Bounding the expected run-time of nonconvex optimization with early stopping

Thomas Flynn **Kwang Min Yu** **Abid Malik** **Nicolas D'Imperio** **Shinjae Yoo**

Computational Science Initiative
Brookhaven National Laboratory
Upton, NY 11973

Abstract

This work examines the convergence of stochastic gradient-based optimization algorithms that use early stopping based on a validation function. The form of early stopping we consider is that optimization terminates when the norm of the gradient of a validation function falls below a threshold. We derive conditions that guarantee this stopping rule is well-defined, and provide bounds on the expected number of iterations and gradient evaluations needed to meet this criterion. The guarantee accounts for the distance between the training and validation sets, measured with the Wasserstein distance. We develop the approach in the general setting of a first-order optimization algorithm, with possibly biased update directions subject to a geometric drift condition. We then derive bounds on the expected running time for early stopping variants of several algorithms, including stochastic gradient descent (SGD), decentralized SGD (DSGD), and the stochastic variance reduced gradient (SVRG) algorithm. Finally, we consider the generalization properties of the iterate returned by early stopping.

1 INTRODUCTION

This work considers the minimization of a differentiable and possibly nonconvex objective function:

$$\min_{x\in\mathbb{R}^d} f(x). \tag{1}$$

For nonconvex problems, a generally accepted notion of success for algorithms that use only first-order information is that an *approximate stationary point* is generated. These are points $x \in \mathbb{R}^d$ where the norm of the gradient of f is small. In a typical machine learning scenario, f is the average loss over a dataset of training examples, and it is common to solve problem (1) using stochastic gradient-based optimization, for instance, stochastic gradient descent (SGD; see Algorithm 1). The success of SGD in machine learning problems has led to many extensions of the algorithm, including variance-reduced and distributed variants (reviewed in Section 1.1).

A common approach to stopping optimization in practice is to use early stopping, in which a performance criterion is periodically evaluated on a validation function and optimization terminates once this condition is met. However, there is little theoretical work on the run-time of nonconvex optimization with such early stopping rules. In general, one would expect that the run-time and performance will depend on several factors, including the similarity between the validation and training functions, the desired level of solution accuracy, and internal settings of the optimization algorithm, such as learning rates.

In this work, we carry out an analysis of early stopping when the criterion is that the algorithm has generated an approximate stationary point for a validation function. Formally, we consider the stopping time defined as the first time, or iteration number, that an iterate has the property of being an approximate stationary point for the validation function, and we derive upper bounds on the expected value of this stopping time. Furthermore, although the stopping time is defined in terms of stationarity of the *validation* function, we also derive a bound on the stationarity gap of the *training* function at the resulting iterate, in terms of the Wasserstein distance between the training and validation sets. As an extension, we describe how to leverage Wasserstein concentration results to bound the expected stationarity gap with respect to the *testing* distribution from which both the training and validation datasets are drawn.

The analysis is carried out for several procedures, including stochastic gradient descent (SGD), decentralized SGD (DSGD), and the stochastic variance reduced gradi-

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

ent (SVRG) algorithm, The result is new bounds on the expected number of Incremental First-order Oracle (IFO) calls needed by these algorithms to generate approximate stationary points. We believe the general technique used to obtain the results will be useful for analyzing the expected running time of other optimization algorithms as well.

Main contributions. Our main contributions include:

- We present a non-asymptotic analysis of SGD with early stopping that leads to a bound on the expected amount of resources needed to find approximate stationary points of the training function, including the number of iterations (Proposition 10) and gradient evaluations (Corollary 12). The analysis allows for biases in the update direction, subject to a geometric drift condition on the error terms (specified in Assumption 9.)

- We specialize the results to decentralized SGD, a variant of SGD designed for distributed computation, resulting in upper bounds on the number of iterations (Proposition 16) and gradient evaluations (Corollary 17) needed by the algorithm. This is done by modeling DSGD as a biased form of SGD, whose bias is controlled in part by the diffusion coefficient of the network communication graph.

- We derive a run-time bound for a variant of nonconvex SVRG with early stopping, obtaining a bound on the expected number of iterations (Proposition 18) and IFO calls (Corollary 19) need to generate approximate stationary points.

- We demonstrate how Wasserstein concentration bounds can be leveraged to bound the generalization performance of the iterate returned by the algorithms (Section 6), expressed in terms of the number of samples used to construct the datasets, and properties of the testing distribution.

1.1 Related work

The study of stochastic optimization goes back (at least) to the pioneering work of Robbins and Monro (Robbins & Monro, 1951). Subsequent developments include the ordinary differential equation (ODE) method (Ljung, 1977) and stochastic approximation (Kushner & Clark, 1978), which emphasizes the asymptotic behavior of the algorithms. Asymptotic performance of biased SGD has been considered in (Bertsekas & Tsitsiklis, 2000) which establishes the asymptotic convergence of the algorithm to stationary points.

The randomized stochastic gradient (RSG) method (Ghadimi & Lan, 2013) uses randomization to obtain a non-asymptotic performance guarantee for SGD applied to nonconvex functions. In one interpretation of RSG, the algorithm (e.g., SGD) is run for a fixed number of iterations, and a random iterate is selected as the final output of optimization (alternatively, the algorithm is executed for a random number of steps, after which the final iterate is returned.) The randomization technique has became a standard tool for analyzing optimization algorithms in the nonconvex setting (Ghadimi & Lan, 2016; Lian et al., 2017; Reddi et al., 2016; Zhang et al., 2016; Lei et al., 2017; Lian et al., 2015). Follow-up works have employed randomization for the analysis of nonconvex optimization in diverse algorithmic settings, such as asynchronous (Lian et al., 2015) and decentralized (Lian et al., 2017) optimization.

Machine learning problems often involve an objective that is a finite sum of functions, and, in this setting, variance reduction techniques lead to improved rates of convergence over SGD (Johnson & Zhang, 2013; L. Roux et al., 2012; Defazio et al., 2014). Analysis of variance reduction has extended beyond convex functions, from an application to principal components analysis (Shamir, 2015) to general nonconvex functions (Allen-Zhu & Hazan, 2016; Reddi et al., 2016; Lei et al., 2017).

In contrast to the aforementioned works, in which randomization is used to analyze performance in the nonconvex case, this work considers algorithms that use a different approach to stopping optimization, based on periodically evaluating a performance criterion with respect to a validation function. There are several variants of early stopping that appear in practice. For instance, one approach is to train until the error on a validation set increases (Wang & Carreira-Perpinan, 2012), (Dai & Le, 2015), or there is no improvement for a number of epochs (Zhang et al., 2019). Alternatively, one can train the model for a fixed number of epochs, and then take the parameter from the epoch at which validation error is lowest (Jaderberg et al., 2017), (Lee et al., 2018), (Franceschi et al., 2019). Despite the prevalence of early stopping, there is comparatively little work on the analysis of this strategy in the nonconvex setting, and our work aims to fill this gap

Several recent works also have explored the average amount of resources needed to reach a desired performance level in optimization. The expected running time of a stochastic trust region algorithm is given in (Blanchet et al., 2016), based on a renewal-reward martingale argument. This proof technique was also used to analyze the expected run time of a stochastic line search method (Paquette & Scheinberg, 2018). Our convergence analysis is similar in spirit, as we also are interested in the expected amount of time or other resources required to meet the performance guarantee. However, our focus is on different algorithms (SGD, DSGD, and SVRG), and the variants of these algorithms that we consider contain explicit stopping mechanisms based on validation functions. Other recent work considering the theoretical aspects of early stopping include (Duvenaud et al., 2016), where the authors developed an interpretation of early stopping in terms of variational Bayesian inference. Early stopping for a least squares problem in a reproducing kernel Hilbert space has been treated in (Lin & Rosasco, 2016), while the implications of early stopping on generalization were studied in (Hardt et al., 2016). To the authors' knowledge, the present work is the first to analyze run-time when using a validation function for early stopping in nonconvex optimization.

2 PRELIMINARIES

Let $f : \mathbb{R}^q \times \mathbb{R}^d \to \mathbb{R}$ be a loss function whose value we denote by $f(y, x)$. Intuitively, the variable y represents an (input, output) pair, and x represents the parameters of a model. Throughout, we shall assume that the gradient of the loss function is Lipschitz continuous:

Assumption 1. *The function $f : \mathbb{R}^q \times \mathbb{R}^d \to \mathbb{R}$ is bounded from below by $f^* \in \mathbb{R}$, and $\nabla_x f$ is L-Lipschitz continuous as a function of x: $\forall y \in \mathbb{R}^q, x_1, x_2 \in \mathbb{R}^d$,*

$$\|\nabla_x f(y, x_1) - \nabla_x f(y, x_2)\| \leq L\|x_1 - x_2\|.$$

This is a standard assumption that is also referred to as smoothness of the loss function. Where appropriate, we will make a distinction between the training function f_T, which is used to calculate gradients, and a validation function f_V used to decide when to stop training:

Assumption 2. *The function f_T is defined using a set $Y_T \subseteq \mathbb{R}^q$ as $f_T(x) = \frac{1}{n_T}\sum_{y \in Y_T} f(y, x)$, where $n_T = |Y_T|$, and the function f_V is defined using a set $Y_V \subseteq \mathbb{R}^q$ as $f_V(x) = \frac{1}{n_V}\sum_{y \in Y_V} f(y, x)$, where $n_V = |Y_V|$.*

Note that there is no assumption that the validation and training sets are disjoint. At times we will assume a bound on the variance of stochastic gradients of f_T:

Assumption 3. *There is a $\sigma_v^2 \geq 0$ such that $\forall\ x \in \mathbb{R}^d$,*

$$\frac{1}{n_T}\sum_{y \in Y_T} \|\nabla_x f(y, x) - \nabla f_T(x)\|^2 \leq \sigma_v^2.$$

In the SGD and DSGD algorithms considered below, optimization takes place on the training function, while the stopping criteria is evaluated using the validation function. To guarantee that this leads to well-defined behavior, we will make use of a bound on the distance between the training and validation functions. Intuitively, the functions f_T and f_V will be close when the datasets Y_T and Y_V are similar. Formally, the datasets Y_T and Y_V determine probability measures μ_T and μ_Y, defined as $\mu_T = \frac{1}{n_T}\sum_{y \in Y_T} \delta_y$ and $\mu_V = \frac{1}{n_V}\sum_{y \in Y_V} \delta_y$, respectively, where δ_y is the delta measure $\delta_y(A) = 1_{y \in A}$ for all sets A. We can compare these measures using the Wasserstein distance, which is defined as follows.

For $q \geq 1, p \geq 1$, denote by $\mathcal{P}_p(\mathbb{R}^q)$ the probability measures on $\mathbb{R}^q$ with finite moments of order p. Recall that a coupling of probability measures μ_1 and μ_2 is a probability measure γ on $\mathbb{R}^q \times \mathbb{R}^q$ such that for all measurable sets A, $\gamma(A \times \mathbb{R}^q) = \mu_1(A)$ and $\gamma(\mathbb{R}^q \times A) = \mu_2(A)$. Intuitively, a coupling transforms data distributed like μ_1 into a data distributed according to μ_2 (and vice versa). The p-Wasserstein distance on $\mathcal{P}_p(\mathbb{R}^q)$, denoted by d_p, is defined as:

$$d_p(\mu_1, \mu_2) = \inf_{\gamma \in \Gamma(\mu_1, \mu_2)} \left(\mathop{\mathbb{E}}_{(x_1,\, x_2) \sim \gamma} [\, \|x_1 - x_2\|^p \,] \right)^{\frac{1}{p}} \quad (2)$$

where $\Gamma(\mu_1, \mu_2)$ is the set of all couplings of μ_1 and μ_2. For more details, including a proof that this definition indeed satisfies the axioms of a metric, the reader is referred to Chapter 6 of (Villani, 2008).

In order to link the distance of the functions ∇f_T and ∇f_V to the distance between the empirical measures μ_T and μ_V, the following assumption will be useful:

Assumption 4. *The function $\nabla_x f$ is G-Lipschitz continuous as a function of y: $\forall x \in \mathbb{R}^d, y_1, y_2 \in \mathbb{R}^q$,*

$$\|\nabla_x f(y_1, x) - \nabla_x f(y_2, x)\| \leq G\|y_1 - y_2\|.$$

Assumption 4 implies the following bound: $\forall x \in \mathbb{R}^d$,

$$\|\nabla f_V(x) - \nabla f_T(x)\| \leq G d_1(\mu_V, \mu_T). \quad (3)$$

To see that (3) follows from Assumption 4, let γ be any coupling of μ_V and μ_T. Then

$$\begin{aligned} \|\nabla f_V(x) - \nabla f_T(x)\| &= \left\| \mathop{\mathbb{E}}_{(u,v) \sim \gamma} [\nabla_x f(u, x) - \nabla_x f(v, x)] \right\| \\ &\leq G \mathop{\mathbb{E}}_{(u,v) \sim \gamma} [\|y_1 - y_2\|]\,. \end{aligned}$$

Taking the infimum over all couplings of μ_V and μ_T yields Equation (3). For an example of a function that satisfies Assumption 4, consider the following:

Example 5. Let $g : \mathbb{R}^q \times \mathbb{R}^d \to \mathbb{R}$ be any smooth (that is, infinitely differentiable) function, and let $h : \mathbb{R}^d \to \mathbb{R}^d$ be the function that applies the hyperbolic tangent function to each of its components: $h(x) = (\tanh(x_1), \ldots, \tanh(x_d))$. Define $f(y, x) = g(y, h(x))$, and further suppose that the training data are bounded: $\|y\| \leq J$ for all $y \in Y_T \cup Y_V$. To guarantee that Assumption 4 is satisfied, it is sufficient that the derivative $\frac{\partial^2 f}{\partial x \partial y}(y, x)$ is bounded as a bilinear map, uniformly in y and x (Proposition 2.4.8 in (Abraham et al., 2012)). It can be shown that this is indeed the case, and we may take $G = \sup_{\|y\| \leq J, \|x\| \leq \sqrt{d}} \|\frac{\partial^2 g}{\partial x \partial y}(y, x)\|$. We defer the details to an appendix.

In our analyses the notion of success is that an algorithm generates an approximate stationary point:

Definition 6. A point $x \in \mathbb{R}^d$ is an *ϵ-approximate stationary point* of f if $\|\nabla f(x)\|^2 \leq \epsilon$.

We measure the complexity of algorithms according to how many function value and gradient queries they make. Formally, an IFO is defined as follows (Agarwal & Bottou, 2015):

Definition 7. An IFO takes a parameter x and an input y and returns the pair $(f(y, x), \nabla_x f(y, x))$.

In Appendix A, we briefly recall the notion of filtration, stopping times, and other concepts from stochastic processes that will be used in this paper.

3 BIASED SGD

In this section we present our analysis of SGD with early stopping. The steps of the procedure are detailed in in Algorithm 1. Starting from an initial point x_1, the parameter is updated at each iteration with an approximate gradient h_n, using a step-size η. The norm of the gradient of the validation function is evaluated every m iterations, and the algorithm ends when the squared norm of the gradient falls below a threshold ϵ.

We assume that the update direction h_t is a sum of two components, v_t and Δ_t, that represent an unbiased gradient estimate and an error term, respectively:

$$h_t = v_t + \Delta_t. \tag{4}$$

Let $\{\mathcal{F}_t\}_{t \geq 0}$ be a filtration such that x_1 is $\mathcal{F}_0$-measurable, and for all $t > 1$, the variables (v_t, Δ_t) are $\mathcal{F}_t$-measurable. Our assumptions on the v_t are as follows.

Assumption 8. *For any $t \geq 1$, it holds that*

$$\mathbb{E}\left[v_t - \nabla f_T(x_t) \mid \mathcal{F}_{t-1}\right] = 0, \tag{5}$$

$$\mathbb{E}\left[\|v_t - \nabla f_T(x_t)\|^2 \mid \mathcal{F}_{t-1}\right] \leq \sigma_v^2. \tag{6}$$

Assumption 8 states that the update terms v_t are valid approximations to the gradient, and have bounded variance. For the bias terms we assume the following growth condition:

Algorithm 1 SGD with early stopping

```
input: Initial point $x_1 \in \mathbb{R}^d$
$t = 1$
/* check if stopping criteria is satisfied. */
while $\|\nabla f_V(x_t)\|^2 > \epsilon$ do
    /* if not, perform an epoch of training. */
    for $n = t$ to $t + m - 1$ do
        $x_{n+1} = x_n - \eta h_n$
    end
    $t = t + m$
end
/* once criteria is met, return current iterate. */
return $x_t$
```

Assumption 9. *There is a sequence of random variables $V_1, V_2, \ldots$, and $U_1, U_2, \ldots$ such that for all $t \geq 1$ the pair (V_t, U_t) is $\mathcal{F}_t$-measurable, $\|\Delta_t\|^2 \leq V_t$, and the V_t satisfy the following geometric drift condition: For some pair of constants $\alpha \in [0, 1)$ and $\beta \geq 0$,*

$$V_1 \leq \beta, \tag{7}$$

$$\forall t \geq 2, \quad V_t \leq \alpha V_{t-1} + U_{t-1}, \tag{8}$$

$$\forall t \geq 1, \quad \mathbb{E}\left[U_t \mid \mathcal{F}_{t-1}\right] \leq \beta. \tag{9}$$

Assumption 9 models a scenario where the bias dynamics are a combination of contracting and expanding behaviors. Contraction shrinks the error and is represented by a factor α. External noise, represented by the U_t terms, prevents the error from vanishing completely. Note that the assumption would be satisfied in the unbiased case by simply setting $V_t = 0$.

We can now state our result on the expected number of iterations required by biased SGD with early stopping:

Proposition 10. *Let $\{x_t\}_{t \geq 1}$ be as in Algorithm 1. Let Assumptions 1, 2, 4, 8, and 9 hold. For $\epsilon > 0$, let $\tau(\epsilon)$ be the stopping time*

$$\tau(\epsilon) = \inf\{n \geq 1 \mid n \equiv 1 (\operatorname{mod} m) \textit{ and } \|\nabla f_V(x_n)\|^2 \leq \epsilon\}.$$

Suppose that $\eta \leq \frac{1}{L}$ and

$$\epsilon - 4Lm\eta\sigma_v^2 - 4m\beta/(1-\alpha) - 2G^2 d_1(\mu_V, \mu_T)^2 > 0.$$

Then

$$\mathbb{E}[\tau(\epsilon)] \leq \frac{G^2 d_1(\mu_V, \mu_T)^2 + 2(f_T(x_1) - f^*)/\eta + \epsilon + 2\beta/(1-\alpha)}{\epsilon/(2m) - 2L\eta\sigma_v^2 - 2\beta/(1-\alpha) - G^2 d_1(\mu_V, \mu_T)^2/m}.$$

Furthermore, the gradient of f_T at $x_{\tau(\epsilon)}$ satisfies

$$\|\nabla f_T(x_{\tau(\epsilon)})\|^2 \leq \left(\sqrt{\epsilon} + G d_1(\mu_V, \mu_T)\right)^2. \tag{10}$$

We present a sketch of the proof below, saving the full proof for an appendix.

Proof sketch. To emphasize the main ideas, we make the simplifying assumptions that there are no error terms ($\Delta_t = 0$), the Lipschitz constant for the gradient is $L = 2$, and the training and validation sets are equal ($Y_T = Y_V$). To establish a bound on $\mathbb{E}[\tau(\epsilon)]$, we first consider the truncated stopping time $\tau(\epsilon) \wedge n$, defined as the minimum of $\tau(\epsilon)$ and an arbitrary iteration number n. We find an upper bound on $\mathbb{E}[\tau(\epsilon) \wedge n]$ that is independent of n, and appeal to the monotone convergence theorem to conclude that this same bound must hold for $\mathbb{E}[\tau(\epsilon)]$.

Using a quadratic growth bound that follows from the Lipschitz property of the gradient (Equation (18) in the appendix), for any n it holds that

$$f(x_{\tau(\epsilon)\wedge n+1}) \leq f(x_1) - \eta(1-\eta) \sum_{t=1}^{\tau(\epsilon)\wedge n} \|\nabla f(x_t)\|^2 \\ - \eta(1-2\eta) \sum_{t=1}^{\tau(\epsilon)\wedge n} \nabla f(x_t)^T \delta_t + \eta^2 \sum_{t=1}^{\tau(\epsilon)\wedge n} \|\delta_t\|^2.$$

Taking expectations and applying Proposition 23, we obtain

$$\mathbb{E}\left[f(x_{\tau(\epsilon)\wedge n+1})\right] \leq f(x_1) - \eta(1-\eta)\mathbb{E}\left[\sum_{t=1}^{\tau(\epsilon)\wedge n} \|\nabla f(x_t)\|^2\right] \\ + \eta^2 \sigma_v^2 \, \mathbb{E}[\tau(\epsilon) \wedge n].$$

Rearranging terms and noting that $f(x_{\tau(\epsilon)\wedge n+1}) \geq f^*$,

$$\eta(1-\eta)\mathbb{E}\left[\sum_{t=1}^{\tau(\epsilon)\wedge n} \|\nabla f(x_t)\|^2\right] \leq \\ f(x_1) - f^* + \eta^2 \sigma_v^2 \, \mathbb{E}[\tau(\epsilon) \wedge n].$$

Next, using the definition of $\tau(\epsilon)$, we have

$$\frac{\epsilon\left(\mathbb{E}[\tau(\epsilon) \wedge n] - 1\right)}{m} \leq \mathbb{E}\left[\sum_{t=1}^{\tau(\epsilon)\wedge n} 1_{t \equiv 1 \pmod m} \|\nabla f(x_t)\|^2\right] \\ \leq \mathbb{E}\left[\sum_{t=1}^{\tau(\epsilon)\wedge n} \|\nabla f(x_t)\|^2\right].$$

Combining the previous two equations, upon rearranging terms we obtain

$$\eta\left((1-\eta)\frac{\epsilon}{m} - \eta\sigma_v^2\right)\mathbb{E}[\tau(\epsilon)\wedge n] \leq f(x_1) - f^* + \frac{\eta(1-\eta)\epsilon}{m}$$

The coefficient on the left hand side of this equation is positive provided that

$$\eta < \frac{\epsilon}{m\sigma^2 + \epsilon}$$

Choose a $c \in (0,1)$ and let $\eta = c \cdot \frac{\epsilon}{m\sigma^2 + \epsilon}$. Rearranging terms, and letting $n \to \infty$, we obtain

$$\mathbb{E}[\tau(\epsilon)] \leq \frac{(f(x_1) - f^*)m^2\sigma^2}{\epsilon^2 c(1-c)} + \mathcal{O}\left(\frac{m}{\epsilon}\right).$$

We refer the reader to the appendix for a complete proof. □

Note that the condition on η in the proposition requires that it scales inversely with the epoch length m. Whether this argument can be refined to yield conditions on ϵ that are independent of m, we leave as an open question. Let us note that the situation is somewhat more favorable in the case of SVRG. In our analysis of SVRG below, the introduction of early stopping does not produce any new constraints on the step-size.

Proposition 10 implies that SGD can find ϵ-approximate stationary points, for any $\epsilon > 4m\beta/(1-\alpha) + 2G^2 d_1(\mu_V, \mu_T)^2$. We can relax this condition, allowing for smaller values of ϵ, by assuming a coupling between the step-size and the expansion coefficient, as demonstrated in the next corollary.

Corollary 11. *Let Assumptions 1, 2, 4, 8, and 9 hold. In the context of Proposition 10, let the constant β be of the form $\beta = \eta R$ for some $R \geq 0$, and suppose that $\epsilon > 2G^2 d_1(\mu_V, \mu_T)^2$. Let $c \in (0,1)$ and let the step-size be*

$$\eta = c \cdot \min\left\{\frac{1}{L}, \frac{\epsilon/2 - G^2 d_1(\mu_V, \mu_T)^2}{m(2L\sigma_v^2 + 2R/(1-\alpha))}\right\}. \quad (11)$$

Then

$$\mathbb{E}[\tau(\epsilon)] = \mathcal{O}\left(\frac{m^2\left(1 + R/(1-\alpha)\right)}{(1-c)\, c\, (\epsilon - 2G^2 d_1(\mu_V, \mu_T)^2)^2}\right).$$

The reader may refer to the full proof contained in an appendix for the complete formula, including lower order terms. This result will be used below, in our analysis of DSGD.

We now specialize the results in the case of using SGD to minimize a finite sum using unbiased gradient estimates.

Corollary 12. *Let Assumptions 1, 2, 3, 4 hold. Suppose each gradient estimate is obtained by selecting an element $y_t \in Y_T$ uniformly at random and setting $v_t = \nabla_x f(y_t, x_t)$. Let $\epsilon > 2G^2 d_1(\mu_V, \mu_T)^2$ and consider running SGD with epoch length $m \geq 1$, and step-size η as defined in* (11) *with $c = 1/2$. Then the expected number of IFO calls used by SGD with early stopping is*

$$\mathbb{E}\left[\text{IFO}\,(\epsilon)\right] = \mathcal{O}\left(\frac{mn_V + m^2}{(\epsilon - 2G^2 d_1(\mu_V, \mu_T)^2)^2} + n_V\right)$$

Algorithm 2 DSGD with early stopping

```
input: Node id $i$, initial parameters $x_1^i$.
$t = 1$
/* check if stopping criteria is satisfied. */
while $\|\nabla f_V(\overline{x}_t)\|^2 > \epsilon$ do
    /* if not, perform an epoch of training. */
    for $n = t$ to $t + m - 1$ do
        /* perform local average and descent steps. */
        $x_{n+1}^i = \sum_{j=1}^{M} a_{i,j} x_n^j - \eta v_n^i$
    end
    $t = t + m$
end
/* once criteria is met, return current iterate. */
return $\overline{x}_t$
```

Note that when $d_1(\mu_V, \mu_T) = 0$, this result states that the expected IFO complexity is $\mathcal{O}(1/(\epsilon^2))$. This can be compared with the RSG algorithm, where $\mathcal{O}(1/(\epsilon^2))$ iterations are sufficient for the expected squared norm of the gradient at a random iterate to be at most ϵ (Corollary 2.2 in (Ghadimi & Lan, 2013)).

4 DECENTRALIZED SGD

In this section we analyze the expected running time of decentralized SGD (DSGD), a variant of SGD designed for distributed optimization across a network of compute nodes. Recently, a randomization-based analysis of DSGD was presented in (Lian et al., 2017). We complement that analysis by studying the expected running time of a variant of DSGD with early stopping. The main idea is to model the algorithm as a biased form of SGD that satisfies the geometric drift condition described in Assumption 9.

The steps of DSGD are shown in Algorithm 2. The procedure involves $M > 0$ worker nodes that participate in the optimization, and an $M \times M$ communication matrix a describing the connectivity among the workers; $a_{i,j} > 0$ means that workers i and j will communicate at each iteration, while $a_{i,j} = 0$ means there is no direct communication between those workers. At each step of optimization, every node computes a weighted average of the parameters in its local neighborhood, as determined by the connectivity matrix. This is combined with a local gradient approximation to obtain the new parameter at the worker. Every m epochs, the norm of gradient of the validation function is evaluated at the average parameter across the system, denoted $\overline{x}_t$:

$$\overline{x}_t = \frac{1}{M} \sum_{i=1}^{M} x_i \tag{12}$$

When this norm falls below a threshold, the algorithm terminates, returning the final value of $\overline{x}_t$.

The intuitive justification for DSGD is that it may be more efficient compared to naive approaches to parallelizing SGD, since two nodes i and j need not communicate when $a_{i,j} = 0$. In (Lian et al., 2017) those authors offer theoretical support for the superiority of DSGD. In the present work, our goal is to analyze the expected running time of DSGD as an example of how the theory developed above may be applied in practice.

For the analysis, define the filtration $\{\mathcal{F}_t\}_{t \geq 0}$ as follows:

$$\begin{aligned} \mathcal{F}_0 &= \sigma\big(\, \{x_1^i \,|\, 1 \leq i \leq M\}\,\big), \\ \forall t \geq 1, \quad \mathcal{F}_t &= \sigma\big(\, \{x_1^i, v_n^i \,|\, 1 \leq n \leq t,\, 1 \leq i \leq M\}\,\big). \end{aligned}$$

We assume that the gradient estimates used at each worker are unbiased and have bounded variance.

Assumption 13. *For any $t \geq 1$ and $1 \leq i \leq M$,*

$$\mathbb{E}\left[v_t^i - \nabla f_T(x_t^i) \mid \mathcal{F}_{t-1}\right] = 0, \tag{13}$$

$$\mathbb{E}\left[\left\|v_t^i - \nabla f_T(x_t^i)\right\|^2 \mid \mathcal{F}_{t-1}\right] \leq \sigma_v^2. \tag{14}$$

The connectivity matrix a is subject to the same conditions as in (Lian et al., 2017), stated below as Assumption 14. In this Assumption, $\lambda_i(a)$ refers to the eigenvalues of the matrix a in nonincreasing order: $\lambda_i(a) \geq \lambda_{i+1}(a)$ for $1 \leq i < M$.

Assumption 14. *The $M \times M$ connectivity matrix, denoted a, is symmetric and stochastic. The diffusion coefficient, denoted by ρ and defined as $\rho = \max_{2 \leq i \leq M} |\lambda_i(a)|^2$, satisfies $\rho < 1$.*

We will show that the sequence of averages $\overline{x}_t$ for $t = 1, 2, \ldots$ can be modeled in terms of biased SGD, using the tools from Section 3. This involves showing that the distance between local parameter values and the system average obey a geometric drift condition, and furthermore, this distance can be controlled through the step-size.

Proposition 15. *Let Assumptions 1, 2, 13, and 14 hold, and let the step-size satisfy*

$$\eta \leq \frac{1 - \sqrt{\rho}}{4L\sqrt{2}}. \tag{15}$$

Define the variables $V_1, U_1, V_2, U_2, \ldots$ and the constants

α, β as follows:

$$V_t = \frac{L^2}{M}\sum_{i=1}^{M} \|x_t^i - \overline{x}_t\|^2, \tag{16a}$$

$$U_t = \frac{32\,\eta^2\,L^2}{M(1-\sqrt{\rho})}\sum_{i=1}^{M} \|v_t^i - \nabla f(x_t^i)\|^2, \tag{16b}$$

$$\alpha = \frac{\left(3+\sqrt{\rho}\right)^2}{16}, \tag{16c}$$

$$\beta = \eta\frac{8L}{1-\sqrt{\rho}}\sigma_v^2. \tag{16d}$$

Then for all $t \geq 1$ it holds that $V_{t+1} \leq \alpha V_t + U_t$ and $\mathbb{E}[U_t \mid \mathcal{F}_{t-1}] \leq \beta$.

We can now move to the main result on decentralized SGD. The result gives conditions that guarantee the expected time $\mathbb{E}[\tau(\epsilon)]$ is finite, and also bounds this time in terms of the problem data, including the epoch length, variance, and the mixing rate of the connectivity matrix.

Proposition 16. *Let Assumptions 1, 2, 4, 14, and 13 hold, and assume that the initial parameters at every node are equal: $x_1^i = x_1^j$ for all $1 \leq i, j \leq M$. Suppose that $\epsilon > 2G^2 d_1(\mu_V, \mu_T)^2$. Let $c \leq (1-\sqrt{\rho})/(4\sqrt{2})$, and let the step-size be*

$$\eta = \frac{c}{L}\min\left\{1, \frac{\epsilon/2 - G^2 d_1(\mu_V,\mu_T)^2}{2m\sigma_v^2(1+128/(7+5\rho+\rho^{3/2}-13\sqrt{\rho}))}\right\}.$$

If $\tau(\epsilon)$ is the stopping time

$$\tau(\epsilon) = \inf\{n \geq 1 \,|\, n \equiv 1 \,(\mathrm{mod}\, m) \textit{ and } \|\nabla f_V(\overline{x}_n)\|^2 \leq \epsilon\}.$$

then

$$\mathbb{E}[\tau(\epsilon)] = \mathcal{O}\left(\frac{m^2}{(1-c)\,c\,(\epsilon - 2G^2 d_1(\mu_V,\mu_T)^2)^2(1-\sqrt{\rho})^2}\right).$$

Note that in the above result, the order of the convergence is the same as for regular SGD.

Using these tools allows us to bound the expected number of IFO calls needed by DSGD to find approximate stationary points.

Corollary 17. *Let Assumptions 1, 2, 3 and 4 hold. Suppose each gradient estimate is obtained by selecting an element $y_t^j \in Y_T$ uniformly at random and setting $v_t^j = \nabla_x f(y_t^j, x_t^j)$. Let $\epsilon > 2G^2 d_1(\mu_V, \mu_T)^2$ and consider running DSGD with epoch-length $m \geq 1$, and step-size η as defined in Proposition 16 with $c = (1-\sqrt{\rho})/(4\sqrt{2})$. Then the expected number of IFO calls used by DSGD with early stopping is*

$$\mathbb{E}\left[\mathrm{IFO}(\epsilon)\right] = \mathcal{O}\left(\frac{m(n_V + mM)}{(1-\sqrt{\rho})^3\sqrt{\rho}\,(\epsilon - 2G^2 d_1(\mu_V,\mu_T)^2)^2} + n_V\right).$$

Algorithm 3 SVRG with early stopping

```
 1: input: Initial point $x_m^1 \in \mathbb{R}^d$
 2: for $s = 1, 2, \dots$ do
 3:     $x_0^{s+1} = x_m^s$
 4:     $g^{s+1} = \frac{1}{n_T}\sum_{y \in Y_T} \nabla_x f(y, x_0^{s+1})$
 5:     if $\|g^{s+1}\|^2 \leq \epsilon$ then return $x_0^{s+1}$
 6:     for $t = 0$ to $m-1$ do
 7:         Sample $y_t^s$ uniformly at random from $Y_T$
 8:         $v_t^s = \nabla_x f(y_t^s, x_t^{s+1}) - \nabla_x f(y_t^s, x_0^{s+1}) + g^{s+1}$
 9:         $x_{t+1}^{s+1} = x_t^{s+1} - \eta v_t^s$
10:     end
11: end
```

Note the factor of M that appears in the numerator. This is due to the fact that M gradients are evaluated at each iteration of the algorithm, one at each node.

5 SVRG

In this section we analyze a variant of SVRG (Johnson & Zhang, 2013) with early stopping. The steps of the procedure are shown in Algorithm 3. Each epoch begins with a full gradient computation (Line 4). Next, the norm of the gradient is computed, and if it falls below the threshold ϵ, the algorithm terminates, returning the current iterate. Otherwise, an inner loop runs for m steps. The first step of the inner loop is to choose a random data point (Line 7). Then, the update direction is computed (Line 8) and used to obtain the next parameter (Line 9).

The technical tools we use to analyze SVRG with early stopping include existing bounds for SVRG (Reddi et al., 2016) along with the optional stopping theorem. Together, they yield the following bound on the expected number of epochs until SVRG with early stopping terminates.

Proposition 18. *Let Assumptions 1 and 2 hold and consider the variables x_t^{s+1} defined by Algorithm 3. Suppose that the step-size is set to $\eta = 1/(4Ln_T^{2/3})$ and the epoch length is $m = \lfloor 4n_T/3 \rfloor$. For $\epsilon > 0$, define $\tau(\epsilon)$ to be the stopping time $\tau(\epsilon) = \inf\left\{s \geq 1 \,\middle|\, \|\nabla f_T(x_0^{s+1})\|^2 \leq \epsilon\right\}$. Then*

$$\mathbb{E}[\tau(\epsilon)] \leq 1 + \frac{40Ln_T^{2/3}(f_T(x_m^1) - f^*)}{\epsilon}.$$

Note that Proposition 18 counts the number of epochs until an approximate stationary point is generated. A bound on the number of IFO calls can be obtained by multiplying τ by the number of IFO calls per epoch, which is $n_T + 2m$. This immediately leads to the following result:

Corollary 19. *Let Assumptions 1 and 2 hold and suppose the step-size η and epoch length m are defined as in Proposition 18. Then, the expected number of IFO calls until SVRG returns an approximate stationary point is* $\mathbb{E}\left[\mathrm{IFO}\left(\epsilon\right)\right] = \mathcal{O}((n_T^{5/3}/\epsilon) + n_T)$.

This result may be compared with Corollary 4 of (Reddi et al., 2016), which concerns an upper bound on the IFO calls needed for the expected (squared) norm of the gradient at a randomly selected iterate to be less than ϵ. Our result concerns the expected number of IFO calls before the algorithm terminates with an iterate that is guaranteed to be an approximate stationary point. Note that introducing early stopping does not add any complexity, compared to SGD. This is because the full gradient is already calculated at each iteration, and the only additional step in the algorithm is computation of the norm.

6 GENERALIZATION PROPERTIES

Typically, the training and validation sets are made of independent samples from a test distribution μ, and it is of interest to estimate the model performance relative to this test distribution. Formally, define the generalization error f_G as $f_G : \mathbb{R}^d \to \mathbb{R}$ as $f_G(x) = \mathbb{E}_{y\sim\mu}[f(y,x)]$. In this section, we consider upper bounds on the quantity

$$\mathbb{E}\left[\left\|\nabla f_G(x_{\tau(\epsilon)})\right\|^2\right], \qquad (17)$$

where $x_{\tau(\epsilon)}$ is the iterate returned by an optimization algorithm with early stopping. Note that this expectation is over both the variates generated by optimization and the random choice of the datasets Y_V and Y_T. In this section we show how Wasserstein concentration results can be used to bound (17), in terms of both the norm of the gradient of the training function, and the Wasserstein distance between μ and its empirical version used for optimization.

To begin, note that under Assumption 4, the gradient of the generalization error can be related to the gradient of the training error by

$$\mathbb{E}[\|\nabla f_G(x_{\tau(\epsilon)})\|] \leq \mathbb{E}[\|\nabla f_T(x_{\tau(\epsilon)})\|] + G\mathbb{E}[d_1(\mu_T, \mu)]$$

The second term on the right is the expected distance between the empirical measure μ_T and the data distribution μ. Intuitively, for large values of n_T the empirical distribution should be a good approximation to the true distribution, and the distance should be small. Investigations into the convergence rate of $d_p(\mu, \mu_T)$ as a function of n_T has received significant attention, beginning with (Dudley, 1969). For more background we refer the reader to (Dereich et al., 2013),(Weed & Bach, 2017) and references therein. For our purposes, the basic idea can be illustrated with the following result.

Theorem 20 ((Dereich et al., 2013), Theorem 1)**.** *For $d \geq 3$, let μ be a measure on $\mathbb{R}^d$, such that $J = \mathbb{E}_{y\sim\mu}\left[\|y\|^3\right]^{1/3} < \infty$, and let μ_N be an empirical version of μ constructed from N samples. Then there is a constant κ_d such that*

$$\mathbb{E}\left[d_2\left(\mu, \mu_N\right)^2\right] \leq \kappa_d J N^{-3/d}.$$

The constant κ_d is explicitly given in ((Dereich et al., 2013), Theorem 3). Note the dependence on the dimension d on the right hand side of this bound, which implies a very slow convergence of the empirical distance in high dimensions. Despite this, the bound is asymptotically tight, for large values of N. An example of a distribution that displays convergence of order $N^{-1/d}$ is the uniform distribution on $[0,1)^d$ (for a proof see Theorem 2 in (Dereich et al., 2013)). In a machine learning context, this would correspond to a regression problem where there is no relation between the input and output. We note however, that stronger rates of convergence can be obtained for restricted classes of measures, and that for smaller values of N the convergence rate can be more favorable. This is explored in (Weed & Bach, 2017) where they improve the bounds for a number of classes of distributions. For instance, when μ is a discrete distribution, the following holds:

Theorem 21 ((Weed & Bach, 2017), Proposition 13)**.** *Let μ be a measure that is supported on at most m points within the unit sphere in $\mathbb{R}^d$, and let μ_N be an empirical version of μ constructed from N samples. Then*

$$\mathbb{E}\left[d_2\left(\mu, \mu_N\right)^2\right] \leq 84\sqrt{\frac{m}{N}}.$$

Depending on the properties of the testing distribution, either one of Theorems 20 or 21 can be used to investigate the dependence of the generalization error on the data set size n_T. This would involve having some prior knowledge about the nature of the testing set.

In the remainder of this section, we consider combining the concentration bounds with the optimization bounds proved for SVRG. Note that the basic ideas can be applied just as well to SGD or DSGD.

For SVRG, it is natural to express the bound in terms of the number of training examples, and we obtain the following

Corollary 22. *Let Assumption 4 and the conditions of Proposition 18 hold. Further assume $J = \mathbb{E}_{y\sim\mu}\left[\|y\|^3\right]^{1/3} < \infty$ and the training set Y_T is an empirical version of μ. If $x_\tau(\epsilon)$ is the output of Algorithm 3, then*

$$\mathbb{E}[\|\nabla f_G(x_{\tau(\epsilon)})\|^2] \leq 2\epsilon + 2G^2\kappa_d J n_T^{-3/d}.$$

Alternatively, if μ is a supported on at most m points, then

$$\mathbb{E}[\|\nabla f_G(x_{\tau(\epsilon)})\|^2] \leq 2\epsilon + 168G^2\sqrt{\frac{m}{n_T}}.$$

Together with bounds on the expected running time, this result could potentially let one balance between the resources needed to minimize the training function, and the resources needed to gather training data. In order to minimize the right hand side, one can either choose a smaller ϵ, leading to longer running times, or choose a large n_T, leading to more sampling.

Note that Corollary 22 is accounts for data distribution properties (via the 3^{rd} moment J, or via the number of points in the discrete case) and does not depend on the number of iterations used in SGD. This result could be compared with (Hardt et al., 2016), where the authors proved a bound on the generalization gap for function values in terms of the number of iterations T and the number samples in the training set. There, the bound is increasing with T. An interesting avenue for future work would be to investigate the combination of the two approaches.

7 DISCUSSION

This work presented an analysis of several stochastic gradient-based optimization algorithms that use early stopping. Our focus was on procedures that return the first point satisfying a stopping criterion, and we analyzed the expected running time and number of gradient evaluations needed to meet this criterion.

For SGD, we analyzed the use of early stopping with a validation function, and obtained a bound on the expected number of gradient evaluations needed to find approximate stationary points. The analysis allows for biases in the update direction, subject to a geometric drift condition on the error terms. We specialized this analysis to bound the expected running time of decentralized SGD, a distributed variant of SGD. We modeled DSGD as a biased form of SGD, with a bias term that is controlled in part by the mixing coefficient of the communication graph. Next, we turned to a variant of nonconvex SVRG that employs early stopping, obtaining a bound on the expected number of IFO calls and gradient evaluations used by the algorithm. Lastly, we considered how Wasserstein concentration bounds can be leveraged to bound the generalization performance of the iterate returned by the algorithms, expressed in terms of the number of samples used to define the input datasets, and properties of the data distribution.

We would like to highlight two avenues for future work. Our analysis of SGD has a condition on the step-size that depends on the epoch length m (Corollary 11). It is an interesting question whether this requirement can be removed. Secondly, in our analysis of SVRG, introducing early stopping let to a convergence bound that is essentially the same as the rate obtained using randomization. For SGD, the expected number of IFO calls increases quadratically with the epoch length, and we leave it as an open question whether this is feature can also be relaxed.

ACKNOWLEDGEMENTS

This material is based upon work supported by the U.S. Department of Energy, Office of Science, under contract number DE-0012704.

REFERENCES

Abraham, R., Marsden, J. E., and Ratiu, T. *Manifolds, tensor analysis, and applications*, volume 75. Springer Science & Business Media, 2012.

Agarwal, A. and Bottou, L. A lower bound for the optimization of finite sums. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 78–86, 2015.

Allen-Zhu, Z. and Hazan, E. Variance reduction for faster non-convex optimization. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 699–707, 2016.

Bertsekas, D. P. and Tsitsiklis, J. N. Gradient convergence in gradient methods with errors. *SIAM Journal on Optimization*, 10(3):627–642, 2000.

Blanchet, J., Cartis, C., Menickelly, M., and Scheinberg, K. Convergence Rate Analysis of a Stochastic Trust Region Method via Submartingales. *ArXiv e-prints*, September 2016.

Bogachev, V. I. *Measure theory*, volume 1. Springer Science & Business Media, 2007.

Dai, A. M. and Le, Q. V. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems 28*, pp. 3079–3087. 2015.

Defazio, A., Bach, F., and Lacoste-Julien, S. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems 27*, pp. 1646–1654. 2014.

Dereich, S., Scheutzow, M., and Schottstedt, R. Constructive quantization: Approximation by empirical measures. *Ann. Inst. H. Poincar Probab. Statist.*, 49(4):1183–1203, 11 2013.

Dudley, R. M. The speed of mean glivenko-cantelli convergence. *Ann. Math. Statist.*, 40(1):40–50, 02 1969.

Duvenaud, D., Maclaurin, D., and Adams, R. Early stopping as nonparametric variational inference. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51, pp. 1070–1077, 2016.

Franceschi, L., Niepert, M., Pontil, M., and He, X. Learning discrete structures for graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 1972–1982, Long Beach, California, USA, 09–15 Jun 2019.

Ghadimi, S. and Lan, G. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Ghadimi, S. and Lan, G. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, pp. 1225–1234, 2016.

Jaderberg, M., Czarnecki, W. M., Osindero, S., Vinyals, O., Graves, A., Silver, D., and Kavukcuoglu, K. Decoupled neural interfaces using synthetic gradients. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1627–1635, 06–11 Aug 2017.

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pp. 315–323. 2013.

Kushner, H. and Clark, D. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Number v. 26 in Applied Mathematical Sciences. Springer-Verlag, 1978.

L. Roux, N., Schmidt, M., and Bach, F. R. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems 25*, pp. 2663–2671. 2012.

Lee, L., Parisotto, E., Chaplot, D. S., Xing, E., and Salakhutdinov, R. Gated path planning networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 2947–2955, Stockholmsmssan, Stockholm Sweden, 10–15 Jul 2018.

Lei, L., Ju, C., Chen, J., and Jordan, M. I. Non-convex finite-sum optimization via scsg methods. In *Advances in Neural Information Processing Systems 30*, pp. 2348–2358. 2017.

Lian, X., Huang, Y., Li, Y., and Liu, J. Asynchronous parallel stochastic gradient for nonconvex optimization. In *Advances in Neural Information Processing Systems 28*, pp. 2737–2745. 2015.

Lian, X., Zhang, C., Zhang, H., Hsieh, C.-J., Zhang, W., and Liu, J. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems 30*, pp. 5330–5340. 2017.

Lin, J. and Rosasco, L. Optimal learning for multi-pass stochastic gradient methods. In *Advances in Neural Information Processing Systems*, pp. 4556–4564, 2016.

Ljung, L. Analysis of recursive stochastic algorithms. *IEEE transactions on automatic control*, 22(4):551–575, 1977.

Nesterov, Y. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Paquette, C. and Scheinberg, K. A Stochastic Line Search Method with Convergence Rate Analysis. *arXiv e-prints*, art. arXiv:1807.07994, July 2018.

Reddi, S., Hefny, A., Sra, S., Poczos, B., and Smola, A. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, 2016.

Robbins, H. and Monro, S. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 09 1951.

Shamir, O. A stochastic pca and svd algorithm with an exponential convergence rate. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, pp. 144–152, 2015.

Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Wang, W. and Carreira-Perpinan, M. Nonlinear low-dimensional regression using auxiliary coordinates. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22, pp. 1295–1304, La Palma, Canary Islands, 21–23 Apr 2012.

Weed, J. and Bach, F. Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance. *ArXiv e-prints*, June 2017.

Williams, D. *Probability with Martingales*. Cambridge University Press, 1991.

Zhang, C., Jia, B., Gao, F., Zhu, Y., Lu, H., and Zhu, S.-C. Learning perceptual inference by contrasting. In *Advances in Neural Information Processing Systems 32*, pp. 1073–1085. 2019.

Zhang, H., J. Reddi, S., and Sra, S. Riemannian svrg: Fast stochastic optimization on riemannian manifolds. In *Advances in Neural Information Processing Systems 29*, pp. 4592–4600. 2016.