from (3) and (5) that

$$0 \leq \left\|\nabla f\left(\tilde{W}^t\right) - \nabla f\left(W^t\right)\right\| \leq L\left\|W^t - \tilde{W}^t\right\|.$$

As $\{W^t\}$ and $\{\tilde{W}^t\}$ converge to the same point, we see that $\left\|W^t - \tilde{W}^t\right\| \to 0$, so $\nabla f\left(\tilde{W}^t\right) - \nabla f(W^t)$ also approaches zero. Hence, the limit of the right-hand side of (34) is

$$\lim_{t\to\infty} \nabla f\left(\tilde{W}^t\right) - \left(\nabla f\left(W^t\right) + \epsilon_t + \beta_t\alpha_t^{-1}\left(\tilde{W}^t - W^0\right)\right) = 0. \tag{35}$$

On the other hand, for the left-hand side of (34), the outer semicontinuity of $\partial\psi$ at W^* and the continuity of ∇f show that

$$\lim_{t\to\infty} \nabla f(\tilde{W}^t) + \partial\psi(\tilde{W}^t) \subseteq \partial\nabla f(W^*) + \psi\left(W^*\right) = \partial F(W^*). \tag{36}$$

Substituting (35) and (36) back into (34) then proves that $0 \in \partial F(W^*)$ and thus $W^* \in \mathcal{Z}$. □

B.5 Proof of Theorem 3

Proof. Our discussion in this proof are all under the event that $\tilde{W}^t \to W^*$. From the argument in Appendix B.4, we can view $\alpha_t^{-1}V^t$ as $\nabla f(W^t)$ plus some noise that asymptotically decreases to zero with probability one as shown in (32). From Lemma 1, we know that $W^t \to W^*$. From (33), there is $U^t \in \partial\psi\left(\tilde{W}^t\right)$ such that

$$U^t = -\alpha_t^{-1}V^t + \alpha_t^{-1}\beta_t\left(\tilde{W}^t - W^0\right). \tag{37}$$

Moreover, we define

$$\gamma_t \coloneqq W^t - \tilde{W}^t. \tag{38}$$

By combining (37)–(38) with (32), we obtain

$$\begin{aligned}
&\min_{Y\in\partial F(\tilde{W}^t)} \|Y\| \\
\leq& \left\|\nabla f\left(\tilde{W}^t\right)+U^t\right\| \\
=& \left\|\nabla f\left(\tilde{W}^t\right)-\nabla f\left(W^t\right)-\epsilon_t-\alpha_t^{-1}\beta_t\left(\tilde{W}^t-W^0\right)\right\| \\
\leq& \left\|\nabla f\left(\tilde{W}^t\right)-\nabla f\left(W^t\right)\right\|+\|\epsilon_t\|+\alpha_t^{-1}\beta_t\left\|\tilde{W}^t-W^0\right\| \\
\leq& L\|\gamma_t\|+\|\epsilon_t\|+\alpha_t^{-1}\beta_t\left(\left\|W^*-\tilde{W}^t\right\|+\left\|W^0-W^*\right\|\right),
\end{aligned} \tag{39}$$

where we used the Lipschitz continuity of ∇f and the triangle inequality in the last inequality.

We now separately bound the terms in (39). From that $W^t \to W^*$ and $\tilde{W}^t \to W^*$, it is straightforward that $\|\gamma_t\| \to 0$. The second term decreases to zero almost surely according to (32) and the argument in Appendix B.4. For the last term, since $\alpha_t^{-1}\beta_t \to 0$, and $\left\|\tilde{W}^t - W^*\right\| \to 0$, we know that

$$\alpha_t^{-1}\beta_t\left\|W^0-W^*\right\| \to 0, \quad \alpha_t^{-1}\beta_t\left\|\tilde{W}^t-W^*\right\| \to 0.$$

Therefore, we conclude from the above argument and (39) that

$$\min_{Y\in\partial F(\tilde{W}^t)} \|Y\| \xrightarrow{\text{a.s.}} 0.$$

As f is smooth with probability one, we know that if ψ is partly smooth at W^* relative to $\mathcal{M}$, then so is $F = f + \psi$ with probability one. Moreover, Lipschitz-continuously differentiable functions are always prox-regular, and the sum of two prox-regular functions is still prox-regular, so F is also prox-regular at W^* with probability one. Following the argument identical to that in Appendix B.4, we know that these probability one events are still probability one conditional on the event of $\tilde{W}^t \to W^*$ as this event has a nonzero probability. As $\tilde{W}^t \to W^*$ and $\nabla f(\tilde{W}^t) + U^t \xrightarrow{\text{a.s.}} 0 \in \partial F(W^*)$ (the inclusion is from (6)), we have from the subdifferential continuity of ψ and the smoothness of f that $F(\tilde{W}^t) \xrightarrow{\text{a.s.}} F(W^*)$. Since we also have $\tilde{W}^t \to W^*$ and $\min_{Y\in\partial F(\tilde{W}^t)} \|Y\| \xrightarrow{\text{a.s.}} 0$, clearly

$$\left(\tilde{W}^t, F\left(W^t\right), \min_{Y\in\partial F\left(\tilde{W}^t\right)} \|Y\|\right) \xrightarrow{\text{a.s.}} \left(W^*, F(W^*), 0\right). \tag{40}$$

Therefore, (40) and (6) together with the assumptions on ψ at W^* imply that with probability one, all conditions of Theorem 4.10 of Lewis & Zhang (2013), or (Theorem 5.3 Hare & Lewis, 2004), are satisfied, so from it, (7) holds almost surely, conditional on the event of $\tilde{W}^t \to W^*$. □

C Additional Discussions on Applications

We now discuss in more technical details the applications in Section 4.1, especially regarding how the regularizers satisfy the properties required by our theory.

C.1 Structured Sparsity

We start our discussion with the simple ℓ_1 norm as the warm-up for the group-LASSO norm. It is clear that $\|W\|_1$ is a convex function that is finite everywhere, so it is prox-regular, subdifferentially continuous, and regular everywhere, hence we just need to discuss about the remaining parts in Definition 1. Consider a problem with dimension $n > 0$. Note that

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|,$$

and the absolute value is smooth everywhere except the point of origin. Therefore, it is clear that $\|x\|_1$ is locally smooth if $x_i \neq 0$ for all i. For any point x^*, when there is an index

set I such that $x_i^* = 0$ for all $i \in I$ and $x_i^* \neq 0$ for $i \notin I$, we see that the part of the norm corresponds to I^C (the complement of I):

$$\sum_{i \in I^C} |x_i^*|$$

is locally smooth around x^*. Without loss of generality, we assume that $I = \{1, 2, \ldots, k\}$ for some $k \geq 0$, then the subdifferential of $\|x\|_1$ at x^* is the set

$$\{\mathrm{sgn}(x_1)\} \times \cdots \times \{\mathrm{sgn}(x_k)\} \times [-1, 1]^{n-k}, \tag{41}$$

and clearly if we move from x^* along any direction $y \coloneqq (y_1, \ldots, y_k, 0, \ldots, 0)$ with a small step, the function value changes smoothly as it is a linear function, satisfying the first condition of Definition 1. Along the same direction y with a small enough step, the set of subdifferential remains the same, so the continuity of subdifferential requirement holds. We can also observe from the above argument that the manifold should be $\mathcal{M}_{x^*} = \{x \mid x_i = 0, \forall i \in I\}$, and clearly it is a subspace of $\mathbb{R}^n$ with its normal space at x^* being $N \coloneqq \{y \mid \langle x^*, y\rangle = 0\} = \{y \mid y_i = 0, \forall i \in I^C\}$, which is clearly the affine span of (41) with the translation being $(\mathrm{sgn}(x_1) \times \cdots \times \mathrm{sgn}(x_k), 0, \ldots, 0)$. Moreover, indeed the manifolds are low dimensional ones, and for iterates approaching x^*, staying in this active manifold means that the (unstructured) sparsity of the iterates is the same as the limit point x^*. We also provide a graphical illustration of $\|x\|_1$ with $n = 2$ in Fig. 4. We can observe that for any x with $x_1 \neq 0$ and $x_2 \neq 0$, the function is smooth locally around any point, meaning that $\|x\|_1$ is partly smooth relative to the whole space at x (so actually smooth locally around x). For x with $x_1 = 0$, the function value corresponds to the sharp valley in the graph, and we can see that the function is smooth along the valley, and this valley corresponds to the one-dimensional manifold $\{x \mid x_1 = 0\}$ for partial smoothness.

Next, we use the same graph to illustrate the importance of manifold identification. Consider that the red point $x^* = (0, 1.5)$ is the limit point of the iterates of a certain algorithm, and the yellow points and black points are two sequences that both converge to x^*. If the iterates of the algorithm are the black points, then clearly except for the limit point itself, all iterates are nonsparse, and thus the final output of the algorithm is also nonsparse unless we can get to exactly the limit point within finite iterations (which is usually impossible for iterative methods). On the other hand, if the iterates are the yellow points, this is the case that the manifold is identified, because all points sit in the valley and enjoy the same sparsity pattern as the limit point x^*. This is why we concern about manifold identification when we solve regularized optimization problems.

From this example, we can also see an explanation for why our algorithm with the property of manifold identification performs better than other methods without such a property. Consider a Euclidean space any point x^* with an index set I such that $x_I^* = 0$ and $|I| > 0$. This means that x^* has at least one coordinate being zero, namely x^* contains sparsity. Now let

$$\epsilon_0 \coloneqq \min_{i \in I^C} \quad |x_i^*|,$$

then from the definition of I, $\epsilon_0 > 0$. Fro any sequence $\{x^t\}$ converging to x^*, for any $\epsilon \in (0, \epsilon_0)$, we can find $T_\epsilon \geq 0$ such that

$$\left\|x^t - x^*\right\|_2 \leq \epsilon, \quad \forall t \geq T_\epsilon.$$

Therefore, for any $i \notin I$, we must have that $x_i^t \neq 0$ for all $t \geq T_\epsilon$. Otherwise, $\|x^t - x^*\|_2 \geq \epsilon_0$, but $\epsilon_0 > \epsilon \geq \|x^t - x^*\|_2$, leading to a contradiction. On the other hand, for any $i \in I$, we can have $x_i^t \neq 0$ for all t without violating the convergence. That being said, for any sequence converging to x^*, eventually the iterates cannot be sparser than x^*, so the sparsity level of x^*, or of its active manifold, is the local upper bound for the sparsity level of points converging to x^*. Therefore, if iterates of two algorithms converge to the same limit point, the one with a proven manifold identification ability clearly will produce a higher sparsity level.

Similar to our example here, in applications other than sparsity, iterates converging to a limit point dwell on super-manifolds of the active manifold, and the active manifold is the

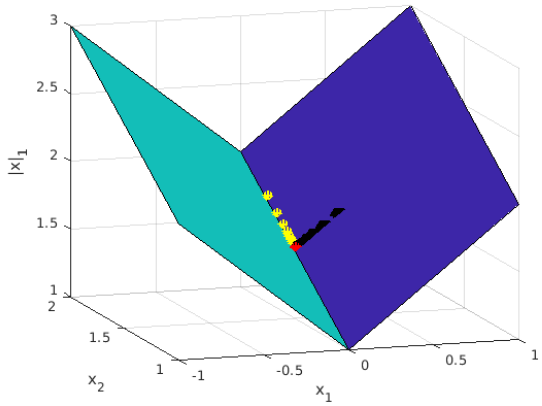


Figure 4: An illustration of partial smoothness of the ℓ_1 norm.

minimum one that locally describes points with the same structure as the limit point, and thus identifying this manifold is equivalent to finding the locally most ideal structure of the application.

Now back to the sparsity case. One possible concern is the case that the limit point is $(0, 0)$ in the two-dimension example. In this case, the manifold is the 0-dimensional subspace $\{0\}$. If this is the case and manifold identification can be ensured, it means that limit point itself can be found within finite iterations. This case is known as the weak sharp minima (Burke & Ferris, 1993) in nonlinear optimization, and its associated finite termination property is also well-studied.

For this example, We also see that $\|x\|_1$ is partly smooth at any point x^*, but the manifold differs with x^*. This is a specific benign example, and in other cases, partial smoothness might happen only locally at some points of interest instead of everywhere.

Next, we further extend our argument above to the case of (8). This can be viewed as the ℓ_1 norm for each group and we can easily obtain similar results. Again, since the group-LASSO norm is also convex and finite everywhere, prox-regularity, regularity, and subdifferential continuity are not issues at all. For the other properties, we consider one group first, then the group-LASSO norm reduces to the ℓ_2 norm. Clearly, $\|x\|_2$ is smooth locally if $x \neq 0$, with the gradient being $x/\|x\|_2$, but it is nonsmooth at the point $x = 0$, where the subdifferential is the unit ball. This is very similar to the absolute value, whose subdifferential at 0 is the interval $[-1, 1]$. Thus, we can directly apply similar arguments above, and conclude that for any W^*, (8) is partly smooth at W^* with respect to the manifold $\mathcal{M}_{W^*} = \{W \mid W_{\mathcal{I}_g} = 0, \forall g : W^*_{\mathcal{I}_g} = 0\}$, which is again a lower-dimensional subspace. Therefore, the manifold of defining the partial smoothness for the group-LASSO norm exactly corresponds to its structured sparsity pattern.

C.2 Binary Neural Networks

We continue to consider the binary neural network problem. For easier description, for the Euclidean space $\mathcal{E}$ we consider, we will use a vectorized representation for $W, A \in \mathcal{E}$ such that the elements are enumerated as $W_1, \ldots, W_n$ and $\alpha_1, \ldots, \alpha_n$. The corresponding optimization problem can therefore be written as

$$\min_{W,A\in\mathcal{E}} \quad \mathbb{E}_{\xi\sim\mathcal{D}}\left[f_\xi\left(W\right)\right] + \lambda \sum_{i=1}^{n} \Big(\alpha_i \left(w_i + 1\right)^2 + \left(1 - \alpha_i\right)\left(w_i - 1\right)^2 + \delta_{[0,1]}\left(\alpha_i\right)\Big), \tag{42}$$

where given any set C, δ_C is the indicator function of C, defined as

$$\delta_C(x) = \begin{cases} 0 & \text{if } x \in C, \\ \infty & \text{else.} \end{cases}$$

We see that except for the indicator function part, the objective is smooth, so the real partly smooth term that we treat as the regularizer is

$$\Phi(\alpha) \coloneqq \sum_{i=1}^{n} \delta_{[0,1]}(\alpha_i).$$

We note that for $\alpha_i \in (0,1)$, the value of $\delta_{[0,1]}(\alpha_i)$ remains a constant zero in a neighborhood of α_i, and for $\alpha_i \notin [0,1]$, the indicator function is also constantly infinite within a neighborhood. Thus, the point of nonsmoothness, happens only at $\alpha_i \in \{0,1\}$, and similar to our discussion in the previous subsection, Φ is partly smooth along directions that we fix those α_i at the boundary (namely, being either 0 or 1) unchanged. The identified manifold therefore corresponds to the entries of α_i that are fixed at 0 or 1, and this can serve as the indicator for the desired binary pattern in this task.

D Experiment Setting Details

For the weights w_g of each group in (8), we consider the following setting. Let L_i be a collection of index sets that belong to the i-th layer in the network, and denote

$$N_{L_i} \coloneqq \sum_{j \in L_i} |\mathcal{I}_j|$$

the number of parameters in this layer, we set $w_g = \sqrt{N_{L_i}}$ for all $g \in L_i$ for all i in our experiments.

All ProxSSI parameter settings expect the regularization weight and the learning rate schedule follow the default values in their package.

The detailed model and algorithm settings as well as statistics of the synthetic data of Section 5.1 are shown in Tables 3 and 4. Tables 5 to 7 provide the detailed setting of Section 5.2 while Tables 8 to 12 are for Section 5.3. For the modified VGG16 model, we follow Deleu & Bengio (2021) to eliminate all fully-connected layers except the output layer, and add one batch-norm layer (Ioffe & Szegedy, 2015) after each convolutional layer to simulate modern CNNs like those proposed in He et al. (2016); Huang et al. (2017). For ResNet50 in the structured sparsity experiment in Section 5.3, our version of ResNet50 is the one constructed by the publicly available script at `https://github.com/weiaicunzai/pytorch-cifar100`.

In the unstructured sparsity experiment presented in Section 5.4, for better comparison with existing works in the literature of pruning, we adopt the version of ResNet50 used by Sundar & Dwaraknath (2021).[3] Table 13 provides detailed settings of Section 5.4.

E Other Regularizers for Possibly Better Group Sparsity and Generalization

A downside of (8) is that it pushes all groups toward zeros and thus introduces bias in the final model. For its remedy, minimax concave penalty (MCP, Zhang, 2010) is then proposed to penalize only the groups whose norm is smaller than a user-specified threshold. More precisely, given hyperparameters $\lambda \geq 0, c \geq 1$, the one-dimensional MCP is defined by

$$\mathrm{MCP}(w) \coloneqq \begin{cases} \lambda|w| - \frac{w^2}{2c} & \text{if} |w| < c\lambda, \\ \frac{c\lambda^2}{2} & \text{if} |w| \geq c\lambda. \end{cases}$$

Given the grouping $\mathcal{G}$, MCP is then defined as (Breheny & Huang, 2009)

$$\mathrm{MCP}\left(W_{\mathcal{I}_g}\right) \coloneqq \begin{cases} \lambda|\mathcal{I}_g|\left\|W_{\mathcal{I}_g}\right\|^2 - \frac{\left\|W_{\mathcal{I}_g}\right\|^2}{2c} & \text{if } \left\|W_{\mathcal{I}_g}\right\| < c\lambda, \\ \frac{c\lambda^2|\mathcal{I}_g|^2}{2} & \text{if} \left\|W_{\mathcal{I}_g}\right\| \geq c\lambda, \end{cases} \quad \psi(W) \coloneqq \sum_{g=1}^{|\mathcal{G}|} MCP\left(W_{\mathcal{I}_g}\right).$$

[3] `https://github.com/varun19299/rigl-reproducibility`.

Table 3: Details of experimental settings of logistic regression on the synthetic data experiment in Section 5.1.

Parameter	Value
Data dimension	$1 \times 28 \times 28$
Number of training data	60,000
Number of testing data	10,000
Model	Logistic regression
Loss function	Cross entropy
regularizer	Group LASSO
Group LASSO weight	1
Total epochs	500
RMDA	
Initial learning rate	10^{-5}
Initial momentum	10^{-2}
Learning rate schedule	multiplied by 0.1 when epoch reaches $50, 100, 150$
Momentum schedule	multiplied by 10 when epoch reaches $50, 100$
Restart epochs	$50, 100, 150$
ProxSSI	
Learning rate schedule	$\eta(\text{epoch}) = 10^{-5-\lfloor \text{epoch}/50 \rfloor}$
momentum	(0.9, 0.999)
ProxSGD	
Learning rate schedule	$\eta(\text{epoch}) = 10^{-6-\lfloor \text{epoch}/50 \rfloor}$
momentum	0.9

It is shown in Deleu & Bengio (2021) that group MCP regularization may simultaneously provide higher group sparsity and better validation accuracy than the group LASSO norm in vision and language tasks. Another possibility to enhance sparsity is to add another ℓ_1-norm or entry-wise MCP regularization to the group-level regularizer. The major drawback of these approaches is the requirement of additional hyperparameters, and we prefer simpler approaches over those with more hyperparameters, as hyperparameter tuning in the latter can be troublesome for users with limited computational resources, and using a simpler setting can also help us to focus on the comparison of the algorithms themselves. The experiment in this subsection is therefore only for illustrating that these more complicated regularizers can be combined with RMDA if the user wishes, and such regularizers might lead to better results. Therefore, we train a simple CNN (LeNet5) on the MNIST dataset with such regularizers using RMDA and display the respective performance of various regularization schemes in Fig. 5. In this figure, group LASSO is abbreviated as GLASSO, ℓ_1-norm plus a group LASSO norm is abbreviated as L1GLASSO, group MCP is abbreviated as GMCP, and element-wise MCP plus group MCP is abbreviated as L1+GMCP. Our results exemplify that different regularization schemes might have different benefits on one of the criterion with proper hyperparameters tuning. The detailed numbers are reported in Table 14 and the experiment settings can be found in Table 15.

Table 4: Details of experimental settings of Convolution Neural Network(ConvNet) on synthesized data experiments

Parameter	Value
Data dimension	$1 \times 28 \times 28$
Number of training data	60,000
Number of testing data	10,000
Model	Convolution Neural Network
Number of layers	2
Number of convolutional layers	1
Number of fully-connected layers	1
Size of convolutional kernels	3×3
Number of output filters	3
Kernel size	3×3
Type of pooling	max pooling
Activation function for convolution/output layer	relu/softmax
Loss function	Cross entropy
Eegularizer	Group LASSO
Group LASSO weight	3
Total epochs	300
RMDA	
Initial learning rate	10^{-6}
Initial momentum	10^{-2}
Learning rate schedule	multiplied by 0.1 when epoch reaches $40, 80$
Momentum schedule	multiplied by 10 when epoch reaches $40, 80$
Restart epochs	$40, 80$
ProxSSI	
Learning rate schedule	$\eta(\text{epoch}) = 10^{-6-\lfloor \text{epoch}/40 \rfloor}$
Momentum	(0.9, 0.999)
ProxSGD	
Learning rate schedule	$\eta(\text{epoch}) = 10^{-6-\lfloor \text{epoch}/40 \rfloor}$
Momentum	0.9

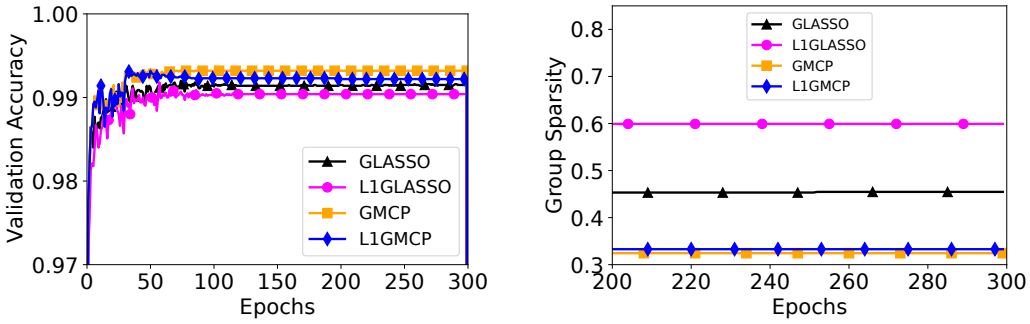


Figure 5: Comparison between group LASSO, L1+group LASSO, Group MCP, L1+Group MCP

Table 5: Details of the experimental settings of logistic regression on MNIST in Section 5.2.

Parameter	Value
Data set	MNIST
Model	Logistic regression
Loss function	Cross entropy
Regularization function	Group LASSO
Total epochs	100
ProxSGD	
Regularization weight	10^{-4}
Learning rate schedule	$\eta(\text{epoch}) = 10^{-1-\lfloor \text{epoch}/10 \rfloor}$
Momentum schedule	0.9
ProxSSI	
Regularization weight	1×10^{-4}
Learning rate schedule	$\eta(\text{epoch}) = 10^{-3-\lfloor \text{epoch}/10 \rfloor}$
RMDA	
Regularization weight	10^{-4}
Learning rate schedule	$\eta(\text{epoch}) = \max(10^{-3}, 10^{-1-\lfloor \text{epoch}/10 \rfloor})$
Momentum schedule	$c(\text{epoch}) = \min(10^{-2+\lfloor \text{epoch}/10 \rfloor}, 1)$
Restart epochs	$10, 20$

Table 6: Details of the experimental settings of the multi-layer fully-connected NN on FashionMNIST in Section 5.2.

Parameter	Value
Data set	FashionMNIST
Model	Fully-connected NN (Table 7)
Loss function	Cross entropy
Regularization function	Group LASSO
Total epochs	300
ProxSGD	
Regularization weight	10^{-4}
Learning rate schedule	$\eta(\text{epoch}) = 10^{-2-\lfloor \text{epoch}/30 \rfloor}$
Momentum schedule	0.9
ProxSSI	
Regularization weight	$7\ 10^{-7}$
Learning rate schedule	$\eta(\text{epoch}) = 10^{-4-\lfloor \text{epoch}/30 \rfloor}$
RMDA	
Regularization weight	7×10^{-6}
Learning rate schedule	$\eta(\text{epoch}) = \max(10^{-6}, 10^{-1-\lfloor \text{epoch}/30 \rfloor})$
Momentum schedule	$c(\text{epoch}) = \min(1, 10^{-2+\lfloor \text{epoch}/30 \rfloor})$
Restart epochs	$30, 60, 90, 120, 150$

Table 7: Details of the multi-layer fully-connected NN.

Parameter	Value
Type of layers	fully-connected layer
Number of layers	7
Number of output neurons each layer: $1, 2, 3, 4, 5, 6, 7$	$512, 256, 128, 64, 32, 16, 10$
Activation function for convolution/output layer	relu/softmax

Table 8: Details of the experimental settings of LeNet5 on MNIST in Section 5.3

Parameter	Value
Data set	MNIST
Model	LeNet5 (Table 11)
Loss function	Cross entropy
Regularization function	Group LASSO
Total epochs	300
ProxSGD	
Regularization weight	1×10^{-4}
Learning rate schedule	$\eta(\text{epoch}) = 10^{-2-\lfloor \text{epoch}/30 \rfloor}$
Momentum schedule	$c(\text{epoch}) = 0.9$
ProxSSI	
Regularization weight	5×10^{-7}
Learning rate schedule	$\eta(\text{epoch}) = 10^{-3-\lfloor \text{epoch}/30 \rfloor}$
RMDA	
Initial learning rate	10^{-1}
Initial momentum	10^{-2}
Learning rate schedule	multiplied by 0.1 when epoch reaches $30, 60, 90, 120$
Momentum schedule	multiplied by 10 when epoch reaches $30, 60$
Restart epochs	$30, 60, 90, 120$

Table 9: Details of the experimental settings of modified VGG16 on CIFAR10 in Section 5.3.

Parameter	Value
Data set	CIFAR10
Model	VGG16 (Table 12)
Loss function	Cross entropy
Regularization function	Group LASSO
Total epochs	500
ProxSGD	
Regularization weight	1×10^{-4}
Learning rate schedule	$\eta(\text{epoch}) = 10^{-2-\lfloor \text{epoch}/50 \rfloor}$
Momentum schedule	$c(\text{epoch}) = 0.9$
ProxSSI	
Regularization weight	2×10^{-7}
Learning rate schedule	$\eta(\text{epoch}) = 10^{-4-\lfloor \text{epoch}/50 \rfloor}$
RMDA	
Regularization weight	2×10^{-5}
Initial learning rate	10^{-1}
Initial momentum	10^{-2}
Learning rate schedule	multiplied by 0.1 when epoch reaches $50, 100, 150, 200, 250$
Momentum schedule	multiplied by 10 when epoch reaches $50, 100$
Restart epochs	$50, 100, 150, 200, 250$

Table 10: Details of the experimental settings of ResNet50 on CIFAR100 in Section 5.3. We use the ResNet50 model from the public script `https://github.com/weiaicunzai/pytorch-cifar100`.

Parameter	Value
Data set	CIFAR100
Model	ResNet50
Loss function	Cross entropy
Regularization function	Group LASSO
Total epochs	5,400
ProxSGD	
Regularization weight	5×10^{-5}
Learning rate schedule	$\eta(\text{epoch}) = 10^{-3-\lfloor \text{epoch}/90 \rfloor}$
Momentum	0.9
ProxSSI	
Regularization weight	4×10^{-8}
Learning rate schedule	$\eta(\text{epoch}) = 10^{-3-\lfloor \text{epoch}/90 \rfloor}$
Momentum	$(0.9, 0.999)$
RMDA	
Regularization weight	4×10^{-6}
Initial learning rate	10^{-1}
Initial momentum	10^{-2}
Learning rate schedule	multiplied by 0.1 when epoch reaches $90, 180, 270, 360, 450, 540$
Momentum schedule	multiplied by 10 when epoch reaches $90, 180$
Restart epochs	$90, 180, 270, 360, 450, 540$

Table 11: Details of the modified LeNet5.

Parameter	Value
Number of layers	5
Number of convolutional layers	13
Number of fully-connected layers	1
Size of convolutional kernels	3×3
Number of output filters $1, 2$	$6, 16$
Number of output neurons $3, 4, 5$	$120, 84, 10$
Kernel size, stride, padding of maxing pooling	2×2, none, invalid
Operations after convolutional layers	max pooling
Activation function for convolution/output layer	relu/softmax

Table 12: Details of the modified VGG16.

Parameter	Value
Number of layers	14
Number of convolutional layers	13
Number of fully-connected layers	1
Size of convolutional kernels	3×3
Number of output filters 1-2, 3-4, 5-7, 8-13	$64, 128, 256, 512$
Kernel size, stride, padding of maxing pooling	$2 \times 2, 2$, invalid
Operations after convolutional layers	max pooling, batchnorm
Activation function for convolution/output layer	relu/softmax

Table 13: Details of experimental settings of ResNet50 on CIFAR10 and CIFAR100 for unstructured sparsity. In this experiment, we adopt the version of ResNet50 in Sundar & Dwaraknath (2021).

Parameter	Value
Model	ResNet50
Loss function	Cross entropy
RMDA	
Data	CIFAR10
Total epochs	500
L1 weight	3^{-5}
Initial learning rate	1.0
Initial momentum	10^{-2}
Learning rate schedule	multiplied by 0.1 when epoch reaches $100, 200, 300$
Momentum schedule	multiplied by 10 when epoch reaches $100, 200$
Restart epochs	$100, 200, 300$
Data	CIFAR100
Total epochs	500
L1 weight	4^{-5}
Initial learning rate	1.0
Initial momentum	10^{-2}
Learning rate schedule	multiplied by 0.1 when epoch reaches $100, 200, 300$
Momentum schedule	multiplied by 10 when epoch reaches $100, 200$
Restart epochs	$100, 200, 300$
RigL	
Data	CIFAR10
Total epochs	1,000
Sparse initialization	erdos-renyi-kernel
Density	0.02
Prune rate	0.3
Decay schedule	cosine
Apply when	step end
Interval	100
End when	65,918
Learning rate	0.1
Momentum	0.9
Weight decay	1^{-4}
Label smoothing	0.1
Decay frequency	$20,000$
Warmup steps	$1,760$
Decay factor	0.2
Data	CIFAR100
Total epochs	1,000
Sparse initialization	erdos-renyi-kernel
Density	0.02
Prune rate	0.3
Decay schedule	cosine
Apply when	step end
Interval	100
End when	65,918
Learning rate	0.1
Momentum	0.9
Weight decay	1^{-4}
Label smoothing	0.1
Decay frequency	$20,000$
Warmup steps	$1,760$
Decay factor	0.2

Table 14: Results of training LeNet5 on MNIST with different regularizers. We report mean and standard deviation of three independent runs.

Regularizers	Validation accuracy	Group sparsity
Group LASSO	99.11 ± 0.06%	45.33 ± 0.99%
Group LASSO+L1	99.02 ± 0.01%	58.92 ± 1.30%
Gruop MCP	99.25 ± 0.08%	32.81 ± 0.96%
Gruop MCP + L1	99.21 ± 0.03%	32.91 ± 0.35%

Table 15: Details of the experimental settings of comparing different regularizers in Appendix E

Parameter	Value
Data set	MNIST
Model	LeNet5 (Table 11)
Loss function	Cross entropy
Algorithms	RMDA
Total epochs	300
Group LASSO only	
Group LASSO weight	1×10^{-5}
Learning rate schedule	$\eta(\text{epoch}) = 10^{-1-\lfloor \text{epoch}/30 \rfloor}$, if epoch $\leq 120; 10^{-5}$, otherwise
Momentum schedule	$c(\text{epoch}) = 10^{-2-\lfloor \text{epoch}/30 \rfloor}$, if epoch $\leq 60; 1$, otherwise
Restart epochs	$30, 60, 90, 120$
L1 + Group LASSO	
L1 weight	1×10^{-4}
Group LASSO weight	1×10^{-5}
Learning rate schedule	$\eta(\text{epoch}) = 10^{-1-\lfloor \text{epoch}/30 \rfloor}$, if epoch $\leq 120; 10^{-5}$, otherwise
Momentum schedule	$c(\text{epoch}) = 10^{-2-\lfloor \text{epoch}/30 \rfloor}$, if epoch $\leq 60; 1$, otherwise
Restart epochs	$30, 60, 90, 120$
Group MCP only	
Group MCP weight	1×10^{-5}
γ	64
Learning rate schedule	$\eta(\text{epoch}) = 10^{-1-\lfloor \text{epoch}/30 \rfloor}$, if epoch $\leq 120; 10^{-5}$, otherwise
Momentum schedule	$c(\text{epoch}) = 10^{-2-\lfloor \text{epoch}/30 \rfloor}$, if epoch $\leq 60; 1$, otherwise
Restart epochs	$30, 60, 90, 120$
L1 + Group MCP	
L1 weight	1×10^{-4}
Group MCP weight	1×10^{-5}
γ	64
Learning rate schedule	$\eta(\text{epoch}) = 10^{-1-\lfloor \text{epoch}/30 \rfloor}$, if epoch $\leq 120; 10^{-5}$, otherwise
Momentum schedule	$c(\text{epoch}) = 10^{-2-\lfloor \text{epoch}/30 \rfloor}$, if epoch $\leq 60; 1$, otherwise
Restart epochs	$30, 60, 90, 120$