# A NON-ASYMPTOTIC COMPARISON OF SVRG AND SGD: TRADEOFFS BETWEEN COMPUTE AND SPEED

**Anonymous authors**
Paper under double-blind review

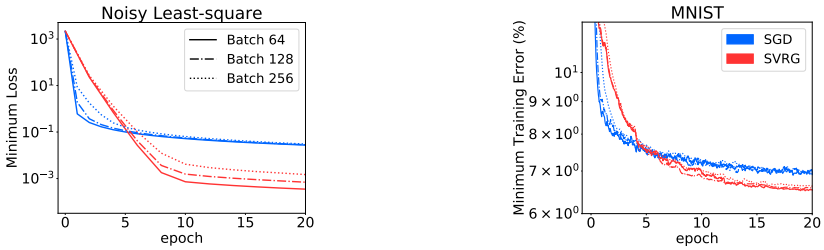
## ABSTRACT

Stochastic gradient descent (SGD), which trades off noisy gradient updates for computational efficiency, is the de-facto optimization algorithm to solve large-scale machine learning problems. SGD can make rapid learning progress by performing updates using subsampled training data, but the noisy updates also lead to slow asymptotic convergence. Several variance reduction algorithms, such as SVRG, introduce control variates to obtain a lower variance gradient estimate and faster convergence. Despite their appealing asymptotic guarantees, SVRG-like algorithms have not been widely adopted in deep learning. The traditional asymptotic analysis in stochastic optimization provides limited insight into training deep learning models under a fixed number of epochs. In this paper, we present a non-asymptotic analysis of SVRG under a noisy least squares regression problem. Our primary focus is to compare the exact loss of SVRG to that of SGD at each iteration $t$. We show that the learning dynamics of our regression model closely matches with that of neural networks on MNIST and CIFAR-10 for both the underparameterized and the overparameterized models. Our analysis and experimental results suggest there is a trade-off between the computational cost and the convergence speed in underparametrized neural networks. SVRG outperforms SGD after the first few epochs in this regime. However, SGD is shown to always outperform SVRG in the overparameterized regime.

## 1 INTRODUCTION

Many large-scale machine learning problems, especially in deep learning, are formulated as minimizing the sum of loss functions on millions of training examples (Krizhevsky et al., 2012; Devlin et al., 2018). Computing exact gradient over the entire training set is intractable for these problems. Instead of using full batch gradients, the variants of stochastic gradient descent (SGD) (Robbins & Monro, 1951; Sutskever et al., 2013; Duchi et al., 2011; Kingma & Ba, 2014) evaluate noisy gradient estimates from small mini-batches of randomly sampled training points at each iteration. The mini-batch size is often independent of the training set size, which allows SGD to immediately adapt the model parameters before going through the entire training set. Despite its simplicity, SGD works very well, even in the non-convex non-smooth deep learning problems (He et al., 2016; Vaswani et al., 2017). However, the optimization performance of the stochastic algorithm near local optima is significantly limited by the mini-batch sampling noise, controlled by the learning rate and the mini-batch size. The sampling variance and the slow convergence of SGD have been studied extensively in the past (Robbins & Monro, 1951; Polyak & Juditsky, 1992; Bottou, 2010). To ensure convergence, machine learning practitioners have to either increase the mini-batch size or decrease the learning rate toward the end of the training (Smith et al., 2017; Ge et al., 2019).

Recently, several clever variance reduction methods (Roux et al., 2012; Defazio et al., 2014; Wang et al., 2013; Johnson & Zhang, 2013) were proposed to alleviate the noisy gradient problem by using control-variates to achieve unbiased and lower-variance gradient estimators. In particular, the variants of Stochastic Variance Reduced Gradient (SVRG) (Johnson & Zhang, 2013), L-SVRG (Kovalev et al., 2019) and Free-SVRG (Sebbouh et al., 2019) construct control-variates from previous staled snapshot model parameters. These methods enjoy a superior asymptotic performance in convex optimization compared to the standard SGD. The control-variate techniques are shown to improve the convergence rate of SGD from a sub-linear to a linear convergence rate. These variance reduction methods can also be combined with momentum (Allen-Zhu, 2017) and preconditioning methods (Moritz et al., 2016) to obtain faster convergence. Despite their strong theoretical guarantees, SVRG-like algorithms have seen limited success in training deep learning models (Defazio &




(a) Least-squares regression (predicted). (b) Logistic regression.

Figure 1: (a) The minimum loss achieved over a set of hyperparameters in a noisy least squares regression problem (simulated dynamics). (b) The minimum loss achieved in real dataset MNIST (a logistic regression model). Our theoretical prediction (a) matched with the training dynamics for real datasets, demonstrating tradeoffs between computational cost and convergence speed. The curves in red are SVRG and curves in blue are SGD. Different markers refer to different per-iteration computational cost, i.e., the number of backpropagation used per iteration on average.

Bottou, 2018). Traditional results from stochastic optimization focus on the asymptotic analysis, but in practice, most of deep neural networks are only trained for hundreds of epochs due to the high computational cost. To address the gap between the asymptotic benefit of SVRG and the practical computational budget of training deep learning models, we provide a non-asymptotic study on the SVRG algorithms under a noisy least squares regression model. Although optimizing least squares regression is a basic problem, it has been shown to characterize the learning dynamics of many realistic deep learning models (Zhang et al., 2019; Lee et al., 2019). Recent works suggest that neural network learning behaves very differently in the underparameterized regime vs the overparameterized regime (Ma et al., 2018; Vaswani et al., 2019), characterized by whether the learnt model can achieve zero expected loss. We account for both training regimes in the analysis by assuming a linear target function and noisy labels. In the presence of label noise, the loss is lower bounded by the label variance. In the absence of the noise, the linear predictor can fit each training example perfectly. We summarize the main contributions as follows:

- We show the exact expected loss of SVRG and SGD along an optimization trajectory as a function of iterations and computational cost.
- Our non-asymptotic analysis provides an insightful comparison of SGD and SVRG by considering their computational cost and learning rate schedule. We discuss the trade-offs between the total computational cost, i.e. the per-iteration cost times the number of epochs, and convergence performance.
- We consider two different training regimes with and without label noise. Under noisy labels, the analysis suggests SGD only outperforms SVRG under a mild total computational cost. However, SGD always exhibits a faster convergence compared to SVRG when there is no label noise.
- Numerical experiments validate our theoretical predictions on both MNIST and CIFAR-10 using various neural network architectures. In particular, we found the comparison of the convergence speed of SGD to that of SVRG in underparameterized neural networks closely matches with our noisy least squares model prediction. Whereas, the effect of overparameterization is captured by the regression model without label noise.

### 1.1 Related Works

**Stochastic variance reduction methods** consider minimizing a finite-sum of a collection of functions using SGD. In case we use SGD to minimize these objective functions, the stochasticity comes from the randomness in sampling a function in each optimization step. Due to the induced noise, SGD can only converge using decaying step sizes with sub-linear convergence rate. Methods such as SAG (Roux et al., 2012), SVRG (Johnson & Zhang, 2013), and SAGA (Defazio et al., 2014), are able to recover linear convergence rate of full-batch gradient descent with the asymptotic cost comparable to SGD. SAG and SAGA achieve this improvement at the substantial cost of storing the most recent gradient of each individual function. In contrast, SVRG spends extra computation at snapshot intervals by evaluating the full-batch gradient. Theoretical results such as Gazagnadou et al. (2019) show that under certain smoothness conditions, we can use larger step sizes with stochastic variance reduction methods than is allowed for SGD and hence achieve even faster convergence. In situations where we know the smoothness constant of functions, there are results on the optimal mini-batch size and the optimal step size given the inner loop size (Sebbouh et al., 2019). Applying variance reduction methods in deep learning has been studied recently (Defazio & Bottou, 2018). The authors

conjectured the ineffectiveness is caused by various elements commonly used in deep learning such as data augmentation, batch normalization and dropout. Such elements can potentially decrease the smoothness and make the stored gradients become stale quickly. The proposed solution is to either remove these elements or update the gradients more frequently than is practical.

**Dynamics of SGD and quadratic models** Our main analysis tool is very closely related to recent work studying the dynamics of gradient-based stochastic methods. Wu et al. (2018) derived the dynamics of stochastic gradient descent with momentum on a noisy quadratic model (Schaul et al., 2013), showing the problem of short horizon bias. In (Zhang et al., 2019), the authors showed the same noisy quadratic model captures many of the essential characteristic of realistic neural networks training. Their noisy quadratic model successfully predicts the effectiveness of momentum, preconditioning and learning rate choices in training ResNets and Transformers. However, these previous quadratic models assume a constant variance in the gradient that is independent of the current parameters and the loss function. It makes them inadequate for analyzing the stochastic variance reduction methods, as SVRG can trivially achieve zero variance under the constant gradient noise. Instead, we adopted a noisy least-squares regression formulation by considering both the mini-batch sampling noise and the label noise. There are also recent works that derived the risk of SGD, for least-squares regression models using the bias-variance decomposition of the risk (Belkin et al., 2018; Hastie et al., 2019). We use a similar decomposition in our analysis. In contrast to the asymptotic analysis in these works, we compare SGD to SVRG along the optimization trajectory for any finite-time horizon under limited computation cost, not just the convergence points of those algorithms.

**Underparameterization vs overparameterization.** Many of the state-of-the-art deep learning models are overparameterized deep neural networks with more parameters than the number of training examples. Even though these models are able to overfit to the data, when trained using SGD, they generalize well (Zhang et al., 2017). As suggested in recent work, underparameterized and overparameterized regimes have different behaviours (Ma et al., 2018; Vaswani et al., 2019; Schmidt & Roux, 2013). Given the infinite width and a proper weight initialization, the learning dynamics of a neural network can be well-approximated by a linear model via the neural tangent kernel (NTK) (Jacot et al., 2018; Chizat & Bach, 2018). In NTK regime, neural networks are known to achieve global convergence by memorizing every training example. On the other hand, previous convergence results for SVRG have been obtained in stochastic convex optimization problems that are similar to that of an underparameterized model (Roux et al., 2012; Johnson & Zhang, 2013). Our proposed noisy least-squares regression analysis captures both the underparameterization and overparameterization behavior by considering the presence or the absence of the label noise.

## 2 PRELIMINARY

### 2.1 NOTATIONS

We will primarily focus on comparing the minibatch version of two methods, SGD and SVRG (Johnson & Zhang, 2013). Denote $L_i$ as the loss on $i^{th}$ data point. The SGD update is written as,

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \alpha^{(t)} \hat{\mathbf{g}}^{(t)}, \tag{1}$$

where $\hat{\mathbf{g}}^{(t)} = \frac{1}{b}\sum_i^b \nabla_{\boldsymbol{\theta}^{(t)}} L_i$ is the minibatch gradient, $t$ is the training iteration, and $\alpha^{(t)}$ is the learning rate. The SVRG algorithm is an inner-outer loop algorithm proposed to reduce the variance of the gradient caused by the minibatch sampling. In the outer loop, for every $T$ steps, we evaluate a large batch gradient $\bar{\mathbf{g}} = \frac{1}{N}\sum_i^N \nabla_{\boldsymbol{\theta}^{(mT)}} L_i$, where $N \gg b$, and $m$ is the outer loop index, and we store the parameters $\boldsymbol{\theta}^{(mT)}$. In the inner loop, the update rule of the parameters is given by,

$$\boldsymbol{\theta}^{(mT+t+1)} = \boldsymbol{\theta}^{(mT+t)} - \alpha^{(t)} \left( \hat{\mathbf{g}}^{(mT+t)} - \tilde{\mathbf{g}}^{(mT+t)} + \bar{\mathbf{g}} \right) \tag{2}$$

where $\hat{\mathbf{g}}^{(mT+t)} = \frac{1}{b}\sum_i^b \nabla_{\boldsymbol{\theta}^{(mT+t)}} L_i$ is the current gradient of the mini-batch and $\tilde{\mathbf{g}}^{(mT+t)} = \frac{1}{b}\sum_i^b \nabla_{\boldsymbol{\theta}^{(mT)}} L_i$ is the old gradient. Note that in our analysis, the reference point is chosen to be the last iterate of previous outer loop $\boldsymbol{\theta}^{(mT)}$, recommended as a practical implementation of the algorithm by the original SVRG paper Johnson & Zhang (2013).

### 2.2 THE NOISY LEAST SQUARES REGRESSION MODEL

We now define the noisy least squares regression model (Schaul et al., 2013; Wu et al., 2018). In this setting, the input data is $d$-dimensional, and the output label is generated by a linear teacher model with additive noise,

$$(\boldsymbol{x}_i, \epsilon_i) \sim P_x \times P_\epsilon; \quad y_i = \boldsymbol{x}_i^\top \boldsymbol{\theta}^* + \epsilon_i,$$

where $\mathbb{E}[\boldsymbol{x}_i] = \boldsymbol{\mu} \in \mathbb{R}^d$ and $\mathrm{Cov}(\boldsymbol{x}_i) = \Sigma$, $\mathbb{E}[\epsilon_i] = 0$, $\mathrm{Var}(\epsilon_i) = \sigma_y^2$. We assume WLOG $\boldsymbol{\theta}^* = \mathbf{0}$. We also assume the data covariance matrix $\Sigma$ is diagonal. This is an assumption adopted in many previous analysis and it is also a practical assumption as we often apply whitening to pre-process the training data. We would like to train a student model $\boldsymbol{\theta}$ that minimizes the squared loss over the data distribution:

$$\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) := \mathbb{E}\left[\frac{1}{2}(\boldsymbol{x}_i^\top \boldsymbol{\theta} - y_i)^2\right]. \tag{3}$$

At each iteration, the optimizer can query an arbitrary number of data points $\{\boldsymbol{x}_i, y_i\}_i$ sampled from data distribution. The SGD method uses $b$ data points to form a minibatch gradient:

$$\hat{\mathbf{g}}^{(t)} = \frac{1}{b}\sum_i^b (\boldsymbol{x}_i \boldsymbol{x}_i^\top \boldsymbol{\theta}^{(t)} - \boldsymbol{x}_i \epsilon_i) = X_b X_b^\top \boldsymbol{\theta}^{(t)} - \frac{1}{\sqrt{b}} X_b \boldsymbol{\epsilon}_b, \tag{4}$$

where $X_b = \frac{1}{\sqrt{b}}[\boldsymbol{x}_1; \boldsymbol{x}_2; \cdots; \boldsymbol{x}_b] \in \mathbb{R}^{d\times b}$, and the noise vector $\boldsymbol{\epsilon}_b = [\epsilon_1; \epsilon_2; \cdots; \epsilon_b]^\top \in \mathbb{R}^b$. SVRG on the other hand, queries for $N$ data points every $T$ steps to form a large batch gradient $\bar{\mathbf{g}} = X_N X_N^\top \boldsymbol{\theta}^{(mT)} - \frac{1}{\sqrt{N}} X_N \boldsymbol{\epsilon}_N$, where $X_N$ and $\boldsymbol{\epsilon}_N$ are defined similarly. At each inner loop step, it further queries for another $b$ data points, to form the update in Eq. 2.

Lastly, note that the expected loss can be written as a function of the second moment of the iterate,

$$L(\boldsymbol{\theta}^{(t)}) = \frac{1}{2}\mathbb{E}\left[\left(\boldsymbol{x}_i^\top \boldsymbol{\theta}^{(t)} - \epsilon_i\right)^2\right] = \frac{1}{2}\left(tr(\Sigma \mathbb{E}[\boldsymbol{\theta}^{(t)} \boldsymbol{\theta}^{(t)\top}]) + \sigma_y^2\right).$$

Hence for the following analysis we mainly focus on deriving the dynamics of the second moment $\mathbb{E}[\boldsymbol{\theta}^{(t)} \boldsymbol{\theta}^{(t)\top}]$, denoted as $A(\boldsymbol{\theta}^{(t)})$. When $\Sigma$ is diagonal, the loss can further be reduced to $\frac{1}{2}\mathrm{diag}(\Sigma)^\top \mathrm{diag}(\mathbb{E}[\boldsymbol{\theta}^{(t)} \boldsymbol{\theta}^{(t)\top}]) + \frac{1}{2}\sigma_y^2$. We denote $\mathrm{diag}(\mathbb{E}[\boldsymbol{\theta}^{(t)} \boldsymbol{\theta}^{(t)\top}])$ by $\mathbf{m}(\boldsymbol{\theta}^{(t)})$.

### 2.3 THE DYNAMICS OF SGD

**Definition 1** (Formula for dynamics)**.** *We define the following functions and identities,*

$$\mathrm{M}(\boldsymbol{\theta}) = \mathbb{E}[\boldsymbol{\theta}\boldsymbol{\theta}^\top], \quad \mathbf{m}(\boldsymbol{\theta}) = diag(\mathbb{E}[\boldsymbol{\theta}\boldsymbol{\theta}^\top]), \quad \mathrm{C}(\boldsymbol{\theta}) = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{\theta}\boldsymbol{\theta}^\top \boldsymbol{x}\boldsymbol{x}^\top] - \Sigma\mathbb{E}[\boldsymbol{\theta}\boldsymbol{\theta}^\top]\Sigma,$$
$$V = \alpha^2 \sigma_y^2 diag(\Sigma), \quad R = (\mathrm{I} - \alpha\Sigma)^2 + 2\alpha^2 B^{-1}\Sigma^2,$$
$$Q = \frac{4\alpha^2}{b}\Sigma^2, \quad P = \mathrm{I} - \alpha\Sigma, \quad F = \frac{2\alpha^2(N+b)}{bN}\Sigma^2.$$

The SGD update (Eq. 1) with the mini-batch gradient of of the noisy least squares model (Eq. 4) is,

$$\boldsymbol{\theta}^{(t+1)} = (\mathrm{I} - \alpha X_b X_b^\top)\boldsymbol{\theta}^{(t)} + \frac{\alpha}{\sqrt{b}} X_b \boldsymbol{\epsilon}_b.$$

We substitute the update rule to derive the following dynamics for the second moment of the iterate:

$$\mathrm{M}(\boldsymbol{\theta}^{(t+1)}) = \underbrace{(\mathrm{I} - \alpha\Sigma)\mathrm{M}(\boldsymbol{\theta}^{(t)})(\mathrm{I} - \alpha\Sigma)}_{\text{①: gradient descent shrinkage}} + \underbrace{\frac{\alpha^2}{b}\mathrm{C}(\boldsymbol{\theta}^{(t)})}_{\text{②: input noise}} + \underbrace{\frac{\alpha^2\sigma_y^2}{b}\Sigma}_{\text{③: label noise}} \tag{5}$$

This dynamics equation can be understood intuitively as follows. The term ① leads to an exponential shrinkage of the loss due to the gradient descent update. Since we are using a noisy gradient, the second term ② represents the variance of stochastic gradient caused by the random input $X_b$. The term ③ comes from the label noise. We show in the next theorem that when the second moment of the iterate approaches zero, ② will also approach zero. However due to the presence of the label noise, the expected loss is lower bounded by ③.

When $\Sigma$ is diagonal, we further analyze and decompose $\mathrm{C}(\boldsymbol{\theta})$ as a function of $\mathbf{m}(\boldsymbol{\theta})$ so as to derive the following dynamics and decay rate for SGD.

**Theorem 2** (SGD Dynamics and Decay Rate)**.** *Given the noisy least squares regression objective function (Eq. 3), under the assumption that $x \sim \mathcal{N}(\mathbf{0}, \Sigma)$ with $\Sigma$ diagonal and $\boldsymbol{\theta}^* = 0$, we can express $\mathrm{C}(\boldsymbol{\theta})$ as a function of $\mathbf{m}(\boldsymbol{\theta})$:*

$$diag(\mathrm{C}(\boldsymbol{\theta})) = 3\Sigma^2 \cdot \mathbf{m}(\boldsymbol{\theta}) \tag{6}$$

*Then we derive the dynamics of expected second moment of $\boldsymbol{\theta}$ following SGD,*

$$\mathbf{m}(\boldsymbol{\theta}^{(t)}) = R^t\Big(\mathbf{m}(\boldsymbol{\theta}^{(0)}) - \frac{V}{b(\mathrm{I}-R)}\Big) + \frac{V}{b(\mathrm{I}-R)}.$$

Theorem 2 shows that the expected second moment of each parameter evolve independent of other parameters. And under the update rule of SGD, $R$ is the decay rate of the second moment of parameters between two iterations.

## 3 A DILEMMA FOR SVRG

By querying a large batch of datapoints $X_N$ every $T$ steps, and a small minibatch $X_b$ at every step, the SVRG method forms the following update rule:

$$\boldsymbol{\theta}^{(mT+t+1)} = \left(\mathrm{I} - \alpha X_b X_b^\top\right)\boldsymbol{\theta}^{(mT+t)} + \alpha\left(X_b X_b^\top - X_N X_N^\top\right)\boldsymbol{\theta}^{(mT)} + \frac{\alpha}{\sqrt{N}} X_N \boldsymbol{\epsilon}_N \tag{7}$$

To derive the dynamics of the second moment of the parameters following the SVRG update, we look at the dynamics of one round of inner loop updates, i.e., from $\boldsymbol{\theta}^{(mT)}$ to $\boldsymbol{\theta}^{(mT+t)}$:

**Lemma 3.** *The dynamics of the second moment of the iterate following SVRG update rule is given by,*

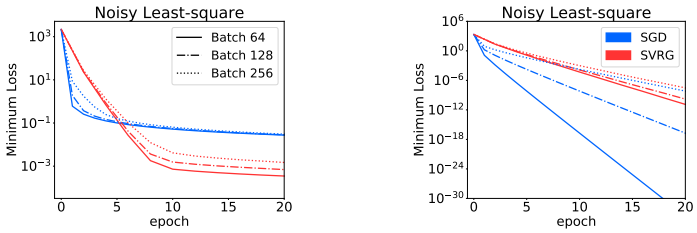
$$\begin{aligned}\mathrm{M}(\boldsymbol{\theta}^{(mT+t+1)}) = &\underbrace{(I-\alpha\Sigma)\mathrm{M}(\boldsymbol{\theta}^{(mT+t)})(\mathrm{I}-\alpha\Sigma)}_{\text{① gradient descent shrinkage}} + \underbrace{\frac{\alpha^2}{b}\mathrm{C}(\boldsymbol{\theta}^{(mT+t)})}_{\text{② input noise}} + \underbrace{\frac{\alpha^2\sigma_y^2}{N}\Sigma}_{\text{③ label noise}} \\ &+ \underbrace{\alpha^2\frac{N+b}{Nb}\mathrm{C}(\boldsymbol{\theta}^{(mT)})}_{\text{④variance due to } \tilde{\mathbf{g}}^{(mT+t)}} \underbrace{-\frac{\alpha^2}{b}\left(\mathrm{C}(\boldsymbol{\theta}^{(mT)})\Big(\mathrm{I}-\alpha\Sigma\Big)^t + \Big(\mathrm{I}-\alpha\Sigma\Big)^t\mathrm{C}(\boldsymbol{\theta}^{(mT)})\right)}_{\text{⑤ Variance reduction from control variate}}.\end{aligned} \tag{8}$$

The dynamics equation above is very illuminating as it explicitly manifests the weakness of SVRG. First notice that terms ①, ②, ③ reappear, contributed by the SGD update. The additional terms, ④ and ⑤, are due to the control variate. Observe that the variance reduction term ⑤ decays exponentially throughout the inner loop, with decay rate $\mathrm{I}-\alpha\Sigma$. We immediately notice that this is the same term that governs the decay rate of the term ①, and hence resulting in a conflict between the two. Specifically, if we want to reduce the term ① as fast as possible, we would prefer a small decay rate and a large learning rate, i.e. $\alpha \to \frac{1}{\lambda_{\max}(\Sigma)}$. But this will also make the boosts provided by the control variate diminish rapidly, leading to a poor variance reduction. The term ④ makes things even worse as it will maintain as a constant throughout the inner loop, contributing to an extra variance on top of the variance from standard SGD. On the other hand, if one chooses a small learning rate for the variance reduction to take effect, this inevitably will make the decay rate for term ① smaller, resulting in a slower convergence. Nevertheless, a good news for SVRG is that the label noise (term ③) is scaled by $\frac{b}{N}$, which lets SVRG converge to a lower loss value than SGD – a strict advantage of SVRG compared to SGD.

To summarize, the variance reduction from SVRG comes at a price of slower gradient descent shrinkage. In contrast, SVRG is able to converge to a lower loss value. This motivates the question, which algorithm to use given a certain computational cost? We hence performed a thorough investigation through numerical simulation as well as experiments on real datasets in Sec. 4.

Similarly done for SGD, we decompose $\mathrm{C}(\boldsymbol{\theta})$ as a function of $\mathbf{m}(\boldsymbol{\theta})$ and derive the following decay rate for SVRG.

**Theorem 4** (SVRG Dynamics and Decay ate)**.** *Given the noisy least squares regression objective function (Eq. 3), under the assumption that $x \sim \mathcal{N}(\mathbf{0}, \Sigma)$ with $\Sigma$ diagonal and $\boldsymbol{\theta}^* = 0$, the dynamics*




(a) With Label Noise (b) Without Label Noise

Figure 2: The minimum loss achieved by following SGD (blue) and SVRG (red) over a set of hyperparameters in a noisy least-square dynamics simulation for cases with and without label noise. The plot suggests that in the presence of label noise, there is a tradeoff between computational cost and convergence speed. In the absence of label noise, SGD strictly dominates SVRG in convergence speed for all computational cost.

*for SVRG in* $\mathbf{m}(\boldsymbol{\theta})$ *is given by*[1]*:*

$$\mathbf{m}(\boldsymbol{\theta}^{((m+1)T)}) = \lambda(\alpha, b, T, \Sigma, N)\mathbf{m}(\boldsymbol{\theta}^{(mT)}) + \frac{\mathrm{I} - R^T}{\mathrm{I} - R}\frac{V}{N}, \tag{9}$$

$$\lambda(\alpha, B, T, \Sigma, N) = R^T\left(\mathrm{I} + \frac{Q}{P - R} + \frac{F}{R - \mathrm{I}}\right) - \frac{P^T Q}{P - R} - \frac{F}{R - \mathrm{I}}. \tag{10}$$

### 3.1 The dynamics without label Noise

In the absence of the label noise (i.e., $\sigma_y = 0$), we observe that both SGD and SVRG enjoy linear convergence as a corollary of Theorem 2 and Theorem 4:

**Corollary 5.** *Without the label noise, the dynamics of the second moment following SGD is given by,*

$$\mathbf{m}(\boldsymbol{\theta}^{(t)}) = R^t \mathbf{m}(\boldsymbol{\theta}^{(0)}),$$

*and the dynamics of SVRG is given by,*

$$\mathbf{m}(\boldsymbol{\theta}^{((m+1)T)}) = \lambda(\alpha, b, T, \Sigma, N)\mathbf{m}(\boldsymbol{\theta}^{(mT)}),$$

*where* $\lambda$ *is defined in Eq. 10.*

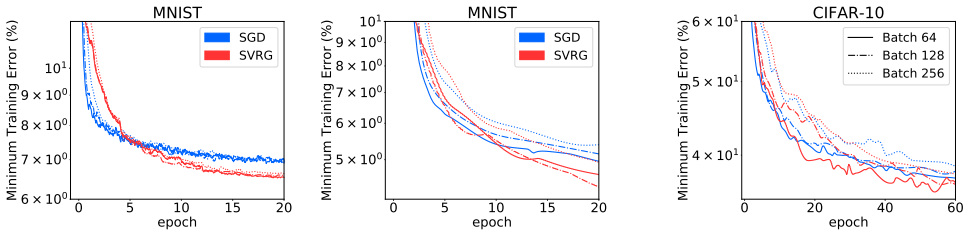
Note that similar results have been shown in the past (Ma et al., 2018; Vaswani et al., 2019; Schmidt & Roux, 2013), where a general condition known as "interpolation regime" is used to show linear convergence of SGD. Specifically they assume that $\nabla L_i(\boldsymbol{\theta}^*) = 0$ for all $i$, and our setting without label noise clearly also belongs to this regime. This setting also has practical implications, as one can treat training overparameterized neural networks as in interpolation regime. This motivates the investigation of the convergence rate of SGD and SVRG without label noise, and was also extensively studied in the experiments detailed as follows.

## 4 Experiments

In Sec. 3 we discussed a critical dilemma for SVRG that is facing a choice between effective variance reduction and faster gradient descent shrinkage. At the same time, it enjoys a strict advantage over SGD as it converges to a lower loss. We define the total computational cost as the total number of back-propagations performed. Similarly, per-iteration computational cost refers to the number of back-propagations performed per iteration. In this section, we study the question, which algorithm converges faster given certain total computational cost? We study this question for both the underparameterized and the overparameterized regimes.

Our investigation consists of two parts. First, numerical simulations of the theoretical convergence rates (Sec. 4.1). Second, experiments on real datasets (Sec. 4.2). In both parts, we first fix the per-iteration computational cost. For SGD, the per-iteration computational budge is equal to the minibatch size. We picked three batch size $\{64, 128, 256\}$. Denote the batchsize of SGD as $b$, the equivalent batch size for SVRG is $b' = \frac{1}{2}(1 - \frac{N}{Tb})b$. We then perform training with an extensive set of hyperparameters for each method with each per-iteration computational cost. For SGD, the hyperparameter under consideration is the learning rate $\alpha$. For SVRG, besides the learning rate,

[1]The following fraction form of matrices is well defined as all the matrices in the numerator and denominator are diagonal and hence commutative.




(a) Logistic Regression. (b) underparameterized MLP (c) underparameterized CNN

Figure 3: The minimum loss achieved by following SGD (blue) and SVRG (red) over a set of hyperparameters for training on MNIST and CIFAR-10 with underparameterized models. All the results in these plot suggested there is a tradeoff between computational cost and convergence speed when comparing SGD and SVRG.

we also ran over a set of snapshot intervals $T$. After running over all sets of hyperparameters, we gather all training curves of all hyperparameters. We then summarize the performance for each algorithm by plotting the lower bound of all training curves, i.e. each point $(l, t)$ on the curve showed the minimum loss $l$ at time step $t$ over all hyperparameters. We compared the two methods under different computational cost.

Remarkably, we found in many cases phenomenon predicted by our theory matches with observations in practice. Our experiments suggested there is a trade-off between the computational cost and the convergence speed for underparameterized neural networks. SVRG outperformed SGD after a few epochs in this regime. Interestingly, in the case of overparameterized model, a setting that matches modern day neural networks training, SGD strictly dominated SVRG by showing a faster convergence throughout the entire training.

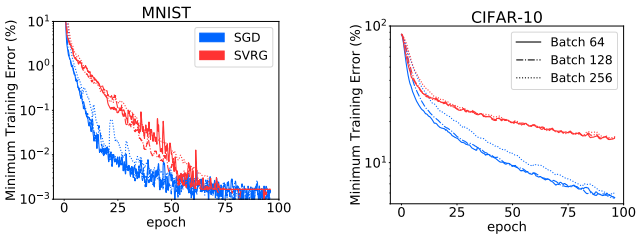
### 4.1 Simulations on Noisy least squares Regression Model

We first performed numerical simulations of the dynamics derived in Theorem 2 for SGD and Theorem 4 for SVRG. We picked a data distribution, with data dimension $d = 100$, and the spectrum of $\Sigma$ is given by an exponential decay schedule from 1 to 0.01. For both methods, we picked 50 learning rate from 1.5 to 0.01 using a exponential decay schedule. For SVRG, we further picked a set of snapshot intervals for each learning rate: $\{64, 128, 256\}$. We performed simulations in both underparameterized and overparameterized setting (namely with and without label noise), and plotted the lower bound curves over all hyperparameters at Figure 2. The $x$-axis represents the normalized total computational cost, denoting $tbN^{-1}$, which is equivalent to the notion of an epoch in finite dataset setting.

We have the following observations from our simulations. In the case with label noise, the plot demonstrated an explicit trade-off between computational cost and convergence speed. We observed a crossing point of between SGD and SVRG appear, indicating SGD achieved a faster convergence speed in the first phase of the training, but converged to a higher loss, for all per-iteration compute cost. Hence it shows that one can trade more compute cost for convergence speed by choosing SGD than SVRG, and vice versa. Interestingly, we found that the the per-iteration computational cost does not seem to affect the time crossing point takes place. For all these three costs, the crossing points in the plot are at around the same time: 5.5 epochs. In the case of no label noise, we observed both methods achieved linear convergence, while SGD achieved a much faster rate than SVRG, showing absolute dominance in this regime.

### 4.2 Benchmark datasets

In this section, we performed a similar investigation as in the last section, on two standard machine learning benchmark datasets: MNIST (LeCun et al., 1998) and CIFAR-10 (Krizhevsky, 2009). We present the results from underparameterized setting first, followed by the overparameterized setting. We performed experiments with three batchsizes for SGD: $\{64, 128, 256\}$, and an equivalent batchsize for SVRG. For each batch size, we pick 8 learning rates varying from 0.3 to 0.001 following an exponential schedule. Additionally, we chose $\{64, 128, 256\}$ searching over the best snapshot interval given the data. Hence for each per-iteration computational cost $\{64, 128, 256\}$, there are 24 groups of experiments for SVRG and 8 groups of experiments for SGD.




(a) Over-paremetrized MLP (b) overparameterized CNN

Figure 4: The minimum loss achieved by following SGD (blue) and SVRG (red) over a set of hyperparameters for training on MNIST and CIFAR-10 with overparameterized models. In this setting we observed strict dominance of SGD over SVRG in convergence speed for all computational cost, matching our previous theoretical prediction.

#### 4.2.1 UNDERPARAMETERIZED SETTING

For MNIST, we trained two underparameterized model: 1. logistic regression $784 - 10$ 2. a underparameterized two layer MLP $784 - 10 - 10$ where the hidden layer has 10 neurons. For CIFAR-10, we chose a underparameterized convolutional neural network model, which has only two 8-channel convolutional layers, and one 16-channel convolutional layer with one additional fully-connected layer. Filter size is 5. The lowest loss achieved over all hyperparameters for these models for each per-iteration computational cost are shown in Figure 3.

From these experiments, we observe that on MNIST, the results with underparameterized model were consistent with the dynamics simulation of noisy least squares regression model with label noise. First of all, SGD converged faster in the early phase, resulting in a crossing point between SGD and SVRG. It showed a trade-offs between computational cost and convergence speed: before the crossing point, SGD converged faster than SVRG; after crossing point, SVRG attained a lower loss. In addition, in Fig 3a, all the crossing points of three costs matched at the same epoch (around 5), which was also consistent with the our findings with noisy least squares regression model. On CIFAR-10, SGD achieved slightly faster convergence in the early phase, but was surpassed by SVRG around $17 - 25$ epochs, again showing a trade-off between compute and speed.

#### 4.2.2 THE OVERPARAMETERIZED SETTING

Lastly, we compared SGD and SVRG on MNIST and CIFAR-10 using overparameterized models. For MNIST, we used a MLP with two hidden layers, each layer having 1024 neurons. For CIFAR-10, we chose a large convolutional network, which has one 64-channel convolutional layer, one 128-channel convolutional layer followed by one 3200 to 1000 fully connected layer and one 1000 to 10 fully connected layer.

The lowest loss achieved over all hyperparameters for these models for each per-iteration computational cost are shown in Figure 4. For training on MNIST, both SGD and SVRG attained close to zero training loss. The results were again consistent with our dynamics analysis on the noisy linear regression model without label noise. SGD has a strict advantage over SVRG, and achieved a much faster convergence rate than SVRG throughout the entire training. As for CIFAR-10, we stopped the training before either of the two got close to zero training loss due to lack of computing time. But we clearly see a trend of approaching to zero loss. Similarly, we also had the same observations as before, where SGD outperforms SVRG, confirms the limitation of variance reduction in the overparameterized regime.

## 5 DISCUSSION

In this paper, we studied the convergence properties of SGD and SVRG in the underparameterized and overparameterized settings. We provided a non-asymptotic analysis of both algorithms. We then investigated the question about which algorithm to use under certain total computational cost. We performed numerical simulations of dynamics equations for both methods, as well as extensive experiments on the standard machine learning datasets, MNIST and CIFAR-10. Remarkably, we found in many cases phenomenon predicted by our theory matched with observations in practice. Our experiments suggested there is a trade-off between the computational cost and the convergence speed for underparameterized neural networks. SVRG outperformed SGD after the first few epochs in this regime. In the case of overparameterized model, a setting that matches with modern day neural networks training, SGD strictly dominated SVRG by showing a faster convergence for all computational cost.

## REFERENCES

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *The Journal of Machine Learning Research*, 18(1):8194–8244, 2017.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine learning and the bias-variance trade-off. *CoRR*, abs/1812.11118, 2018.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

Lénaïc Chizat and Francis Bach. A note on lazy training in supervised differentiable programming. *CoRR*, abs/1812.07956, 2018.

Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. *CoRR*, abs/1812.04529, 2018.

Aaron Defazio, Francis R. Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Neural Information Processing Systems (NeurIPS)*, pp. 1646–1654, 2014.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Nidham Gazagnadou, Robert M. Gower, and Joseph Salmon. Optimal mini-batch and step sizes for SAGA. In *International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2142–2150. PMLR, 2019.

Rong Ge, Sham M Kakade, Rahul Kidambi, and Praneeth Netrapalli. The step decay schedule: A near optimal, geometrically decaying learning rate procedure. *arXiv preprint arXiv:1904.12838*, 2019.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *CoRR*, abs/1903.08560, 2019.

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

Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, pp. 8580–8589, 2018.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Neural Information Processing Systems (NeurIPS)*, pp. 315–323, 2013.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Dmitry Kovalev, Samuel Horváth, and Peter Richtárik. Don't jump through hoops and remove those loops: Svrg and katyusha are better without the outer loop. *arXiv preprint arXiv:1901.08689*, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. ISSN 0018-9219. doi: 10.1109/5.726791.

Jaehoon Lee, Lechao Xiao, Samuel S Schoenholz, Yasaman Bahri, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019.

Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3331–3340. PMLR, 2018.

Philipp Moritz, Robert Nishihara, and Michael Jordan. A linearly-convergent stochastic l-bfgs algorithm. In *Artificial Intelligence and Statistics*, pp. 249–258, 2016.

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Nicolas Le Roux, Mark W. Schmidt, and Francis R. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Neural Information Processing Systems (NeurIPS)*, pp. 2672–2680, 2012.

Tom Schaul, Sixin Zhang, and Yann LeCun. No more pesky learning rates. In *ICML (3)*, volume 28 of *JMLR Workshop and Conference Proceedings*, pp. 343–351. JMLR.org, 2013.

Mark Schmidt and Nicolas Le Roux. Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint arXiv:1308.6370*, 2013.

Othmane Sebbouh, Nidham Gazagnadou, Samy Jelassi, Francis Bach, and Robert M Gower. Towards closing the gap between the theory and practice of svrg. *arXiv preprint arXiv:1908.02725*, 2019.

Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pp. 1139–1147, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In *AISTATS*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1195–1204. PMLR, 2019.

Chong Wang, Xi Chen, Alexander J Smola, and Eric P Xing. Variance reduction for stochastic gradient optimization. In *Advances in Neural Information Processing Systems*, pp. 181–189, 2013.

Yuhuai Wu, Mengye Ren, Renjie Liao, and Roger B. Grosse. Understanding short-horizon bias in stochastic meta-optimization. In *ICLR (Poster)*. OpenReview.net, 2018.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*. OpenReview.net, 2017.

Guodong Zhang, Lala Li, Zachary Nado, James Martens, Sushant Sachdeva, George E Dahl, Christopher J Shallue, and Roger Grosse. Which algorithmic choices matter at which batch sizes? insights from a noisy quadratic model. *arXiv preprint arXiv:1907.04164*, 2019.

# A APPENDIX

# B LEMMA ABOUT GRADIENT COVARIANCE

**Lemma 6** (Gradient Covariance)**.** *Given the noisy linear regression objective function (Eq. 3), under the assumption that $x \sim \mathcal{N}(\mathbf{0}, \Sigma)$ with $\Sigma$ diagonal and $\boldsymbol{\theta}^* = 0$, we have:*

$$diag(\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top \boldsymbol{x}\boldsymbol{x}^\top]) = 3\Sigma^2 \mathbb{E}[{\boldsymbol{\theta}^{(t)}}^{\circ 2}] \tag{11}$$

$$\mathbb{E}[X_b X_b^\top \boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top X_b X_b^\top] - \Sigma\mathbb{E}[\boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top]\Sigma = \frac{1}{B}\Big(\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top \boldsymbol{x}\boldsymbol{x}^\top] - \Sigma\mathbb{E}[\boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top]\Sigma\Big) \tag{12}$$

*Proof.* Under the assumption $\mu_x = 0$, $\theta^* = 0$ and $\Sigma$ diagonal, we first consider the main diagonal entry of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top \boldsymbol{x}\boldsymbol{x}^\top]$:

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{\theta}^{(t)}}[\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top \boldsymbol{x}\boldsymbol{x}^\top]_{ii} = \mathbb{E}_{\boldsymbol{x}}\Big[\sum_{\substack{1\leqslant l\leqslant d\\ 1\leqslant p\leqslant d}} [\boldsymbol{x}]_{i1}[\boldsymbol{x}^\top]_{1l}\mathbb{E}[\boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top]_{lp}[\boldsymbol{x}]_{p1}[\boldsymbol{x}]_{1j}\Big] \tag{13}$$

For equation 13, since $\mu_x = 0$ and $\Sigma$ is diagonal, $[\boldsymbol{x}]_{i1}[\boldsymbol{x}^\top]_{1l}\mathbb{E}[\boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top]_{lp}[\boldsymbol{x}]_{p1}[\boldsymbol{x}]_{1i}$ will not be equal to zero if and only if $l = i, p = i$. Thus,

$$\begin{aligned}\mathbb{E}_{\boldsymbol{x},\boldsymbol{\theta}^{(t)}}[\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top \boldsymbol{x}\boldsymbol{x}^\top]_{ii} =& \mathbb{E}[\boldsymbol{x}^{\circ 4}]_i \mathbb{E}[\boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top]_{ii} &(14)\\ =& 3[\Sigma^2]_{ii}\mathbb{E}[\boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top]_{ii} &(15)\end{aligned}$$

Thus,

$$diag(\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top \boldsymbol{x}\boldsymbol{x}^\top]) = 3\Sigma^2 \mathbb{E}[{\boldsymbol{\theta}^{(t)}}^{\circ 2}] \tag{16}$$

which is the first conclusion of Theorem 2.

Then, for off-diagonal entry of $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top \boldsymbol{x}\boldsymbol{x}^\top]$ ($i \neq j$ in equation 13), $[\boldsymbol{x}]_{i1}[\boldsymbol{x}^\top]_{1l}\mathbb{E}[\boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top]_{lp}[\boldsymbol{x}]_{p1}[\boldsymbol{x}]_{1j}$ will not be equal to zero if and only if $l = i, p = j$ or $p = i, l = j$. Thus,

$$\begin{aligned}\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top \boldsymbol{x}\boldsymbol{x}^\top]_{ij} =& \mathbb{E}[\boldsymbol{x}^{\circ 4}]_i \mathbb{E}[\boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top]_{ii} + \mathbb{E}[\boldsymbol{x}^{\circ 2}]_j \mathbb{E}[\boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top]_{jj} &(17)\\ &+ \mathbb{E}[\boldsymbol{x}^{\circ 2}]_i \mathbb{E}[\boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top]_{ij} \mathbb{E}[\boldsymbol{x}^{\circ 2}]_j &(18)\\ &+ \mathbb{E}[\boldsymbol{x}^{\circ 2}]_i \mathbb{E}[\boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top]_{ji} \mathbb{E}[\boldsymbol{x}^{\circ 2}]_j &(19)\end{aligned}$$

For batch gradient $X_b X_b^\top \boldsymbol{\theta}^{(t)}$, we have:

$$\begin{aligned}\mathbb{E}[X_b X_b^\top \boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top X_b X_b^\top] =& \frac{1}{b^2}\mathbb{E}\Big[(\sum_{i\in[n]_b} \boldsymbol{x}_i \boldsymbol{x}_i^\top)\boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top (\sum_{i\in[n]_b} \boldsymbol{x}_i \boldsymbol{x}_i^\top)\Big] &(20)\\ =& \frac{1}{b^2}\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{\theta}^{(t)} \boldsymbol{\theta}^{(t)} \boldsymbol{x}\boldsymbol{x}^\top] + \frac{1}{b}(b^2 - B)\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]\mathbb{E}[\boldsymbol{\theta}^{(t)}\boldsymbol{\theta}^{(t)}]\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] &(21)\\ =& \frac{1}{b}\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top \boldsymbol{x}\boldsymbol{x}^\top] + \frac{b-1}{b}\Sigma\mathbb{E}[\boldsymbol{\theta}^{(t)} {\boldsymbol{\theta}^{(t)}}^\top]\Sigma &(22)\end{aligned}$$

. □

## C THE PROOF OF THEOREM 2

**Theorem 7.** *Given the noisy linear regression objective function (Eq. 3), under the assumption that $x \sim \mathcal{N}(\mathbf{0}, \Sigma)$ with $\Sigma$ diagonal and $\boldsymbol{\theta}^* = 0$, we can express $\mathrm{C}(\boldsymbol{\theta})$ as a function of $\mathbf{m}(\boldsymbol{\theta})$:*

$$diag(\mathrm{C}(\boldsymbol{\theta})) = 3\Sigma^2 \cdot \mathbf{m}(\boldsymbol{\theta}) \tag{23}$$

*Then we derive following dynamics of expected second moment of $\boldsymbol{\theta}$:*

$$\mathbf{m}(\boldsymbol{\theta}^{(t)}) = R^t\Big(\mathbf{m}(\boldsymbol{\theta}^{(0)}) - \frac{V}{b(\mathrm{I}-R)}\Big) + \frac{V}{b(\mathrm{I}-R)},$$

*Proof.*

$$\boldsymbol{\theta}^{(t+1)}{\boldsymbol{\theta}^{(t+1)}}^\top = (\mathbf{I} - \alpha X_b X_b^\top)\boldsymbol{\theta}^{(t)}{\boldsymbol{\theta}^{(t)}}^\top(\mathbf{I} - \alpha X_b X_b^\top) + \frac{\alpha}{\sqrt{b}}(\mathbf{I} - \alpha X_b X_b^\top)\boldsymbol{\theta}^{(t)}\boldsymbol{\epsilon}_b^\top X_b^\top \tag{24}$$

$$+ \frac{\alpha}{\sqrt{b}} X_b \boldsymbol{\epsilon}_b {\boldsymbol{\theta}^{(t)}}^\top(\mathbf{I} - \alpha X_b X_b^\top) + \frac{\alpha^2}{B} X_b \boldsymbol{\epsilon}_b \boldsymbol{\epsilon}_b^\top X_b^\top \tag{25}$$

$$\tag{26}$$

Since, $\mathbb{E}[\boldsymbol{\epsilon}_b] = \mathbf{0}$, and $\boldsymbol{\epsilon}_b$ is independent with $\boldsymbol{\theta}^{(t)}, X_b$, we have:

$$\mathbb{E}[X_b \boldsymbol{\epsilon}_b {\boldsymbol{\theta}^{(t)}}^\top(\mathbf{I} - \alpha X_b X_b^\top)] = \mathbf{0} \tag{27}$$

$$\mathbb{E}[(\mathbf{I} - \alpha X_b X_b^\top)\boldsymbol{\theta}^{(t)}\boldsymbol{\epsilon}_b^\top X_b^\top] = \mathbf{0} \tag{28}$$

And,

$$\mathbb{E}_{\boldsymbol{\epsilon}_b, X_b}\left[\frac{\alpha^2}{b} X_b \boldsymbol{\epsilon}_b \boldsymbol{\epsilon}_b^\top X_b^\top\right] = \frac{\alpha^2}{b}\mathbb{E}_{X_b}\left[X_b\Big(\mathbb{E}[\boldsymbol{\epsilon}_b \boldsymbol{\epsilon}_b^\top]\Big)X_b\right] \tag{29}$$

$$= \frac{\alpha^2}{b}\mathbb{E}_{X_b}\left[X_b\Big(\sigma_y^2 \mathbf{I}\Big)X_b\right] \tag{30}$$

$$= \frac{\alpha^2 \sigma_y^2}{b}\Sigma \tag{31}$$

Thus,

$$\mathbb{E}[\boldsymbol{\theta}^{(t+1)}{\boldsymbol{\theta}^{(t+1)}}^\top] = (\mathbf{I} - \alpha\Sigma)\mathbb{E}[\boldsymbol{\theta}^{(t)}{\boldsymbol{\theta}^{(t)}}^\top](\mathbf{I} - \alpha\Sigma) + \frac{\alpha^2}{B}(\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{\theta}^{(t)}{\boldsymbol{\theta}^{(t)}}^\top \boldsymbol{x}\boldsymbol{x}^\top] \tag{32}$$

$$- \Sigma\mathbb{E}[\boldsymbol{\theta}^{(t)}{\boldsymbol{\theta}^{(t)}}^\top]\Sigma) + \frac{\alpha^2\sigma_y^2}{B}\Sigma \tag{33}$$

$$\mathbb{E}[\boldsymbol{\theta}^{(t+1)^{\circ 2}}] = \Big((\mathbf{I} - \alpha\Sigma)^2 + \frac{2\alpha^2}{b}\Sigma^2\Big)\mathbb{E}[\boldsymbol{\theta}^{(t)^{\circ 2}}] + \frac{\alpha^2\sigma_y^2}{b} diag(\Sigma) \tag{34}$$

$$\mathbb{E}[\boldsymbol{\theta}^{(t+1)^{\circ 2}}] + \frac{b^{-1}V}{R - \mathrm{I}} = R\Big(\mathbb{E}[\boldsymbol{\theta}^{(t)^{\circ 2}}] + \frac{b^{-1}V}{R - \mathrm{I}}\Big) \tag{35}$$

$$\mathbb{E}[\boldsymbol{\theta}^{(t+1)^{\circ 2}}] = R^{t+1}\Big(\mathbb{E}[\theta_0^{\circ 2}] + \frac{b^{-1}V}{R - \mathrm{I}}\Big) - \frac{b^{-1}V}{R - \mathrm{I}}, \tag{36}$$

$$\mathbf{m}(\boldsymbol{\theta}^{(t)}) = R^t\Big(\mathbf{m}(\boldsymbol{\theta}^{(0)}) - \frac{V}{b(\mathrm{I}-R)}\Big) + \frac{V}{b(\mathrm{I}-R)},$$

where

$$R = (\mathrm{I} - \alpha\Sigma)^2 + 2\alpha^2 b^{-1}\Sigma^2, \quad V = \alpha^2\sigma_y^2 diag(\Sigma). \tag{37}$$

□

## D THE PROOF OF LEMMA 3

**Lemma 8.** *The dynamics of the second moment of the iterate following SVRG update rule is given by,*

$$\mathrm{M}(\boldsymbol{\theta}^{(mT+t+1)}) = \underbrace{(I-\alpha\Sigma)\mathrm{M}(\boldsymbol{\theta}^{(mT+t)})(\mathrm{I}-\alpha\Sigma)}_{\text{① gradient descent shrinkage}} + \underbrace{\frac{\alpha^2}{b}\mathrm{C}(\boldsymbol{\theta}^{(mT+t)})}_{\text{② input noise}} + \underbrace{\frac{\alpha^2\sigma_y^2}{N}\Sigma}_{\text{③ label noise}} \tag{38}$$

$$+ \underbrace{\alpha^2\frac{N+b}{Nb}\mathrm{C}(\boldsymbol{\theta}^{(mT)})}_{\text{④variance due to } \tilde{\mathbf{g}}^{(mT+t)}} \underbrace{-\frac{\alpha^2}{b}\left(\mathrm{C}(\boldsymbol{\theta}^{(mT)})\Big(\mathrm{I}-\alpha\Sigma\Big)^t + \Big(\mathrm{I}-\alpha\Sigma\Big)^t\mathrm{C}(\boldsymbol{\theta}^{(mT)})\right)}_{\text{⑤ Variance reduction from control variate}}$$

*Proof.* For SVRG update rule Eq. 7, we have:

$$\boldsymbol{\theta}^{(mT+t+1)} = \left(\mathrm{I}-\alpha X_b X_b^\top\right)\boldsymbol{\theta}^{(mT+t)} + \alpha\left(X_b X_b^\top - X_N X_N^\top\right)\boldsymbol{\theta}^{(mT)} + \frac{\alpha}{\sqrt{N}}X_N\boldsymbol{\epsilon}_N \tag{39}$$

$$\begin{aligned}
\boldsymbol{\theta}^{(t+1)}\boldsymbol{\theta}^{(t+1)^\top} =& (\mathrm{I}-\alpha X_b X_b^\top)\boldsymbol{\theta}^{(t)}\boldsymbol{\theta}^{(t)^\top}(\mathrm{I}-\alpha X_b X_b^\top) && (40)\\
&+ \alpha(\mathrm{I}-\alpha X_b X_b^\top)\boldsymbol{\theta}^{(t)}\boldsymbol{\theta}^{(mT)^\top}(X_b X_b^\top - X_N X_N^\top) && (41)\\
&+ \alpha(X_b X_b^\top - X_N X_N^\top)\boldsymbol{\theta}^{(mT)}\boldsymbol{\theta}^{(t)^\top}(\mathrm{I}-\alpha X_b X_b^\top) && (42)\\
&+ \alpha^2(X_b X_b^\top - X_N X_N^\top)\boldsymbol{\theta}^{(mT)}\boldsymbol{\theta}^{(mT)^\top}(X_b X_b^\top - X_N X_N^\top) && (43)\\
&+ \frac{\alpha}{\sqrt{N}}\boldsymbol{X}\boldsymbol{\epsilon}_N\boldsymbol{\theta}^{(t)^\top}(\mathrm{I}-\alpha X_b X_b^\top) + \frac{\alpha^2}{\sqrt{N}}\boldsymbol{X}\boldsymbol{\epsilon}_N\boldsymbol{\theta}^{(mT)^\top}(X_b X_b^\top - X_N X_N^\top) && (44)\\
&+ \frac{\alpha}{\sqrt{N}}(\mathrm{I}-\alpha X_b X_b^\top)\boldsymbol{\theta}^{(t)}\boldsymbol{\epsilon}_N^\top X^\top + \frac{\alpha^2}{\sqrt{N}}(X_b X_b^\top - X_N X_N^\top)\boldsymbol{\theta}^{(mT)}\boldsymbol{\epsilon}_N^\top X^\top && (45)\\
&+ \frac{\alpha^2}{N}\boldsymbol{X}\boldsymbol{\epsilon}_N\boldsymbol{\epsilon}_N^\top X^\top && (46)
\end{aligned}$$

Again, since $\boldsymbol{\epsilon}_N = \mathbf{0}$ and $\boldsymbol{\epsilon}_N$ is independent with $X_b, \boldsymbol{\theta}^{(t)}$, we have the expectation of equation 44, equation 45 equal to $\mathbf{0}$. And same as SGD, we also have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\epsilon}_N, X}[\boldsymbol{X}\boldsymbol{\epsilon}_N\boldsymbol{\epsilon}_N{}^\top X_N^\top] =& \mathbb{E}_X\left[X\mathbb{E}_{\boldsymbol{\epsilon}_N}[\boldsymbol{\epsilon}_N\boldsymbol{\epsilon}_N^\top]X^\top\right] && (47)\\
=& \mathbb{E}\left[X\Big(\sigma_y^2\mathrm{I}\Big)X^\top\right] && (48)\\
=& \sigma_y^2\Sigma && (49)
\end{aligned}$$

Moreover,

$$\begin{aligned}
\mathbb{E}[(\mathrm{I}-\alpha X_b X_b^\top)\boldsymbol{\theta}^{(t)}\boldsymbol{\theta}^{(t)^\top}(\mathrm{I}-\alpha X_b X_b^\top)] =& \Big(\mathrm{I}-\alpha\Sigma\Big)\mathbb{E}[\boldsymbol{\theta}^{(t)}\boldsymbol{\theta}^{(t)^\top}](\mathrm{I}-\alpha\Sigma\Big) && (50)\\
&+ \alpha^2 b^{-1}\Big(\mathbb{E}[xx^\top\boldsymbol{\theta}^{(t)}\boldsymbol{\theta}^{(t)^\top}xx^\top] - \Sigma\mathbb{E}[\boldsymbol{\theta}^{(t)}\boldsymbol{\theta}^{(t)^\top}]\Sigma\Big) && (51)
\end{aligned}$$

$$\begin{aligned}
&\mathbb{E}\big[\alpha(X_b X_b^\top - X_N X_N^\top)\boldsymbol{\theta}^{(mT)}\boldsymbol{\theta}^{(t)^\top}(\mathrm{I}-\alpha X_b X_b^\top)\big] && (52)\\
&\quad= -\alpha^2 b^{-1}\Big(\mathrm{I}-\alpha\Sigma\Big)^t\Big(\mathbb{E}[xx^\top\boldsymbol{\theta}^{(mT)}\boldsymbol{\theta}^{(mT)^\top}xx^\top] - \Sigma\mathbb{E}[\boldsymbol{\theta}^{(mT)}\boldsymbol{\theta}^{(mT)^\top}]\Sigma\Big) && (53)
\end{aligned}$$

$$\begin{aligned}
&\mathbb{E}\big[\alpha^2(X_b X_b^\top - X_N X_N^\top)\boldsymbol{\theta}^{(mT)}\boldsymbol{\theta}^{(mT)^\top}(X_b X_b^\top - X_N X_N^\top)\big] && (54)\\
&\quad= \alpha^2\frac{N+b}{Nb}\Big(\mathbb{E}[xx^\top\boldsymbol{\theta}^{(mT)}\boldsymbol{\theta}^{(mT)^\top}xx^\top] - \Sigma\mathbb{E}[\boldsymbol{\theta}^{(mT)}\boldsymbol{\theta}^{(mT)^\top}]\Sigma\Big) && (55)
\end{aligned}$$

Thus,

$$\mathbb{E}[\boldsymbol{\theta}^{(t+1)}\boldsymbol{\theta}^{(t+1)\top}] = \Big(\mathrm{I}-\alpha\Sigma\Big)\mathbb{E}[\boldsymbol{\theta}^{(t)}\boldsymbol{\theta}^{(t)\top}]\Big(\mathrm{I}-\alpha\Sigma\Big) \tag{56}$$

$$+\alpha^2 b^{-1}\Big(\mathbb{E}[xx^\top\boldsymbol{\theta}^{(t)}\boldsymbol{\theta}^{(t)\top}xx^\top]-\Sigma\mathbb{E}[\boldsymbol{\theta}^{(t)}\boldsymbol{\theta}^{(t)\top}]\Sigma\Big) \tag{57}$$

$$+\alpha^2\frac{N+b}{Nb}\Big(\mathbb{E}[xx^\top\boldsymbol{\theta}^{(mT)}\boldsymbol{\theta}^{(mT)\top}xx^\top]-\Sigma\mathbb{E}[\boldsymbol{\theta}^{(mT)}\boldsymbol{\theta}^{(mT)\top}]\Sigma\Big) \tag{58}$$

$$-\alpha^2 b^{-1}\Big(\mathbb{E}[xx^\top\boldsymbol{\theta}^{(mT)}\boldsymbol{\theta}^{(mT)\top}xx^\top]-\Sigma\mathbb{E}[\boldsymbol{\theta}^{(mT)}\boldsymbol{\theta}^{(mT)\top}]\Sigma\Big)\Big(\mathrm{I}-\alpha\Sigma\Big)^t \tag{59}$$

$$-\alpha^2 b^{-1}\Big(\mathrm{I}-\alpha\Sigma\Big)^t\Big(\mathbb{E}[xx^\top\boldsymbol{\theta}^{(mT)}\boldsymbol{\theta}^{(mT)\top}xx^\top]-\Sigma\mathbb{E}[\boldsymbol{\theta}^{(mT)}\boldsymbol{\theta}^{(mT)\top}]\Sigma\Big) \tag{60}$$

$$+\frac{\alpha^2\sigma_y^2}{N}\Sigma \tag{61}$$

Under our definition, it can be expressed as:

$$\mathrm{M}(\boldsymbol{\theta}^{(mT+t+1)}) = \underbrace{(I-\alpha\Sigma)\mathrm{M}(\boldsymbol{\theta}^{(mT+t)})(\mathrm{I}-\alpha\Sigma)}_{\text{① gradient descent shrinkage}} + \underbrace{\frac{\alpha^2}{b}\mathrm{C}(\boldsymbol{\theta}^{(mT+t)})}_{\text{② input noise}} + \underbrace{\frac{\alpha^2\sigma_y^2}{N}\Sigma}_{\text{③ label noise}} \tag{62}$$

$$+\underbrace{\alpha^2\frac{N+b}{Nb}\mathrm{C}(\boldsymbol{\theta}^{(mT)})}_{\text{④variance due to } \tilde{\mathbf{g}}^{(mT+t)}} \underbrace{-\frac{\alpha^2}{b}\left(\mathrm{C}(\boldsymbol{\theta}^{(mT)})\Big(\mathrm{I}-\alpha\Sigma\Big)^t+\Big(\mathrm{I}-\alpha\Sigma\Big)^t\mathrm{C}(\boldsymbol{\theta}^{(mT)})\right)}_{\text{⑤ Variance reduction from control variate}}$$

□

## E THE PROOF OF THEOREM 4

**Theorem 9.** *Given the noisy linear regression objective function (Eq. 3), under the assumption that $x \sim \mathcal{N}(\mathbf{0}, \Sigma)$ with $\Sigma$ diagonal and $\boldsymbol{\theta}^* = 0$, the dynamics for SVRG in $\mathbf{m}(\boldsymbol{\theta})$ is given by:*

$$\mathbf{m}(\boldsymbol{\theta}^{((m+1)T)}) = \lambda(\alpha, b, T, \Sigma, N)\mathbf{m}(\boldsymbol{\theta}^{(mT)}) + \frac{\mathrm{I}-R^T}{\mathrm{I}-R}\frac{V}{N}, \tag{63}$$

$$\lambda(\alpha, B, T, \Sigma, N) = R^T\Big(\mathrm{I}+\frac{Q}{P-R}+\frac{F}{R-\mathrm{I}}\Big)-\frac{P^TQ}{P-R}-\frac{F}{R-\mathrm{I}} \tag{64}$$

*Proof.* Form lemma 3 and lemma 6, we can get:

we will get

$$\mathbb{E}[\boldsymbol{\theta}^{(mT+t+1)\circ 2}] = R\mathbb{E}[\boldsymbol{\theta}^{(mT+t)\circ 2}] - P^tQ\mathbb{E}[\boldsymbol{\theta}^{(mT)\circ 2}] + F\mathbb{E}[\boldsymbol{\theta}^{(mT)\circ 2}] + N^{-1}V \tag{65}$$

$$\mathbb{E}[\boldsymbol{\theta}^{(mT+t+1)\circ 2}] + \frac{P^{t+1}Q\mathbb{E}[\boldsymbol{\theta}^{(mT)\circ 2}]}{P-R} + \frac{F\mathbb{E}[\boldsymbol{\theta}^{(mT)\circ 2}]+N^{-1}V}{R-\mathrm{I}} \tag{66}$$

$$= R\left(\mathbb{E}[\boldsymbol{\theta}^{(mT+t)\circ 2}] + \frac{P^tQ\mathbb{E}[\boldsymbol{\theta}^{(mT)\circ 2}]}{P-R} + \frac{F\mathbb{E}[\boldsymbol{\theta}^{(mT)\circ 2}]+N^{-1}V}{R-\mathrm{I}}\right) \tag{67}$$

$$\mathbb{E}[\boldsymbol{\theta}^{((m+1)T)\circ 2}] + \frac{P^TQ\mathbb{E}[\boldsymbol{\theta}^{(mT)\circ 2}]}{P-R} + \frac{F\mathbb{E}[\boldsymbol{\theta}^{(mT)\circ 2}]+N^{-1}V}{R-\mathrm{I}} \tag{68}$$

$$= R^T\left(\mathbb{E}[\boldsymbol{\theta}^{(mT)\circ 2}] + \frac{P^0Q\mathbb{E}[\boldsymbol{\theta}^{(mT)\circ 2}]}{P-R} + \frac{F\mathbb{E}[\boldsymbol{\theta}^{(mT)\circ 2}]+N^{-1}V}{R-\mathrm{I}}\right) \tag{69}$$

In other word, equation 68 describe the dynamic of parameter second moment between two nearby snapshots,

$$\mathbb{E}[\boldsymbol{\theta}^{((m+1)T)^{\circ 2}}] = \lambda(\alpha, B, T, N, \Sigma)\mathbb{E}[\boldsymbol{\theta}^{(mT)^{\circ 2}}] + \frac{R^T - \mathrm{I}}{R - \mathrm{I}} N^{-1} V \tag{70}$$

which in our definition can be formulized as:

$$\mathbf{m}(\boldsymbol{\theta}^{((m+1)T)}) = \lambda(\alpha, b, T, \Sigma, N)\mathbf{m}(\boldsymbol{\theta}^{(mT)}) + \frac{\mathrm{I} - R^T}{\mathrm{I} - R}\frac{V}{N}, \tag{71}$$

$$\lambda(\alpha, B, T, \Sigma, N) = R^T\Big(\mathrm{I} + \frac{Q}{P - R} + \frac{F}{R - \mathrm{I}}\Big) - \frac{P^T Q}{P - R} - \frac{F}{R - \mathrm{I}} \tag{72}$$

where

$$\lambda(\alpha, B, T, N, \Sigma) = R^T\Big(\mathrm{I} + \frac{Q}{P - R} + \frac{F}{R - \mathrm{I}}\Big) - \frac{P^T Q}{P - R} - \frac{F}{R - \mathrm{I}} \tag{73}$$

□

## F THE PROOF OF COROLLARY 5

**Corollary 10.** *Without the label noise, the dynamics of the second moment following SGD is given by,*

$$\mathbf{m}(\boldsymbol{\theta}^{(t)}) = R^t \mathbf{m}(\boldsymbol{\theta}^{(0)}),$$

*and the dynamics of SVRG is given by,*

$$\mathbf{m}(\boldsymbol{\theta}^{((m+1)T)}) = \lambda(\alpha, b, T, \Sigma, N)\mathbf{m}(\boldsymbol{\theta}^{(mT)}),$$

*where $\lambda$ is defined in Eq.( 10).*

*Proof.* For the setting without label noise, i.e. $\sigma_y^2 = 0$, we directly draw the corollary about setting without label noise, based on Theorem 2 and Theorem 4. By setting $\sigma_y^2 = 0$, we can draw:

$$\mathbf{m}(\boldsymbol{\theta}^{(t)}) = R^t \mathbf{m}(\boldsymbol{\theta}^{(0)}),$$

and the dynamics of SVRG is given by,

$$\mathbf{m}(\boldsymbol{\theta}^{((m+1)T)}) = \lambda(\alpha, b, T, \Sigma, N)\mathbf{m}(\boldsymbol{\theta}^{(mT)}),$$

where $\lambda$ is defined in Eq.( 10). □