Multi-Level Stochastic Gradient Methods for Nested Composition Optimization

Shuoguang Yang* Mengdi Wang† Ethan X. Fang‡

Abstract

Stochastic gradient methods are scalable for solving large-scale optimization problems that involve empirical expectations of loss functions. Existing results mainly apply to optimization problems where the objectives are one- or two-level expectations. In this paper, we consider the multi-level compositional optimization problem that involves compositions of multi-level component functions and nested expectations over a random path. It finds applications in risk-averse optimization and sequential planning. We propose a class of multi-level stochastic gradient methods that are motivated from the method of multi-timescale stochastic approximation. First we propose a basic T-level stochastic compositional gradient algorithm. Then we develop accelerated multi-level stochastic gradient methods by using an extrapolation-interpolation scheme to take advantage of the smoothness of individual component functions. When all component functions are smooth, we show that the convergence rate improves to $\mathcal{O}(n^{-4/(7+T)})$ for general objectives and $\mathcal{O}(n^{-4/(3+T)})$ for strongly convex objectives. We also provide almost sure convergence and rate of convergence results for nonconvex problems. The proposed methods and theoretical results are validated using numerical experiments.

Keywords: Stochastic gradient · Stochastic optimization · Convex Optimization · Sample complexity · Simulation · Statistical learning

1 Introduction

Over the past decade, stochastic gradient-type methods have drawn significant attention from various communities such as mathematical programming, signal processing and machine learning, mainly due to their practical efficiency in minimizing expected-value objective functions or empirical sums of a large number of loss functions [2, 5, 11, 12, 13, 15, 19, 23, 30]. They are particularly

*Department of Industrial Engineering and Operations Research, Columbia University, New York City, NY, USA; e-mail: sy2614@columbia.edu

†Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ, USA; email: mengdiw@princeton.edu

‡Department of Statistics and Department of Industrial and Manufacturing Engineering, Pennsylvania State University, University Park, PA, USA; email: xxf13@psu.edu

popular methods for tackling large-scale problems such as statistical estimation [7, 21], matrix and tensor factorization [9] and training deep neural networks [14, 29]. Stochastic gradient methods mainly apply to minimizing the expectation of a stochastic function, i.e.,

$$\min_x \mathbb{E}_\omega[f_\omega(x)],$$

where the expectation is taken over a random variable ω. Note that this problem involves one level of expectation.

In this paper, we propose to study a richer class of stochastic optimization problems, which involve nested expectations over a sequence of random variables. In particular, we consider the *T-level stochastic compositional optimization problem*, given by

$$\min_{x\in\mathcal{X}} F(x) = \mathbb{E}_{\omega_1}\Big[f_{\omega_1}^{(1)}\Big(\mathbb{E}_{\omega_2}\big[f_{\omega_2}^{(2)}\big(\cdots\big(\mathbb{E}_{\omega_T}[f_{\omega_T}^{(T)}(x)]\big)\cdots\big)\big]\Big)\Big], \tag{1.1}$$

where $f_{\omega_j}^{(j)}(\cdot):\mathbb{R}^{d_j}\mapsto\mathbb{R}^{d_{j-1}}$ for $j=1,\cdots,T$ are continuous mappings, $\mathcal{X}$ is a convex and closed set, and $d_0=1$, i.e., $F(x)$ is a real-valued function. The nested composition structure provides a rich modeling tool for data analysis and decision-making applications. For instance, online principal component analysis and policy evaluation in reinforcement learning can be formulated into two-level stochastic compositional optimization [16, 27]. We illustrate an example that arises from operations research in Section 4, the mean-deviation risk-averse optimization problem. It can be formulated into a three-level compositional problem [1, 22].

In problem (1.1), for each $f_{\omega_j}^{(j)}$, we use the subscript ω_j to denote a random variable and use the superscript (j) to denote its level. We focus on situations whether there exist deterministic functions $f^{(1)},\ldots,f^{(T)}$ such that

$$f^{(j)}(x_j) = \mathbb{E}[f_{\omega_j}^{(j)}(x_j)|\omega_1,\cdots,\omega_{j-1}],$$

for all $j=1,\ldots,T$ with probability 1. We refer to $f^{(1)},\ldots,f^{(T)}$ as *component functions*. However, these component functions are not explicitly known to us. Note that the multi-level random variables $\omega_1,\ldots,\omega_T$ are not necessarily independent of one another. When we sample from their joint distribution, we may generate a sample path $(\omega_1,\ldots,\omega_T)$ sequentially by sampling each ω_j conditioned on realizations at the previous level's $(\omega_1,\ldots,\omega_{j-1})$. Throughout this paper, we assume that the component functions $f^{(1)},\ldots,f^{(T)}$ are continuous and that there exists at least one optimal solution x^* to problem (1.1). In some part of our analysis, we require the overall objective function $F(x)$ be convex, but we *never* require that any individual component function $f_{\omega_j}^{(j)}(\cdot)$ be convex, linear or monotone. We say that a function f is "smooth" if it has Lipschitz continuous gradients, and say that it is "non-smooth" otherwise.

Our goal is to solve the T-level stochastic compositional optimization problem (1.1) by sampling multiple paths of $(\omega_1,\ldots,\omega_T)$. We are interested in scenarios where we do not have the explicit knowledge of the expected-value component functions $f^{(j)}$'s. This often occurs when evaluating $f^{(j)}$ requires making expensive passes over large data sets. This also occurs in online learning applications where $f^{(j)}$ can not be accurately calculated using finitely many samples. Instead of knowing $f^{(j)}$'s, we suppose that there is a Sample Oracle ($\mathcal{SO}$) such that:

- Upon each query $(x\in\mathcal{X}, y_1\in\mathbb{R}^{d_1},\ldots,y_T\in\mathbb{R}^{d_T})$, the $\mathcal{SO}$ generates a sample path $(\omega_1,\ldots,\omega_T)$ independently from the query.

- The $\mathcal{SO}$ returns a vector $f_{\omega_T}^{(T)}(x) \in \mathbb{R}^{d_{T-1}}$ and a gradient/subgradient $\widetilde{\nabla} f_{\omega_T}^{(T)}(x) \in \mathbb{R}^{d_T \times d_{T-1}}$.
- The $\mathcal{SO}$ returns a vector $f_{\omega_j}^{(j)}(y_j) \in \mathbb{R}^{d_j}$ and a gradient $\nabla f_{\omega_j}^{(j)}(y_j) \in \mathbb{R}^{d_j \times d_{j-1}}$.
- The $\mathcal{SO}$ returns a gradient $\nabla f_{\omega_1}^{(1)}(y_1) \in \mathbb{R}^{d_1}$.

In the above, we denote by $\widetilde{\nabla} f_{\omega_T}^{(T)}(x)$ a gradient/subgradient, which is to be specified in the context. Let us emphasize that this $\mathcal{SO}$ does *not* return unbiased first-order information regarding the overall objective function. The $\mathcal{SO}$ can be viewed as a *component-wise stochastic first-order oracle* that returns noisy first-order information for individual component functions $f^{(j)}$'s. Detailed assumptions on the $\mathcal{SO}$ will be specified later.

One might attempt to apply the sample average approximation (SAA) method to attack the multi-level expectation problem (1.1). However, replacing the nested expectations with empirical averages will not solve the optimization problem. It will reduce one problem with expectations to another one with empirical expectations. However, the two problems share similar structures and the latter one is not necessarily easier to solve. What we need is an implementable algorithm that computes the optimal solution by iteratively querying the $\mathcal{SO}$ and making efficient updates.

Another attempt would be to use some version of gradient method or stochastic gradient method. Stochastic gradient method will not work automatically. The main challenge is that we do not have access to the unbiased sample gradient of F due to the multi-level nested expectations. To see this, let us consider the case where each $f^{(j)}$ is differentiable and apply the chain rule to get

$$\nabla F(x) = \nabla f^{(T)}(x) \nabla f^{(T-1)}\big(f^T(x)\big) \cdots \nabla f^{(1)}\left(f^{(2)} \circ \cdots \circ f^{(T)}(x)\right).$$

For a given $x \in \mathcal{S}$ and a given sample path $(\omega_1, \ldots, \omega_T)$, one may formulate an unbiased estimate of $\nabla F(x)$ as

$$\nabla f_{\omega_T}^{(T)}(x) \nabla f_{\omega_{T-1}}^{(T-1)}\big(f^{(T)}(x)\big) \cdots \nabla f_{\omega_1}^{(1)}\left(f^{(2)} \circ \cdots \circ f^{(T)}(x)\right),$$

which unfortunately cannot be calculated by calling the $\mathcal{SO}$ once (or even finitely many times). This is because that computing the preceding unbiased gradient sample requires querying the $\mathcal{SO}$ at values $f^{(T)}(x), f^{(T-1)} \circ f^{(T)}(x), \ldots, f^{(2)} \circ \cdots \circ f^{(T)}(x)$, which are unfortunately not known. As a result, the nested composition structure induces substantial bias in the sample gradients for F as long as $T \geq 2$. In contrast, when $T = 1$, the objective function is linear in the distribution of the random variable ω. For problems with $T \geq 2$, the nonlinear composition between expectations and component functions creates an objective function that is highly nonlinear with respect to the joint probability distribution of $\omega_1, \ldots, \omega_T$. A graphical illustration of the level of difficulty for dealing with multi-level composition optimization is given in Figure 1. We can view the optimization problem (1.1) under the $\mathcal{SO}$ as a form of estimation problem, in which we want to estimate the optimal solution x^* by taking independent sample paths. We can see that the nonlinear composition makes this estimation/optimization problem fundamentally challenging.

Existing work on stochastic compositional optimization traces back to [8] which considered the two-level problem. In Section 6.7 of [8], a two-timescale stochastic approximation scheme was proposed and its almost sure convergence was established without rate analysis. Recently, [26] developed a general class of stochastic compositional gradient descent (SCGD) method for two-level problems and established convergence rate results under various assumptions. [28] developed an

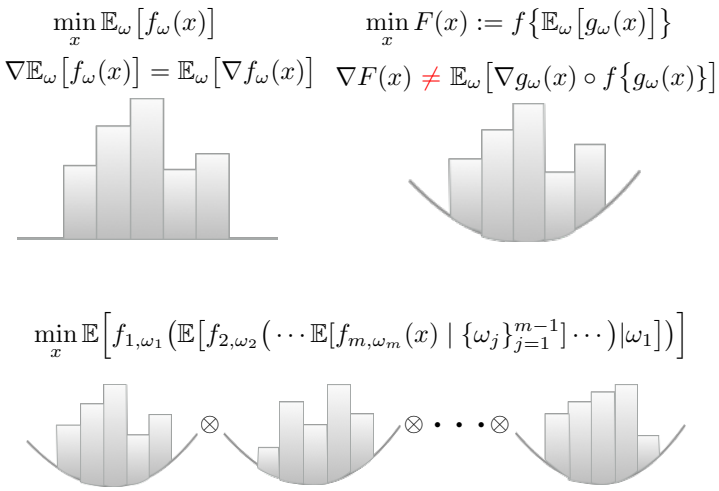

Figure 1: In one-level stochastic optimization, the objective function is linear in the probability distribution of ω. In multi-level stochastic compositional optimization, the objective is no longer linear in the joint probability distribution of the random variables $(\omega_1, \ldots, \omega_m)$, making the problem fundamentally harder.

accelerated stochastic compositional proximal gradient (ASC-PG) method for the two-level problem and proved faster convergence in some cases. [17] considered a special case of the two-level problem where each expectation takes the form of a finite sum of loss functions and developed variance-reduced versions of the compositional gradient methods. As for the general T-level problem, to the best of our knowledge, all existing results only apply to the case where $T = 1, 2$. Multi-level stochastic compositional optimization remains largely open.

In this paper, we develop sampling-based algorithms and complexity theory for the T-level stochastic compositional problem (1.1). We draw motivation from the optimality conditions of problem (1.1). In particular, we expand the first-order condition into a system of variational equalities and inequalities by introducing auxiliary variables that correspond to a sequence of *value functions* at the optimal solution, i.e., tail compositions of the component functions. Our first attempt is a basic multi-timescale stochastic approximation iteration to solve this system. We establish its almost sure convergence using a T-element super-martingale argument for both convex and non-convex problems. We also show that it converges to the optimal solution at a rate of $\mathcal{O}(n^{-1/2^T})$ where n is the number of iterations/oracle queries. This result suggests that the sample complexity for obtaining an approximate-optimal solution depends exponentially on the number of nested levels T. Such an exponential dependence is somewhat expected in the worst case. It is consistent with the sample path complexity for solving multi-stage stochastic programming, although the optimization formulations and assumptions are slightly different. An $\mathcal{O}(n^{-2/(T+1)})$ rate of convergence is also obtained for strongly convex objectives. These convergence rates are not yet satisfactory.

Furthermore, we develop accelerated multi-level stochastic gradient methods. The accelerated algorithms apply to "smooth" composition problems and takes advantages of the smoothness of individual component functions $f^{(j)}$. An *extrapolation-interpolation* scheme is used to balance the

		Non-convex	Convex	Strongly convex
1-level		$\mathcal{O}(n^{-1/2})$ [10]	$\mathcal{O}(n^{-1/2})$ [24]	$\mathcal{O}(n^{-1})$ [20]
2-level	Smooth	$\mathcal{O}(n^{-4/9})$ [28]	$\mathcal{O}(n^{-4/9})$ [28]	$\mathcal{O}(n^{-4/5})$ [28]
	Non-smooth	$\mathcal{O}(n^{-1/4})$ [26]	$\mathcal{O}(n^{-1/4})$ [26]	NA
3-level	Smooth	$\mathcal{O}(n^{-2/5})$ [*]	$\mathcal{O}(n^{-2/5})$ [*]	$\mathcal{O}(n^{-2/3})$ [*]
T-level	Smooth	$\mathcal{O}(n^{-4/(7+T)})$ [*]	$\mathcal{O}(n^{-4/(7+T)})$ [*]	$\mathcal{O}(n^{-4/(3+T)})$ [*]

Table 1: Best-known n-sample error bound for solving multi-level stochastic compositional optimization. These bounds are achieved by stochastic gradient-type methods, so they are n-iteration error bounds at the same time. Note that we say the composition problem is "smooth" if *all* the component functions have Lipschitz continuous gradients. We use [*] to denote the current paper.

bias-variance tradeoff in approximating each value function. We establish its almost sure convergence using a T-element super-martingale argument for both convex and nonconvex problems. The accelerated updates for the auxiliary variables can be viewed as first-order running approximations of the true values, while the basic method without acceleration uses zeroth-order running approximations. As a result, the accelerated updates are more accurate and thus the overall convergence rate is improved. In the case when all component functions are smooth, we improve the convergence rate to $\mathcal{O}(n^{-4/(7+T)})$ for convex objective functions and $\mathcal{O}(n^{-4/(3+T)})$ for strongly convex ones. We have also obtained convergence and rate of convergence results for nonconvex problems. Table 1 summarizes our results and compare them with the best known ones for the single- and two-level stochastic compositional optimization problems [10, 20, 24, 26, 28]. We also provide numerical experiments with a risk-averse regression problem. The numerical results validate our theory.

To the best of our knowledge, this paper proposes for the first time the multi-level stochastic gradient methods for the composition optimization problem (1.1), where we establish almost sure convergence results and obtain fast convergence rates. For the case where $T = 1$, our results match the best known sample complexity upper- and lower-bounds. For the case where $T = 2$, our results improve the convergence rate from $\mathcal{O}(n^{-2/9})$ of the a-SCGD in [26] to $\mathcal{O}(n^{-2/5})$. Besides, with additional assumption that the inner level function $f^{(T)}$ in (1.1) has Lipschitz continuous gradients, we obtain a convergence rate $\mathcal{O}(n^{-4/9})$ for two-level problems, which matches the state-of-art result achieved by ASC-PG in [28]. A natural further question is how big are the hidden constants in these error bounds. In the case where $T = 1$, the hidden constant is merely determined by the variance of stochastic gradients and the condition number, which can be derived in straightforward way by analyzing a telescoping sum [10, 20]. However, when $T > 1$, the hidden constants depend on a tedious formula involving sums and products of multi-level variances and Lipschitz continuity constants. Within the scope of the current paper, we focus the dominating order of the error bounds, leaving the constants unspecified. For the cases where $T \geq 3$, our results fill the open gaps and provide the first few sample complexity benchmarks.

The proposed methods of the paper, while being optimization algorithms, can be viewed as updating an online estimator by drawing samples from a data stream. Let us evaluate its performance from statistical perspectives. For comparison, the most related result is given by [6], which uses an sample average approximation approach to solve the T-level compositional problem where

the multi-level random variables are independently identically distribution random variables. For this case, [6] proved that the batch method achieves an error rate of $\mathcal{O}(1/\sqrt{n})$ which is obviously statistically nonimprovable. More remarkably, the error bound obtained in [6] is independent of T. In this paper, our result for the smooth convex case is $\mathcal{O}(n^{-4/(7+T)})$, which deteriorates as the number of levels T increases. There are two possible explanations. First, the problem considered in this paper is slightly more general than that of [6] because we do not assume independence between random variables at different levels. Second, the proposed algorithms use multi-timescale updates so that certain random samples are given less weights than the others, while the batch approach treats all samples equally. The use of multi-timescale updates, which is critical for the proposed online method, may have resulted in inefficient use of data and slowed down the convergence. It remains open whether there exists an online algorithm that can achieve the same rate of convergence as the batch method. We hope the developments of this paper can pave the way to more complete understanding of the complexity of multi-level composition optimization.

Paper Organization. Section 2 gives a basic algorithm based on multi-timescale stochastic approximation and establishes its convergence. Section 3 develops accelerated versions of the algorithm and shows that they achieve faster convergence for smooth problems. Section 4 illustrates one motivating application in operations research and gives numerical experiments.

Notation and Definitions. For $x \in \mathbb{R}^n$, we denote by x' its transpose, and by $\|x\|$ its Euclidean norm (i.e.,$\|x\| = \sqrt{x'x}$). For two sequences $\{x_k\}$ and $\{y_k\}$, we write $x_k = \mathcal{O}(y_k)$ if there exists a constant $c > 0$ such that $\|x_k\| \leq c\|y_k\|$ for each k. We denote by $\mathbb{I}_{condition}^{value}$ the indicator function, which returns " value " if the " condition " is satisfied; otherwise 0. We denote by F^* the optimal objective function value for (1.1), and denote by $\mathcal{X}^*$ the set of optimal solutions. For a set $\mathcal{X} \subset \mathbb{R}^n$ and a vector $y \in \mathbb{R}^n$, we denote by $\Pi_{\mathcal{X}}\{y\} = \operatorname{argmin}_{x\in\mathcal{X}} \|y - x\|^2$ the Euclidean projection of y on $\mathcal{X}$, where the minimization is always uniquely attained if $\mathcal{X}$ is nonempty, convex and closed. For a function $f(x)$, we denote by $\nabla f(x)$ its gradient at x if f is differentiable, denote by $\partial f(x)$ its subdifferential at x, and denote by $\widetilde{\nabla} f(x)$ some noisy estimate of the gradient/subgradient of f at x. We denote by "w.p.1" as "with probability 1".

2 A Basic Algorithm Based On Multi-Timescale Stochastic Approximation

We start by writing down the optimality condition of problem (1.1) (assuming that the problem is convex):

$$\nabla F(x^*)'(x - x^*) \geq 0, \qquad \forall x \in \mathcal{X},$$

where

$$\nabla F(x) = \nabla f^{(T)}(x) \cdot \nabla f^{(T-1)}\left(f^T(x)\right) \cdots \nabla f^{(1)}\left(f^{(2)} \circ \cdots \circ f^{(T)}(x)\right).$$

However, this optimality condition is not easy to work with. As we have discussed in Section 1, the chain rule makes obtaining unbiased samples of $\nabla F(x)$ difficult. Let us rewrite the the

optimality condition as follows

$$\begin{aligned}
&\left(\nabla f^{(T)}(x)\nabla f^{(T-1)}(y^{T-1}))\cdots\nabla f^{(1)}\left(y^{(1)}\right)\right)'(x-x^*)\geq 0, \qquad \forall x\in\mathcal{X},\\
&y^{(T-1)}=f^{(T)}(x)\\
&y^{(T-2)}=f^{(T-1)}(y^{(T-1)})=f^{(T-1)}\circ f^{(T)}(x)\\
&y^{(1)}=f^{(2)}(y^{(2)})=f^{(2)}\circ\cdots\circ f^{(T)}(x).
\end{aligned}$$

We refer to $f^{(j)}\circ\cdots\circ f^{(T)}(x)$, $j=1,\ldots,T-1$ as the *value functions*, i.e., tail compositions of multi-level component functions. By introducing the auxiliary variables $y^{(j)}$'s to represent the value functions, we can decouple the chain product. Now for a given $(x,y^{(1)},\ldots,y^{(T-1)})$, our sampling oracle allows us to get unbiased estimates for all the quantities in the preceding system of optimality conditions.

2.1 A T-Level Stochastic Gradient Method

Motivated by the system of optimality conditions, we develop our first algorithm - a multi-timescale approximation iteration. It is also a generalization of the basic-SCGD in [26] which applies only to two-level problems. Our algorithm runs iteratively. Denote by k the iteration counter. A key ingredient of our algorithm is to introduce auxiliary variables $y_k^{(j)}$'s, defined recursively, as running estimates for the value functions $\mathbb{E}_{\omega_{j,k}}[f^{(j)}_{\omega_{j,k}}(y_k^{(j+1)})|\omega_{1,k},\cdots,\omega_{j-1,k}]$, where $j=1,\cdots,T-1$, and $x_k=y_k^{(T)}$. At the k-th iteration, we update the current solution x_k by using a quasi-stochastic gradient step given by

$$x_{k+1}=\Pi_{\mathcal{X}}\Big\{x_k-\alpha_k\widetilde{\nabla}f^{(T)}_{\omega_{T,k}}(x_k)\nabla f^{(T-1)}_{\omega_{T-1,k}}(y_k^{(T-1)})\cdots\nabla f^{(1)}_{\omega_{1,k}}(y_k^{(1)})\Big\}.$$

Then, we update the auxiliary variables $y_k^{(j)}$'s by taking a weighted average between the previous values and the new samples returned by the $\mathcal{SO}$, i.e., for $j=T-1,T-2,\ldots,1$,

$$y_{k+1}^{(j)}=(1-\beta_{j,k})y_k^{(j)}+\beta_{j,k}f^{(j+1)}_{\omega_{j+1,k+1}}(y_{k+1}^{(j+1)}), \tag{2.1}$$

where $\omega_{j,k}$ denotes the realization of j-th level random variable at the k-th iteration, $\beta_{j,k}$'s are pre-specified stepsizes. We refer to this update for $y_k^{(j)}$ as a *basic update step*. Letting $y_k^{(T)}=x_k$ and $\alpha_k=\beta_{T,k}$ to simplify the notation, we refer to the preceding iteration as the basic T-level Stochastic Compositional Gradient Descent (T-SCGD) method and summarize it in Algorithm 1. Note that we choose the stepsizes such that $\beta_{j+1,k}/\beta_{j,k}\to 0$ as $k\to\infty$ for all j's, in order to control and balance the convergence speed for each auxiliary variables.

To analyze the convergence of the algorithm, we impose the following assumptions on the smoothness and bounded second-order moments for the stochastic component functions.

Assumption 2.1. Let $C_1,C_2,\cdots,C_T,V_1,\cdots,V_T,L_2,L_3,\cdots,L_T$ be positive scalars.

(i) The outer functions $f^{(T-1)},f^{(T-2)},\cdots,f^{(1)}$ are continuously differentiable, the inner function $f^{(T)}$ is continuous, the feasible set $\mathcal{X}$ is closed and convex, and there exists at least one optimal solution x^* to problem (1.1).

Algorithm 1 Basic Stochastic Compositional Gradient Descent (T-SCGD)

Input : $x_0 \in \mathbb{R}^{d_T}$, $y_0^{(j)} \in \mathbb{R}^{d_j}$, for $j = T-1, \cdots, 1$, $\mathcal{SO}$, K, stepsizes $\{\alpha_k\}_{k=0}^K$, $\{\beta_{j,k}\}_{k=0}^K$ for $j = T-1, ..., 1$.

Output : The sequence $\{x_k\}_{k=0}^K$.

for $k = 0, 1, 2, ..., K$ **do**

Query the $\mathcal{SO}$ for the sample values of $f^{(T)}, \cdots, f^{(1)}$ at $(x_k, y_k^{(T-1)}, \cdots, y_k^{(1)})$, obtain the sample gradients/ subgradients $\widetilde{\nabla} f_{\omega_{T,k}}^{(T)}(x_k), \nabla f_{\omega_{T-1,k}}^{(T-1)}(y_k^{(T-1)}), \cdots, \nabla f_{\omega_{1,k}}^{(1)}(y_k^{(1)})$.

Update the main iterate by

$$x_{k+1} = \Pi_{\mathcal{X}}\Big\{x_k - \alpha_k \widetilde{\nabla} f_{\omega_{T,k}}^{(T)}(x_k) \nabla f_{\omega_{T-1,k}}^{(T-1)}(y_k^{(T-1)}) \cdots \nabla f_{\omega_{1,k}}^{(1)}(y_k^{(1)})\Big\}.$$

Query the $\mathcal{SO}$ for the sample value of $f^{(T)}(\cdot)$ at x_{k+1}, obtain $f_{\omega_{T,k+1}}^{(T)}(x_{k+1})$.

Update $y_k^{(T-1)}$ by

$$y_{k+1}^{(T-1)} = (1 - \beta_{T-1,k}) y_k^{(T-1)} + \beta_{T-1,k} f_{\omega_{T,k+1}}^{(T)}(x_{k+1}).$$

for $j = T-2, \cdots, 1$ **do**

Query the $\mathcal{SO}$ for the sample value of $f^{(j)}$ at $y_{k+1}^{(j)}$, obtain $f_{\omega_{j,k+1}}^{(j)}(y_{k+1}^{(j)})$.

Update

$$y_{k+1}^{(j)} = (1 - \beta_{j,k}) y_k^{(j)} + \beta_{j,k} f_{\omega_{j+1,k+1}}^{(j+1)}(y_{k+1}^{(j+1)}).$$

end for

end for

(ii) The sample paths $(\omega_{1,0}, \omega_{2,0}, \cdots, \omega_{T,0})$, $(\omega_{1,1}, \omega_{2,1}, \cdots, \omega_{T,1})$,...,$(\omega_{1,k}, \omega_{2,k}, \cdots, \omega_{T,k})$ are independent across k and satisfy with probability 1

$$\mathbb{E}[f_{\omega_{j,0}}^{(j)}(x_j) | \omega_{1,0}, \cdots, \omega_{j-1,0}] = f^{(j)}(x_j), \forall x_j \in \mathbb{R}^{d_j} \text{ for } j = 1, \cdots, T, \text{ and } \mathbb{E}[\widetilde{\nabla} F_{\omega_0}(x)] \in \partial F(x),$$

for all $x \in \mathcal{X}$, where $\widetilde{\nabla} F_{\omega_0}(x) \equiv \widetilde{\nabla} f_{\omega_{T,0}}^{(T)}(x) \nabla f_{\omega_{T-1,0}}^{(T-1)}\big(f^{(T)}(x)\big) \cdots \nabla f_{\omega_{1,0}}^{(1)}\Big(f^{(2)} \circ \cdots \circ f^{(T)}(x)\Big)$.

(iii) The function $f^{(T)}(\cdot)$ is Lipschitz continuous with parameter C_T, and the samples $f_{\omega_{T,0}}^{(T)}(\cdot)$, $\widetilde{\nabla} f_{\omega_{T,0}}^{(T)}(\cdot)$ have bounded second-order moments such that with probability 1

$$\mathbb{E}\big[\|\widetilde{\nabla} f_{\omega_{T,0}}^{(T)}(x)\|^2 | \omega_{T-1,0}, \cdots, \omega_{1,0}\big] \le C_T, \mathbb{E}\big[\|f_{\omega_{T,0}}^{(T)}(x) - f^{(T)}(x)\|^2 | \omega_{T-1,0}, \cdots, \omega_{1,0}\big] \le V_T,$$

for all $x \in \mathcal{X}$.

(iv) For $j = 1, \cdots, T-1$, the functions $f^{(j)}(\cdot)$'s and $f_{\omega_{j,0}}^{(j)}(\cdot)'s$ have L_j-Lipschitz continuous gradients such that with probability 1

$$\mathbb{E}\big[\|\nabla f_{\omega_{j,0}}^{(j)}(x_j)\|^2 | \omega_{j-1,0}, \cdots, \omega_{1,0}\big] \le C_j, \mathbb{E}\big[\|f_{\omega_{j,0}}^{(j)}(x_j) - f^{(j)}(x_j)\|^2 | \omega_{j-1,0}, \cdots, \omega_{1,0}\big] \le V_j,$$

$$\text{and } \|\nabla f^{(j)}_{\omega_{j,0}}(x_j) - \nabla f^{(j)}_{\omega_{j,0}}(\bar{x}_j)\| \leq L_j \|x_j - \bar{x}_j\|,$$

for all $x_j, \bar{x}_j \in \mathbb{R}^{d_j}$.

In some part of the analysis, we also assume that the overall objective is sufficiently smooth as follows.

Assumption 2.2. The function $F(x)$ has Lipschitz continuous gradient, i.e., there exists $L_F > 0$ such that

$$F(z) - F(x) \leq \langle \nabla F(x), z - x \rangle + \frac{L_F}{2}\|z - x\|^2, \ \ \forall x, z.$$

Note that in Assumption 2.1, we require the functions $f^{(1)}(\cdot), \cdots, f^{(T-1)}(\cdot)$ to have Lipschitz continuous gradients, and we do not impose such assumptions on $f^{(T)}(\cdot)$. Hence, we cannot guarantee that $F(x)$ has a Lipschitz continuous gradient, which means Assumption 2.1 does not imply Assumption 2.2.

Although Assumptions 2.1-2.2 may seem complicated, they are actually quite mild. They essentially require that the component function be sufficiently smooth and the samples have bounded second moments. The conditions on smoothness can be easily satisfied when the component functions are polynomial functions. The conditions on second-moment boundedness can be satisfied when the random variables are drawn from a finite set or when the random variables have subgaussian distributions, which are typically satisfied in big data applications. Please see the numerical example used in Section 4 as an example.

2.2 Convergence Results for T-SCGD

Theoretical analysis of Algorithm 1 is challenging due to the nested level of expectations over a path of random variables. The multiple nested levels of expectations need to be carefully estimated and balanced to ensure convergence of the algorithm.

We first show the almost sure convergence of the algorithm as long as the step-sizes are properly chosen and diminishing under Assumption 2.1. For convex problems, we show that the algorithm generates a sequence of solutions that converges to an optimal solution to problem (1.1) with probability 1. For nonconvex problems with smooth objective, we show that all limiting points of the sequence generated by this algorithm are stationary points with probability 1 under mild assumptions.

Meanwhile, we analyze the convergence rate of Algorithm 1. Specifically, we derive the rate through taking the averaged iterates $\widehat{x}_n = \frac{1}{N_n}\sum_{k=n-N_n+1}^{n} x_k$, where $N_n = \lceil n/2 \rceil$. Note that similar results still hold if we let $N_n = n/C$, where $C > 1$ is a positive integer. Clearly, the rate of convergence is closely related to the stepsizes α_k's and $\beta_{j,k}$'s. We consider stepsizes of the form

$$\alpha_k = k^{-a} \text{ and } \beta_{j,k} = k^{-b_j} \text{ for all } j = T-1, ..., 1, \tag{2.2}$$

where a and b_j's are real numbers, and obtain the convergence rate by optimizing over a and b_j's.

Furthermore, we consider multi-level compositional problems with *optimally strongly convex* objective. Algorithm 1 achieves a much faster convergence rate for such problems. In particular,

denote by $\mathcal{X}^*$ the set of optimal solutions x^* to problem (1.1). We say that the objective function F is optimally strongly convex with parameter $\lambda > 0$ if

$$F(x) - F\big(\Pi_{\mathcal{X}^*}(x)\big) \geq \lambda \|x - \Pi_{\mathcal{X}^*}(x)\|^2, \quad \forall x \in \mathcal{X}. \tag{2.3}$$

Clearly, the class of optimally strongly convex functions strictly contains all strongly convex functions, and is thus more general.

Theorem 2.1 (Convergence of T-SCGD)**.** Let Assumption 2.1 hold, and let $\{(x_k, y_k^{(T-1)}, \cdots, y_k^{(1)})\}_{k=0}^{\infty}$ be the sequence generated by the T-SCGD Algorithm 1 starting with an arbitrary initial point $(x_0, y_0^{(T-1)}, \cdots, y_0^{(1)})$.

(a) Let the step-sizes $\{\alpha_{1,k}\}, \{\beta_{2,k}\}, \cdots, \{\beta_{T,k}\}$ be such that

$$\sum_{k=0}^{\infty} \alpha_k = \infty, \sum_{k=0}^{\infty} \beta_{j,k} = \infty, \text{ for all } j = T-1, ..., 1,$$

and

$$\sum_{k=0}^{\infty} \left(\alpha_k^2 + \beta_{T-1,k}^2 + \cdots + \beta_{1,k}^2 + \frac{\alpha_k^2}{\beta_{2,k}} + \frac{\alpha_k^2}{\beta_{3,k}} + \cdots + \frac{\alpha_k^2}{\beta_{T-1,k}} + \frac{\beta_{T-1,k}^2}{\beta_{T-2,k}} + \cdots + \frac{\beta_{2,k}^2}{\beta_{1,k}} \right) < \infty.$$

(i) If F is convex, $\{x_k\}$ converges almost surely to a random point in the set of optimal solutions to problem (1.1).

(ii) Suppose Assumption 2.2 holds in addition, $\mathcal{X} = \mathbb{R}^{d_T}$, and all samples generated by the $\mathcal{SO}$ are uniformly bounded, then any limiting point of the sequence $\{x_k\}_{k=0}^{\infty}$ is a stationary point to problem (1.1) almost surely.

(b) If F is convex, let $D_x > 0$ be such that $\sup_{x \in \mathcal{X}} \|x - x^*\| \leq D_x$ and set the stepsizes be $\alpha_k = k^{-a}$, $\beta_{j,k} = k^{-b_j}$ for $j = T-1, \cdots, 1$, where $(a, b_{T-1}, b_{T-2}, \cdots, b_1) \in (0,1)$, then we obtain

$$\mathbb{E}[F(\widehat{x}_n) - F^*] \leq \mathcal{O}(n^{-1/2^T}),$$

by choosing $a = 1 - \frac{1}{2^T}, b_{T-1} = 1 - \frac{1}{2^{T-1}}, \cdots, b_1 = 1 - \frac{1}{2}$.

(c) Suppose that Assumption 2.2 holds, and F is optimally strongly convex with some parameter $\lambda > 0$ defined in (2.3). Letting $\alpha_k = \frac{1}{\lambda} k^{-a}$, $\beta_{j,k} = k^{-b_j}$ for $j = T-1, \cdots, 1$, we obtain

$$\mathbb{E}[\|x_n - \Pi_{\mathcal{X}^*}(x_n)\|^2] \leq \mathcal{O}(n^{-2/(T+1)}),$$

by choosing $a = 1$ and $b_j = \frac{j+1}{T+1}$ for $j = T-1, T-2, \cdots, 1$.

This result characterizes the conditions under which Algorithm 1 converges almost surely. It also provides a sample complexity upper bound for the multi-level stochastic compositional optimization problems. In the case where $T = 2$, this result guarantees a convergence rate of $\mathcal{O}(n^{-1/4})$ for convex problems and an $\mathcal{O}(n^{-2/3})$ rate of convergence for strongly convex problems, which matches the convergence rates of convex and strongly convex basic-SCGD given in [26] respectively.

When dealing with nonconvex objectives (e.g., part (ii)), we assume the condition "all samples generated by the $\mathcal{SO}$ are uniformly bounded," which may seem somewhat restrictive. The same condition will be used in subsequent analysis for nonconvex problems. It can be verified in many practical training problems in machine learning that involve finite data sets. We also note that it is possible to extend the result to the case where this condition is replaced by a milder condition like "$\mathcal{X}$ is a closed and bounded." Such an extension would require a more sophisticated update rule and more complex analysis to deal with the constraint, which is beyond the scope of the current paper. In this paper, we choose to present the most succinct result for nonconvex problems under the uniformly bounded assumption.

The detailed proof of Theorem 2.1 can be derived similarly as in the proofs of Theorem 3.1, 3.2 and 3.3. In this paper, we omit the proof to avoid repetition, which can be found in our online supplementary materials.

3 Accelerated Multi-Level Stochastic Gradient Algorithm

In the previous section, we establish an $\mathcal{O}(n^{-1/2^T})$ rate of convergence for the T-level stochastic compositional optimization problem. A key question is whether and when we can better utilize noisy gradients of component functions and improve the overall convergence rate.

Throughout this section, in addition to Assumption 2.1, we impose the following assumption:

Assumption 3.1. Let $C_1, C_2, \cdots, C_T, V_1, \cdots, V_T$ be positive scalars.

(i) The samples $f^{(j)}_{\omega_{T,k}}(\cdot), \widetilde{\nabla} f^{(j)}_{\omega_{T,k}}(\cdot)$ have bounded fourth-order moments such that with probability 1,

$$\mathbb{E}[\|\widetilde{\nabla} f^{(T)}_{\omega_{T,0}}(x)\|^4 | \omega_{1,0}, \cdots, \omega_{T-1,0}] \le C_T^2,$$
$$\text{and } \mathbb{E}[\|f^{(T)}_{\omega_{T,0}}(x) - f^{(T)}(x)\|^4 | \omega_{1,0}, \cdots, \omega_{T-1,0}] \le V_T^2, \ \ \forall x \in \mathcal{X}.$$

(ii) The samples $f^{(j)}_{\omega_{j,k}}(\cdot)$'s and $\nabla f^{(j)}_{\omega_{j,k}}(\cdot)$'s have bounded fourth-order moments such that with probability 1,

$$\mathbb{E}[\|\nabla f^{(j)}_{\omega_{j,0}}(x_j)\|^4 | \omega_{1,0}, \cdots, \omega_{j-1,0}] \le C_j^2,$$
$$\text{and } \mathbb{E}[\|f^{(j)}_{\omega_{j,0}}(x_j) - f^{(j)}(x_j)\|^4 | \omega_{1,0}, \cdots, \omega_{j-1,0}] \le V_j^2, \ \ \forall x_j \in \mathbb{R}^{d_j}, \quad \text{and for } j = T-1, \cdots, 1.$$

We also consider the case when the first inner level function $f^{(T)}$ also has Lipschitz continuous gradients. In some part of our subsequent analysis, we make the following assumption.

Assumption 3.2. The function $f^{(T)}$ has Lipschitz continuous gradient such that

$$\|\nabla f^{(T)}(x) - \nabla f^{(T)}(\bar{x})\| \le L_T \|x - \bar{x}\|,$$

for all $x, \bar{x} \in \mathcal{X}$.

In what follows, we propose an accelerated algorithm to better utilize those smoothness properties and achieve improved convergence rates.

3.1 An Extrapolation-Interpolation Scheme For Acceleration

The basic idea of acceleration is to refine the running estimates of the value functions by using additional extrapolations. The same idea has been used for the case where $T = 2$. Specifically, in [26], with an additional bounded fourth moments assumption, the authors developed an accelerated SCGD (a-SCGD) algorithm and achieved faster convergence rate using an extra extrapolation step per iteration.

Now we develop a new accelerated algorithm for the multi-level problem that runs as follows: At the k-th iteration, we first update the main iterate solution x_{k+1} by the chain rule,

$$x_{k+1} = \Pi_{\mathcal{X}}\Big\{x_k - \alpha_k \widetilde{\nabla} f^{(T)}_{\omega_{T,k}}(x_k)\nabla f^{(T-1)}_{\omega_{T-1,k}}(y^{(T-1)}_k)\cdots\nabla f^{(1)}_{\omega_{1,k}}(y^{(1)}_k)\Big\}.$$

We then update the running estimate $y^{(T-1)}_k$ for $\mathbb{E}_{\omega_{T,k}}[f^{(T)}_{\omega_{T,k}}(x_k)|\omega_{1,k},\cdots,\omega_{T-1,k}]$ by taking weighted average between the new sample and the previous estimate. Specifically, we update $y^{(T-1)}_k$ by letting

$$y^{(T-1)}_{k+1} = (1-\beta_{T-1,k})y^{(T-1)}_k + \beta_{T-1,k} f^{(T)}_{\omega_{T,k+1}}(x_{k+1}).$$

Next, we conduct extrapolation steps for acceleration. The intuition is that we can use sample gradients of individual component functions more efficiently when these functions are smooth, which allows us to obtain better estimates of $f^{(j)}$'s. In particular, our accelerated updates for the auxiliary variables are performing first-oder running approximations of the true values. In comparison, the corresponding updates used in T-SCGD can be viewed as zeroth-order running approximations. Specifically, at the k-th iteration, we refine our estimate $y^{(j)}_{k+1}$ by taking an additional extrapolation step and obtaining a new auxiliary variable $\widehat{y}^{(j)}_{k+1}$:

$$\widehat{y}^{(j)}_{k+1} = (1-1/\beta_{j,k})y^{(j+1)}_k + y^{(j+1)}_{k+1}/\beta_{j,k}.$$

Then, when we update $y^{(j)}_{k+1}$, we plug in this auxiliary variable aiming for a better estimate that

$$y^{(j)}_{k+1} = (1-\beta_{j,k})y^{(j)}_k + \beta_{j,k}\cdot f^{(j+1)}_{\omega_{j+1,k+1}}(\widehat{y}^{(j)}_{k+1}).$$

We point out that this is essentially a weighted smoothing scheme, where $\widehat{y}^{(j)}_k$'s are obtained through extrapolation steps to further utilize the smoothness in order to improve the convergence rate. Roughly speaking, this further extrapolation step helps us achieve estimators $y^{(j)}_{k+1}$'s for $f^{(j+1)}(y^{(j+1)}_{k+1})$'s accurate up to the second order terms if we take Taylor expansions of $f^{(j)}$'s. In comparison, without the extrapolation, if we directly plug in $y^{(j+1)}_{k+1}$'s instead, the estimators are only accurate up to the first order terms. We call this an *accelerating update step*. Note that here we do not assume $f^{(T)}$ has Lipschitz continuous gradient as in some applications, $f^{(T)}$ includes some sparse-inducing regularization terms and is not continuously differentiable.

When Assumption 3.2 holds, we update the main iteration by the chain rule, and then apply extrapolation to this level to better utilize the smoothness. That is, we refine our estimate $y^{(T-1)}_{k+1}$ with an additional extrapolation step and an auxiliary variable $\widehat{y}^{(T-1)}_{k+1}$ as

$$\widehat{y}^{(T-1)}_{k+1} = (1-1/\beta_{T-1,k})x_k + x_{k+1}/\beta_{T-1,k}.$$

Next, we update $y_{k+1}^{(T-1)}$ by this auxiliary variable such that

$$y_{k+1}^{(T-1)} = (1-\beta_{T-1,k})y_k^{(T-1)} + \beta_{T-1,k} f_{\omega_{T,k+1}}^{(T)}(\widehat{y}_{k+1}^{(T-1)}).$$

For the remaining levels, we apply the same procedure as in the accelerating update steps previously described. We summarize those two slightly different accelerated algorithms in Algorithm 2.

In the remaining part of this section, we provide theoretical guarantees for this accelerated algorithm. We first provide the almost sure convergence result that almost surely, our algorithm converges to an optimal solution when the problem is convex, and any limiting point of the generated solution path is a stationary point. Next, we obtain an improved convergence rate for our algorithm for general nonconvex objective functions. Furthermore, we investigate the case when the objective function is strongly convex, and show that one can achieve faster convergence. For all results, we provide outlines and key lemmas in the main text, and defer the detailed proofs in Appendix A, B and C.

3.2 Almost Sure Convergence Of a-TSCGD

We first investigate whether and under what condition the algorithm converges almost surely. In particular, we provide sufficient conditions of the stepsizes, such that when the problem is convex, the algorithm converges to an optimal solution almost surely, and when the problem is nonconvex, all limiting points of the solution path generated by the algorithm are stationary points almost surely when $F(x)$ has Lipschitz continuous gradient.

Theorem 3.1 (Almost sure convergence for a-TSCGD)**.** Let Assumptions 2.1 and 3.1 hold, and let the stepsizes $\{\alpha_k\}, \{\beta_{T-1,k}\}, \cdots, \{\beta_{1,k}\}$ be such that

$$\sum_{k=0}^{\infty} \alpha_k = \infty, \sum_{k=0}^{\infty} \beta_{T,k} = \infty, \cdots, \sum_{k=0}^{\infty} \beta_{1,k} = \infty,$$

$$\sum_{k=0}^{\infty} \left(\alpha_k^2 + \beta_{T-1,k}^2 + \cdots + \beta_{1,k}^2 + \frac{\alpha_k^2}{\beta_{T-1,k}} + \cdots + \frac{\alpha_k^2}{\beta_{1,k}} \right) < \infty,$$

and

$$\sum_{k=0}^{\infty} \left(\frac{\beta_{T-1,k}^4}{\beta_{T-2,k}^3} + \cdots + \frac{\beta_{2,k}^4}{\beta_{1,k}^3} \right) < \infty.$$

Let $\left\{(x_k, y_k^{(T-1)}, \cdots, y_k^{(1)})\right\}_{k=0}^{\infty}$ be the sequence generated by Algorithm 2 starting with an arbitrary initial point $(x_0, y_0^{(T-1)}, \cdots, y_0^{(1)})$. Then:

(a) If F is convex, the sequence $\{x_k\}_{k=0}^{\infty}$ converges almost surely to a random point in the set of optimal solutions to problem (1.1).

(b) Suppose in addition that Assumption 2.2 holds, $\mathcal{X} = \mathbb{R}^{d_T}$, and all samples generated by the $\mathcal{SO}$ are uniformly bounded, then any limiting point of the sequence $\{x_k\}_{k=0}^{\infty}$ is a stationary point of problem (1.1) almost surely.

Algorithm 2 Accelerated T-Level Stochastic Compositional Gradient Descent (a-TSCGD)

Input: $x_0 \in \mathbb{R}^{d_T}, y_0^{(j)} \in \mathbb{R}^{d_j}$ for $j = T-1, ..., 1$, $\mathcal{SO}$, K, stepsizes $\{\alpha_k\}_{k=0}^K$,$\{\beta_{j,k}\}_{k=0}^K$ for $j = T-1, ..., 1$.

Output: The sequence $\{x_k\}_{k=0}^K$.

for $k = 0, 1, 2, ..., K$ **do**

Query the $\mathcal{SO}$ for the sample values of $f^{(T)}, \cdots, f^{(1)}$ at $x_k, y_k^{(T-1)}, \cdots, y_k^{(1)}$, obtain $\widetilde{\nabla} f_{\omega_{T,k}}^{(T)}(x_k)$, $\nabla f_{\omega_{T-1,k+1}}^{(T-1)}(y_k^{(T-1)}), \cdots, \nabla f_{\omega_{1,k}}^{(1)}(y_k^{(1)})$.

Update the main iterate by

$$x_{k+1} = \Pi_{\mathcal{X}}\Big\{x_k - \alpha_k \widetilde{\nabla} f_{\omega_{T,k}}^{(T)}(x_k) \nabla f_{\omega_{T-1,k}}^{(T-1)}(y_k^{(T-1)}) \cdots \nabla f_{\omega_{1,k}}^{(1)}(y_k^{(1)})\Big\}.$$

if Assumption 3.2 is known to hold **then**

Update the auxiliary variable $\widehat{y}_{k+1}^{(T-1)}$ by

$$\widehat{y}_{k+1}^{(T-1)} = (1 - 1/\beta_{T-1,k})x_k + x_{k+1}/\beta_{T-1,k}.$$

Query the $\mathcal{SO}$ for the sample value of $f^{(T)}$ at $\widehat{y}_{k+1}^{(T-1)}$, obtain $f_{\omega_{T,k+1}}^{(T)}(\widehat{y}_{k+1}^{(T-1)})$.

Update

$$y_{k+1}^{(T-1)} = (1 - \beta_{T-1,k})y_k^{(T-1)} + \beta_{T-1,k} f_{\omega_{T,k+1}}^{(T)}(\widehat{y}_{k+1}^{(T-1)}).$$

else if Assumption 3.2 is NOT known to hold **then**

Query the $\mathcal{SO}$ for the sample values of $f^{(T)}$ at x_{k+1}, obtain $f_{\omega_{T,k+1}}^{(T)}(x_{k+1})$.

Update $y^{(T-1)}$ by

$$y_{k+1}^{(T-1)} = (1 - \beta_{T-1,k})y_k^{(T-1)} + \beta_{T-1,k} f_{\omega_{T,k+1}}^{(T)}(x_{k+1}).$$

end if

for $j = T-1, \cdots, 2$ **do**

Update the auxiliary variable $\widehat{y}_{k+1}^{(j-1)}$ by

$$\widehat{y}_{k+1}^{(j-1)} = (1 - \frac{1}{\beta_{j-1,k}})y_k^{(j)} + \frac{1}{\beta_{j-1,k}} y_{k+1}^{(j)}.$$

Query the $\mathcal{SO}$ for the sample value of $f^{(j)}$ at $z_{k+1}^{(j-1)}$, obtain $f_{\omega_{j,k+1}}^{(j)}(z_{k+1}^{(j-1)})$.

Update $y^{(j)}$ by

$$y_{k+1}^{(j-1)} = (1 - \beta_{j-1,k})y_k^{(j-1)} + \beta_{j-1,k} f_{\omega_{j,k+1}}^{(j)}(\widehat{y}_{k+1}^{(j-1)}).$$

end for

end for

Furthermore, if Assumption 3.2 also holds, i.e., when $f^{(T)}$ has Lipschitz continuous gradient, then if the stepsizes also satisfy

$$\sum_{k=0}^{\infty} \frac{\alpha_k^4}{\beta_k^3} < \infty,$$

the assertions in (a) and (b) also hold.

Proof Outline. We provide the proof outline here for the case when the first inner level function $f^{(T)}$ is non-smooth. The analysis for problems with a smooth first inner level function could be derived from the non-smooth case, and we present the details for both cases in Appendix A.

We denote by $\mathbb{F}_k$ the collection of random variables up to the k-th iteration to help us better analyze the convergence properties:

$$\mathbb{F}_k = \Big\{ \{x_i\}_{i=0}^{k}, \{y_i^{(T-1)}\}_{i=0}^{k-1}, \cdots, \{y_i^{(1)}\}_{i=0}^{k-1}, \{\widehat{y}_i^{(T-2)}\}_{i=0}^{k-1}, \cdots, \{\widehat{y}_i^{(1)}\}_{i=0}^{k-1}, \{\omega_{T,i}\}_{i=1}^{k-1}, \cdots, \{\omega_{1,i}\}_{i=1}^{k-1} \Big\}.$$

To derive the almost sure convergence of Algorithm 2, we construct two different T-element super-martingales for the convex and non-convex objectives, respectively.

Firstly, for problems with convex objective F, in the k-th iteration, we have the following lemma to analyze the improvement from $\|x_k - x^*\|$ to $\|x_{k+1} - x^*\|$ by $\|y_k^{(T-1)} - f^{(T)}(x_k)\|$, $\|y_k^{(T-2)} - f^{(T-1)}(y_k^{(T-1)})\|$, $\cdots$, and $\|y_k^{(1)} - f^{(2)}(y_k^{(2)})\|$.

Lemma 3.1. Let Assumption 2.1 hold, and let $F = f^{(1)} \circ f^{(2)} \circ \cdots \circ f^{(T)}$ be convex. Then Algorithm 2 generates a sequence $\{(x_k, y_k^{(T-1)}, \cdots, y_k^{(1)})\}_{k=0}^{\infty}$ such that there exists a constant $C_0 > 0$ and an optimal solution $x^* \in \mathcal{X}^*$, for all k, with probability 1,

$$\begin{aligned}
&\mathbb{E}[\|x_{k+1} - x^*\|^2 | \mathbb{F}_k] \\
\leq &\Big(1 + \big[\frac{\alpha_k^2}{\beta_{T-1,k}} + \cdots + \frac{\alpha_k^2}{\beta_{1,k}}\big] C_0\Big) \|x_k - x^*\|^2 + \alpha_k^2 C_1 C_2 \cdots C_T - 2\alpha_k \big(F(x_k) - F^*\big) \\
&+ (T-1)\beta_{T-1,k} \mathbb{E}[\|y_k^{(T-1)} - f^{(T)}(x_k)\|^2 | \mathbb{F}_k] + (T-2)\beta_{T-2,k} \mathbb{E}[\|y_k^{(T-2)} - f^{(T-1)}(y_k^{(T-1)})\|^2 | \mathbb{F}_k] \\
&+ \cdots + \beta_{1,k} \mathbb{E}[\|y_k^{(1)} - f^{(2)}(y_k^{(2)})\|^2 | \mathbb{F}_k].
\end{aligned} \tag{3.1}$$

Lemma 3.1 states that for T-level SCGD with convex objective function F, the optimality error $\|x_{k+1} - x^*\|$ can be bounded by $\|x_k - x^*\|$, $\|y_k^{(T-1)} - f^{(T)}(x_k)\|$, $\|y_k^{(T-2)} - f^{(T-1)}(y_k^{(T-1)})\|$, $\cdots$, and $\|y_k^{(1)} - f^{(2)}(y_k^{(2)})\|$ in a super-martingale form.

Next, we present a lemma used in the analysis in part (b).

Lemma 3.2. Suppose that Assumption 2.1 and 2.2 hold, and $\mathcal{X} = \mathbb{R}^{d_T}$. Let $F^* = \min_{x \in \mathcal{X}} F(x)$, then Algorithm 2 generates a sequence $\{(x_k, y_k^{(T-1)}, \cdots, y_k^{(1)})\}_{k=0}^{\infty}$ such that

$$\begin{aligned}
&\mathbb{E}[F(x_{k+1}) - F^* | \mathbb{F}_k] \\
\leq\ & F(x_k) - F^* - \frac{\alpha_k}{2} \|\nabla F(x_k)\|^2 + \frac{1}{2} \alpha_k^2 L_F C_1 C_2 \cdots C_T + (T-1)\beta_{T-1,k} \mathbb{E}[\|y_k^{(T-1)} - f^{(T)}(x_k)\|^2 | \mathbb{F}_k] \\
&+ (T-2)\beta_{T-2,k} \mathbb{E}[\|y_k^{(T-2)} - f^{(T-1)}(y_k^{(T-1)})\|^2 | \mathbb{F}_k] + \cdots + \beta_{1,k} \mathbb{E}[\|y_k^{(1)} - f^{(2)}(y_k^{(2)})\|^2 | \mathbb{F}_k],
\end{aligned}$$

for k sufficiently large, with probability 1.

This lemma tells us that for T-level SCGD with general nonconvex objective function F, $(F(x_{k+1}) - F^*)$ can be bounded by $(F(x_k) - F^*)$, $\|y_k^{(T-1)} - f^{(T)}(x_k)\|$, $\|y_k^{(T-2)} - f^{(T-1)}(y_k^{(T-1)})\|$, $\cdots$, and $\|y_k^{(1)} - f^{(2)}(y_k^{(2)})\|$ in a super-martingale form. Similar as in Lemma 3.1, we shall construct the super-martingales for $\|y_k^{(T-1)} - f^{(T)}(x_k)\|$ and $\|y_k^{(j)} - f^{(j+1)}(y_k^{(j+1)})\|$ for $j = T-2, \cdots, 1$ respectively, and then use Lemma 3.6 to show the almost sure convergence of $(F(x_k) - F^*)$ for a T-level SCGD with nonconvex objective F. With further analysis, we show that any limiting point of the sequence $\{x_k\}_{k=0}^{\infty}$ is a stationary point with probability 1, which proves part (b) of Theorem 3.1.

Next, we analyze the term $\|y_k^{(j)} - f^{(j+1)}(y_k^{(j+1)})\|$ for $j = T-1, \cdots, 1$ and construct the proper super-martingales for them respectively.

Essentially, we construct a T-element super-martingale to derive the almost sure convergence of the algorithm. For the first inner level, since $f^{(T)}$ is non-smooth, we construct the super-martingale for this level as follows:

Lemma 3.3. Let Assumption 2.1 hold, and let $\{(x_k, y_k^{(T-1)}, \cdots, y_k^{(1)})\}_{k=0}^{\infty}$ be the sequence generated by Algorithm 2. Suppose $\mathbb{E}[\|x_{k+1} - x_k\|^2] \le \mathcal{O}(\alpha_k^2)$ for all k, then we have

(a) For all k, with probability 1,

$$\begin{aligned} &\mathbb{E}[\|y_{k+1}^{(T-1)} - f^{(T)}(x_{k+1})\|^2 | \mathbb{F}_{k+1}] \\ \le &(1 - \beta_{T-1,k})\|y_k^{(T-1)} - f^{(T)}(x_k)\|^2 + \beta_{T-1}^{-1} C_T \mathbb{E}[\|x_{k+1} - x_k\|^2 | \mathbb{F}_{k+1}] + 2V_T \beta_{T-1,k}^2. \end{aligned} \tag{3.2}$$

(b) If $\sum_{k=1}^{\infty} \alpha_k^2 / \beta_{T-1,k} < \infty$, then

$$\sum_{k=1}^{\infty} \beta_{T-1,k}^{-1} \mathbb{E}[\|x_{k+1} - x_k\|^2 | \mathbb{F}_{k+1}] < \infty, \quad w.p.1.$$

(c) There exists a constant $D_{T-1} \ge 0$ such that $\mathbb{E}[\|y_{k+1}^{(T-1)} - f^{(T)}(x_{k+1})\|^2] \le D_{T-1}$ for all k.

(d) $\mathbb{E}[\|y_{k+1}^{(T-1)} - y_k^{(T-1)}\|^2] \le \mathcal{O}(\beta_{T-1,k}^2)$ for all k.

With the additional finite fourth-moment Assumption 3.1, we can derive a stronger result in the following lemma.

Lemma 3.4. Let Assumptions 2.1 and 3.1 hold, and let $\{(x_k, y_k^{(T-1)}, \cdots, y_k^{(1)})\}_{k=0}^{\infty}$ be the sequence generated by Algorithm 2. Suppose $\mathbb{E}[\|x_{k+1} - x_k\|^4] \le \mathcal{O}(\alpha_k^4)$ for all k and $\alpha_k / \beta_{T-1,k} \to 0$ as $k \to 0$, in addition to Lemma 3.3 (a) (b) (c) and (d), we have:

(a) There exists a constant $S_{T-1} > 0$ such that $\mathbb{E}[\|y_k^{(T-1)} - f^{(T)}(x_k)\|^4] \le S_{T-1}$ for all k.

(b) $\mathbb{E}[\|y_{k+1}^{(T-1)} - y_k^{(T-1)}\|^4] \le \mathcal{O}(\beta_{T-1,k}^4)$ for all k.

Note that here we use $y_k^{(T)} = x_k$ and $\beta_{T,k} = \alpha_k$ for ease of notation. This lemma constructs super-martingales of $\left\{\|y_k^{(j)} - f^{(j+1)}(y_k^{(j+1)})\|\right\}_{k=1}^{\infty}$ for $j = T-1 \cdots, 1$ respectively, and it also shows that under proper assumptions, the tail part for the super-martingale, $\beta_j^{-1} C_{j+1} \mathbb{E}[\|y_{k+1}^{(j)} - y_k^{(j)}\|^2 | \mathbb{F}_k] + 2V_{j+1}\beta_{j,k}^2$, converges almost surely.

Next, to construct the super-martingale for the accelerating update steps, we present the following lemma.

Lemma 3.5. Let Assumption 2.1 and 3.1 hold, and let $\{(x_k, y_k^{(T-1)}, \cdots, y_k^{(1)})\}_{k=1}^{\infty}$ be the sequence generated by Algorithm 2. For $j = T-2, \cdots, 1$, suppose $\mathbb{E}[\|y_{k+1}^{(j+1)} - y_k^{(j+1)}\|^4] \leq \mathcal{O}(\beta_{j+1,k}^4)$ for all k and $\beta_{j+1,k}/\beta_{j,k} \to 0$ as $k \to 0$, then there exists a random variable $e_k^{(j)} \in \mathbb{F}_{k+1}$ for all k satisfying $\|y_k^{(j)} - f^{(j+1)}(y_k^{(j+1)})\| \leq e_k^{(j)}$ such that:

(a) For all k, with probability 1,

$$\mathbb{E}[[e_{k+1}^{(j)}]^2|\mathbb{F}_{k+1}] \leq (1 - \frac{\beta_{j,k}}{2})[e_k^{(j)}]^2 + 2\beta_{j,k}^2 V_{j+1} + \mathcal{O}\left(\frac{\mathbb{E}[\|y_{k+1}^{(j+1)} - y_k^{(j+1)}\|^4|\mathbb{F}_{k+1}]}{\beta_{j,k}^3}\right).$$

(b) If $\sum_{k=1}^{\infty} \beta_{j+1,k}^4/\beta_{j,k}^3 < \infty$, we have

$$\sum_{k=1}^{\infty} \frac{\mathbb{E}[\|y_{k+1}^{(j+1)} - y_k^{(j+1)}\|^4|\mathbb{F}_{k+1}]}{\beta_{j,k}^3} < \infty \quad w.p.1.$$

(c) There exists a constant $D_j \geq 0$ such that $\mathbb{E}[e_k^{(j)}]^2 \leq D_j$ for all k.

(d) There exists a constant $S_j \geq 0$ such that $\mathbb{E}[\|y_k^{(j)} - f^{(j+1)}(y_k^{(j+1)})\|^4] \leq S_j$ for all k.

(e) $\mathbb{E}[\|y_{k+1}^{(j)} - y_k^{(j)}\|^4] \leq \mathcal{O}(\beta_{j,k}^4)$ for all k.

Previous lemmas provide basic blocks for us to build a T-element super-martingale. We then provide the T-element super-martingale convergent lemma to establish the convergence property of $\{x_k - x^*\}$.

Lemma 3.6 (T-element supermartingale convergence)**.** Let $\{X_k\}$,$\{Y_k^{(T-1)}\}$,$\cdots$,$\{Y_k^{(1)}\}$, $\{\eta_k\}$, and $\{u_k^{(j)}\}$,$\{\mu_k^{(j)}\}$,$\{\theta_k^{(j)}\}$ for $j = 1, \cdots, T$ be sequences of nonnegative random variables such that

$$\mathbb{E}[X_{k+1}|\mathbb{G}_k] \leq (1 + \eta_k)X_k - u_k^{(T)} + \sum_{j=1}^{T-1} c_j \theta_k^{(j)} Y_k^{(j)} + \mu_k^{(T)},$$

and

$$\mathbb{E}[Y_{k+1}^{(T-1)}|\mathbb{G}_k] \leq (1 - \theta_k^{(j)})Y_k^{(j)} - u_k^{(j)} + \mu_k^{(j)}, \text{for } j = T-1, ..., 1,$$

for all k, where $\mathbb{G}_k$ is the collection of random variables

$$\left\{\{X_i\}_{i=0}^k, \{Y_i^{(T-1)}\}_{i=0}^k, \cdots, \{Y_i^{(1)}\}_{i=0}^k, \{\eta_i\}_{i=0}^k, \{u_i^{(j)}\}_{i=0}^k, \{\mu_i^{(j)}\}_{i=0}^k, \{\theta_i^{(j)}\}_{i=0}^k, \text{ for } j = 1, \cdots, T\right\},$$

and $c_{T-1}, c_{T-2} \cdots, c_1$ are positive scalars. Assume that

$$\sum_{k=0}^{\infty} \eta_k < \infty, \sum_{k=0}^{\infty} \mu_k^{(j)} < \infty, \quad \text{for } j = 1, \cdots, T.$$

Then $\{X_k\}, \{Y_k^{(1)}\}, \{Y_k^{(2)}\}, \cdots, \{Y_k^{(T-1)}\}$ converge almost surely to T nonnegative random variables respectively, and we have

$$\sum_{j=1}^{T} \sum_{k=0}^{\infty} u_k^{(j)} < \infty, \sum_{k=0}^{\infty} \sum_{j=1}^{T-1} c_j \theta_k^{(j)} Y_k^{(j)} < \infty \ \ w.p.1.$$

By Lemmas 3.1, 3.3, 3.4 and 3.5, we construct the T-element super-martingale and show its convergence by letting

$$
\begin{aligned}
&X_k = \|x_k - x^*\|^2, Y_k^{(T-1)} = \mathbb{E}[\|y_k^{(T-1)} - f^{(T)}(x_k)\|^2 | \mathbb{F}_k], \\
&Y_k^{(T-2)} = \mathbb{E}[[e_k^{(T-2)}]^2 | \mathbb{F}_k], \cdots, Y_k^{(1)} = \mathbb{E}[[e_k^{(1)}]^2 | \mathbb{F}_k], \\
&\eta_k = [\frac{\alpha_k^2}{\beta_{T-1,k}} + \cdots + \frac{\alpha_k^2}{\beta_{1,k}}] C_0, u_k^{(T)} = 2\alpha_k (F(x_k) - F^*), \\
&u_k^{(1)} = u_k^{(2)} = \cdots = u_k^{(T-1)} = 0, c_1 = 2, \cdots, c_{T-2} = 2(T-2), c_{T-1} = T-1, \\
&\mu_k^{(T-1)} = C_T \beta_{T-1,k}^{-1} \mathbb{E}[\|x_{k+1} - x_k\|^2 | \mathbb{F}_k] + 2V_T \beta_{T-1,k}^2, \\
&\mu_k^{(T-2)} = 2\beta_{T-2,k}^2 V_{T-1} + \mathcal{O}(\frac{\mathbb{E}[\|y_{k+1}^{(T-1)} - y_k^{(T-1)}\|^4 | \mathbb{F}_k]}{\beta_{T-2,k}^3}), \cdots, \\
&\mu_k^{(1)} = 2\beta_{1,k}^2 V_1 + \mathcal{O}(\frac{\mathbb{E}[\|y_{k+1}^{(2)} - y_k^{(2)}\|^4 | \mathbb{F}_k]}{\beta_{1,k}^3}), \\
&\mu_k^{(T)} = \alpha_k^2 C_1 C_2 \cdots C_T, \\
&\theta_k^{(1)} = \beta_{1,k}/2, \cdots, \theta_k^{(T-2)} = \beta_{T-2,k}/2, \theta_k^{(T-1)} = \beta_{T-1,k}.
\end{aligned}
$$

Under the conditions in Theorem 3.1, we obtain that the T-element super-martingale converges almost surely to T random variables by Lemma 3.6, thus $\|x_k - x^*\|$ converges almost surely, and

$$
\sum_{k=0}^{\infty} \alpha_k (F(x_k) - F^*) < \infty, \quad w.p.1,
$$

which further implies that

$$
\liminf_{k \to \infty} F(x_k) = F^*, \quad w.p.1.
$$

Finally, the following lemma shows the the sequence $\{x_k\}_{k=0}^{\infty}$ converges almost surely to an optimal solution to problem (1.1), which completes the proof of part (a).

Lemma 3.7. Let $\{(x_k, y_k^{(T-1)}, \cdots, y_k^{(1)})\}_{k=0}^{\infty}$ be the sequence generated by Algorithm 2. Let $F^* = F(x^*)$, where x^* is an optimal solution to problem (1.1). Suppose

$$
\liminf_{k \to \infty} F(x_k) = F^*, \quad w.p.1,
$$

then $\{x_k\}$ converges almost surely to a random point in the set of optimal solutions to problem (1.1).

For part (b), by Lemma 3.2 and Lemma 3.3, we construct the T-element super-martingale for general non-convex functions, and show $\{F(x_k) - F^*\}$ converges almost surely by Lemma 3.6, which further implies $\sum_{k=0}^{\infty} \alpha_k \|\nabla F(x_k)\|^2 < \infty$ with probability 1. Then, we have the following lemma concluding that any limiting point of the sequence $\{x_k\}$ is a stationary point of $F(x)$ with probability 1.

Lemma 3.8. Let $\{(x_k, y_k^{(T-1)}, \cdots, y_k^{(1)})\}_{k=0}^{\infty}$ be the sequence generated by Algorithm 2. Suppose $\sum_{k=0}^{\infty} \alpha_k = \infty$ and $\sum_{k=0}^{\infty} \alpha_k \|\nabla F(x_k)\|^2 < \infty$ with probability 1, and all the random variables generated by $\mathcal{SO}$ are uniformly bounded, then any limiting point of the sequence $\{x_k\}$ is a stationary point of $F(x)$ with probability 1.

This concludes the proof for part (b).

□

3.3 Convergence Rate Results For a-TSCGD

In this subsection, we study the rate of convergence of the algorithm. We consider stepsizes of the form

$$\alpha_k = k^{-a}, \beta_{T-1,k} = k^{-b_{T-1}}, \text{ and } \beta_{j,k} = 2k^{-b_j} \text{ for all } j = T-2, \cdots, 1,$$

where a and b_j's are real numbers if the first inner level function $f^{(T)}$ is nonsmooth, and we choose the step-sizes to be

$$\alpha_k = k^{-a}, \text{ and } \beta_{j,k} = 2k^{-b_j} \text{ for all } j = T-1, \cdots, 1,$$

if $f^{(T)}$ is smooth. After optimizing the rate over all a and b_j's, we get the following result for both convex and nonconvex $F(x)$.

Theorem 3.2 (Convergence rate of a-TSCGD)**.** Suppose that Assumptions 2.1, 2.2 and 3.1 hold and $\mathcal{X} = \mathbb{R}^{d_T}$. Let the stepsizes be $\alpha_k = k^{-a}$, $\beta_{T-1,k} = k^{-b_{T-1}}$ and $\beta_{j,k} = 2k^{-b_j}$ for $j = T-2, \cdots, 1$, where $a, b_{T-1}, ..., b_1 \in (0,1)$. If we choose the step-sizes as $a = \frac{4+T}{8+T}$ and $b_j = \frac{j+3}{8+T}$ for $j = T-2, ..., 1$, letting $\{(x_k, y_k^{(T-1)}, \cdots, y_k^{(1)})\}_{k=0}^{\infty}$ be the sequence generated by a-TSCGD Algorithm 2, we obtain

$$\frac{\sum_{k=1}^{n} \mathbb{E}[\|\nabla F(x_k)\|^2]}{n} \leq \mathcal{O}(n^{-4/(8+T)}).$$

Furthermore, if Assumption 3.2 also holds, Algorithm 2 achieves

$$\frac{\sum_{k=1}^{n} \mathbb{E}[\|\nabla F(x_k)\|^2]}{n} \leq \mathcal{O}(n^{-4/(7+T)}),$$

with $\alpha_k = k^{-a}$ and $\beta_{j,k} = 2k^{-b_j}$, where $a = \frac{3+T}{7+T}$ and $b_j = \frac{j+3}{7+T}$ for $j = T-1, ..., 1$.

Proof Outline. We present the outline of proof here and defer the detailed analysis in Appendix B.

We first derive the convergence rate of $\|y_{k+1}^{(j)} - f^{(j+1)}(y_{k+1}^{(j+1)})\|$ and $\|y_{k+1}^{(j)} - y_k^{(j)}\|$ for $j = T-1, \cdots, 1$. By Lemma 3.3 and Lemma B.1 in the Appendix, we have the following lemma characterizing the corresponding convergence rates:

Lemma 3.9. Let Assumption 2.1 hold, and let $\{(x_k, y_k^{(T-1)}, \cdots, y_k^{(1)})\}_{k=0}^{\infty}$ be the sequence generated by Algorithm 2. Consider the basic update step for the first inner level, we have

$$\mathbb{E}[\|y_k^{(T-1)} - f^{(T)}(x_k)\|^2] \leq \mathcal{O}(k^{-2a+2b_{T-1}}) + \mathcal{O}(k^{-b_{T-1}}) \quad \text{for all } k.$$

For the accelerating update steps, by Lemma 3.5 and Lemma B.1 in the Appendix, we have the following result:

Lemma 3.10. Let Assumptions 2.1 and 3.1 hold, and let $\{(x_k, y_k^{(T-1)}, \cdots, y_k^{(1)})\}_{k=0}^{\infty}$ be the sequence generated by Algorithm 2. Then for any accelerated update step, we have for all k

$$\mathbb{E}[\|y_{k+1}^{(j)} - f^{(j+1)}(y_{k+1}^{(j+1)})\|^2] \leq \mathcal{O}(k^{4(b_j - b_{j+1})}) + \mathcal{O}(k^{-b_j}), \quad j = T-2, \cdots, 1.$$

Under additional Assumption 2.2 that F has Lipschitz gradient, we obtain the following result.

Lemma 3.11. Let Assumptions 2.1, 2.2 and 3.1 hold, and let $\{(x_k, y_k^{(T-1)}, \cdots, y_k^{(1)})\}_{k=0}^{\infty}$ be the sequence generated by Algorithm 2, then we have for all k

$$\begin{aligned}
&\mathbb{E}[\|\nabla F(x_k)\|^2] \\
\leq& 2\alpha_k^{-1}\mathbb{E}\Big[F(x_k)] - 2\alpha_k^{-1}\mathbb{E}[F(x_{k+1})\Big] + \mathcal{O}\left(\mathbb{E}[\|y_{k+1}^{(T-1)} - f^{(T)}(x_k)\|^2]\right) + \mathcal{O}\left(\mathbb{E}[\|y_{k+1}^{(T-2)} - f^{(T-1)}(y_{k+1}^{(T-1)})\|^2]\right) \\
&+ \cdots + \mathcal{O}\left(\mathbb{E}[\|y_{k+1}^{(1)} - f^{(2)}(y_{k+1}^{(2)})\|^2]\right) + \mathcal{O}(\alpha_k).
\end{aligned}$$

Summing up the inequalities in the previous lemma from $k = 0$ to n, by Lemma 3.9 and Lemma 3.10, we obtain

$$\begin{aligned}
\frac{\sum_{k=1}^n \mathbb{E}[\|\nabla F(x_k)\|^2]}{n} &\leq \mathcal{O}\Big(n^{a-1} + n^{-2a+2b_{T-1}}\mathbb{I}^{\log n}_{2(a-b_{T-1})=1} + n^{-b_{T-1}} + n^{-a}\Big) \\
&\quad + \mathcal{O}\Big(\sum_{j=1}^{T-2}[n^{4(b_j - b_{j+1})}\mathbb{I}^{\log n}_{4(b_{j+1}-b_j)=1} + n^{-b_j}]\Big) \\
&\leq \mathcal{O}(n^{-4/(8+T)}),
\end{aligned}$$

by choosing $a = \frac{4+T}{8+T}$ and $b_j = \frac{3+j}{8+T}$ for $j = T-1, \cdots, 1$.

Furthermore, if Assumption 3.2 also holds, i.e., the first inner level function $f^{(T)}$ has Lipschitz continuous gradient, then the first inner level could also be updated by the accelerating update rule. By similar analysis as in Lemma 3.10, we have for all k,

$$\mathbb{E}[\|y_{k+1}^{(T-1)} - f^{(T)}(x_{k+1})\|^2] \leq \mathcal{O}(k^{4(b_{T-1}-a)}) + \mathcal{O}(k^{-b_{T-1}}).$$

Combine this inequality with Lemmas 3.10 and 3.11, by choosing $a = \frac{3+T}{7+T}$ and $b_j = \frac{3+j}{7+T}$ for $j = T-1, \cdots, 1$, we obtain

$$\frac{\sum_{k=1}^n \mathbb{E}[\|\nabla F(x_k)\|^2]}{n} \leq \mathcal{O}(n^{-4/(7+T)}),$$

which completes the proof. □

This result shows that one can solve the multi-level composition problem using few calls to the sampling oracle when individual component functions are smooth. In the special case where $T = 2$, when the first inner level is smooth, our result strictly improves the convergence rate of the a-SCGD in [26] from $\mathcal{O}(n^{-2/7})$ to $\mathcal{O}(n^{-4/9})$. In this case our result matches the convergence rate by ASC-PG in [28]. To the best of our knowledge, our results for the T-level problem strictly improve and generalize existing results which work for the case where $T = 2$.

Next we investigate the convergence rate of Algorithm 2 for optimally strongly convex objective defined in (2.3). In the next theorem, we prove that for optimally strongly convex objective, our algorithm converges faster. We defer the detailed proof to Appendix C.

Theorem 3.3 (Convergence rate of a-TSCGD for strongly convex problems)**.** Let Assumptions 2.1, 2.2 and 3.1 hold. Suppose that the objective function $F(x)$ in (1.1) is optimally strongly convex with some parameter $\lambda > 0$ defined in (2.3) . Set $\alpha_k = \frac{1}{\lambda}k^{-a}$, $\beta_{T-1,k} = k^{-b_{T-1}}$ and $\beta_{j,k} = 2k^{-b_j}$ for $j = T-2, \cdots, 1$. Let $\{(x_k, y_k^{(T-1)}, \cdots, y_k^{(1)})\}_{k=0}^{\infty}$ be the sequence generated by a-TSCGD Algorithm 2, then

$$\mathbb{E}[\|x_n - \Pi_{\mathcal{X}^*}(x_n)\|^2] \leq \mathcal{O}\Big(n^{-a} + n^{-2(a-b_{T-1})} + n^{-b_{T-1}} + \sum_{j=1}^{T-2}[n^{-4(b_{j+1}-b_j)} + n^{-b_j}]\Big).$$

With the choice of $a = 1, b_{T-1} = \frac{2+T}{4+T}, b_{T-2} = \frac{1+T}{4+T}, \cdots, b_1 = \frac{4}{4+T}$, we have

$$\mathbb{E}[\|x_n - \Pi_{\mathcal{X}^*}(x_n)\|^2] \leq \mathcal{O}(n^{-4/(4+T)}).$$

Furthermore, if Assumption 3.2 also holds, Algorithm 2 achieves

$$\mathbb{E}[\|x_n - \Pi_{\mathcal{X}^*}(x_n)\|^2] \leq \mathcal{O}(n^{-4/(3+T)}),$$

with the stepsizes being $\alpha_k = \frac{1}{\lambda}k^{-a}$ and $\beta_{j,k} = 2k^{-b_j}$, where $a = 1$ and $b_j = \frac{3+j}{3+T}$ for $j = T-1, \cdots, 1$.

This result shows that our algorithm achieves a faster convergence for those problems of optimally strongly convexity in the objective functions. For the speical case $T = 1$ with a smooth strongly convex function, this result achieves a convergence rate of $\mathcal{O}(n^{-1})$, which meets the convergence rate of the single-level strongly convex stochastic optimization. Besides, for a special case $T = 2$ with a smooth first inner level function, this result achieves a convergence rate of $\mathcal{O}(n^{-4/5})$, which matches the convergence rate ASC-PG in [28] for optimally strongly convex problems.

4 Example and Numerical Experiments

In this section, we provide a practical example of the T-level stochastic compositional optimization problem (1.1), the risk-averse stochastic optimization, and conduct numerical experiments. Risk-averse stochastic optimization finds wide applications in many fields such as risk management [4] and government planning [3]. Among different formulations of risk-averse stochastic optimization problems, one particular important problem is the mean-deviation risk-averse optimization problem that

$$\min_x \rho(U(x,w)) := \min_x \left\{ \mathbb{E}_\omega \left[U(x,\omega)\right] + \lambda \mathbb{E}\Big[\left(\mathbb{E}\left[U(x,\omega)\right] - U(x,\omega)\right)_+^p \Big]^{1/p} \right\}. \tag{4.1}$$

Here the objective ρ is the composition of three expected-value functions. It is also a law-invariant coherent risk measure. See [22, 1] for more detailed discussions.

This problem falls into the problem class (1.1) as a three-level stochastic compositional optimization problem. In particular, the problem is equivalent to

$$\min_x (f^{(1)} \circ f^{(2)} \circ f^{(3)})(x),$$

where

$$f^{(1)}\big((y_1, y_2)\big) = y_1 - y_2^{1/p}, f^{(2)}\big(z, x\big) = \big(z, \mathbb{E}_\omega\left[(z - U(x,\omega))_+^p\right]\big), \text{ and } f^{(3)}(x) = \left(\mathbb{E}_\omega\left[U(x,\omega)\right], x\right).$$

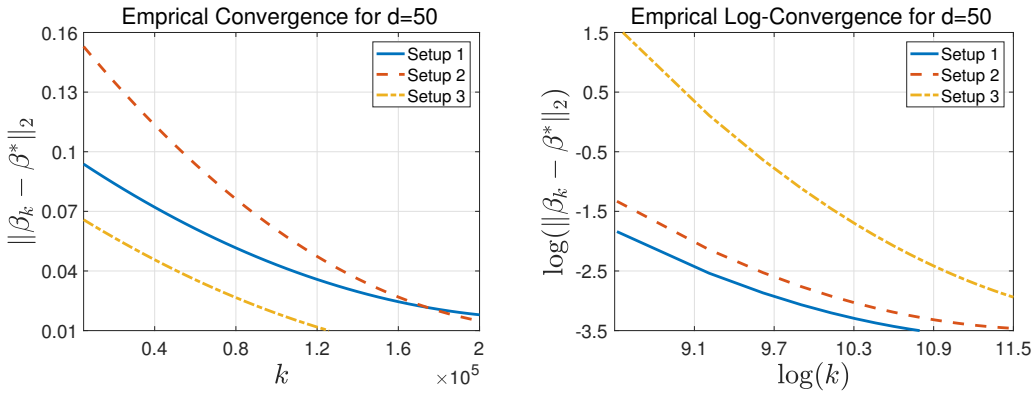

Figure 2: Averaged difference between generated solution and the optimal solution and empirical convergence rate when $d = 50$

Note that this problem involves only two random variables (in the nested inner functions $f^{(2)}$ and $f^{(3)}$), yet it is a three-level composition problem due to the outer function $f^{(1)}$. We remark that stochastic composition optimization is challenging due to the bias induced by using the chain rule to calculate stochastic gradients. In three-level problems, the bias is caused by the inner two levels, so the current problem is as hard as any three-level problem, even though its most outer level is deterministic. As a result, it can not be solved using existing methods for the two-level problem. Using methods developed in this paper, we can now solve it using a three-level SCGD algorithm.

Next, we conduct numerical experiments. We consider the risk-averse stochastic optimization in a regression setting. In particular, consider a linear model $Y = X\beta^* + \epsilon$, where we assume all samples of X and ϵ are independently and identically distributed. Our goal is to estimate β^*, and we consider a risk-averse formulation. Consider the risk-averse optimization problem (4.1). Denoting the i-th sample by $\omega_i = \{x_i, y_i\}$, we take

$$U(\beta, \omega_i) = (y_i - x_i^T \beta)^2,$$

and we set $p = 2$. To the best of our knowledge, our algorithm is the first gradient-based method which can be adopted to solve this 3-level stochastic optimization problem. We point out that this approach of risk-averse regression tends to provide "stable" solutions. This defines a general notion of stability in statistics in [18, 25], where the stability is usually defined as variance, and we also penalize the "good" cases when the empirical error is smaller than its expectation. In comparison, in our approach, we do not penalize these "good" cases.

Let the dimension of the covariate x_i be d. We consider three setups to generate the data that

- Setup 1: $X \sim N(0, I_d)$.
- Setup 2: $X \sim N(0, \Sigma)$, where $\Sigma_{jj} = 1$ and $\Sigma_{jk} = 0.5$ for $j, k = 1, ..., d$ and $j \neq k$.
- Setup 3: $X \sim N(0, \Sigma)$, where $\Sigma_{jk} = 0.5e^{-\frac{|j-k|}{d}}$.

Since our problem is convex, by our theoretical analysis, the generated sequence of solutions converges to the optimal solution. As the true optimal solution is unknown (Note that β^* is

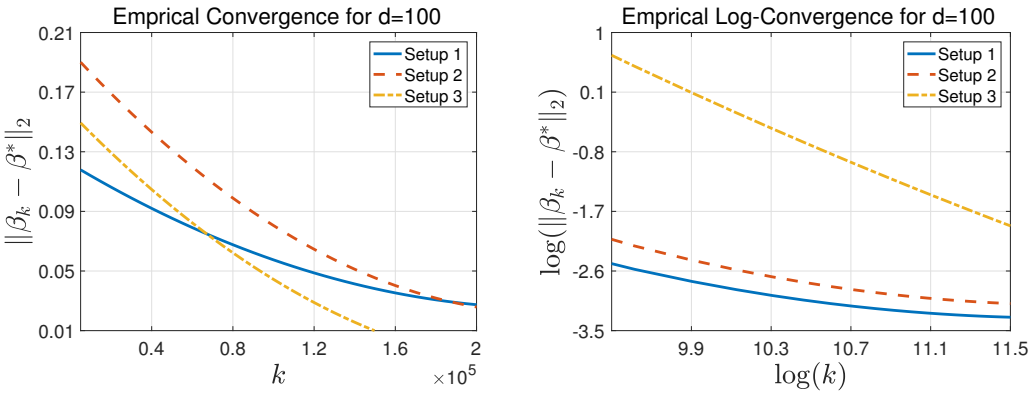

Figure 3: Averaged difference between generated solution and the optimal solution and empirical convergence rate when $d = 100$.

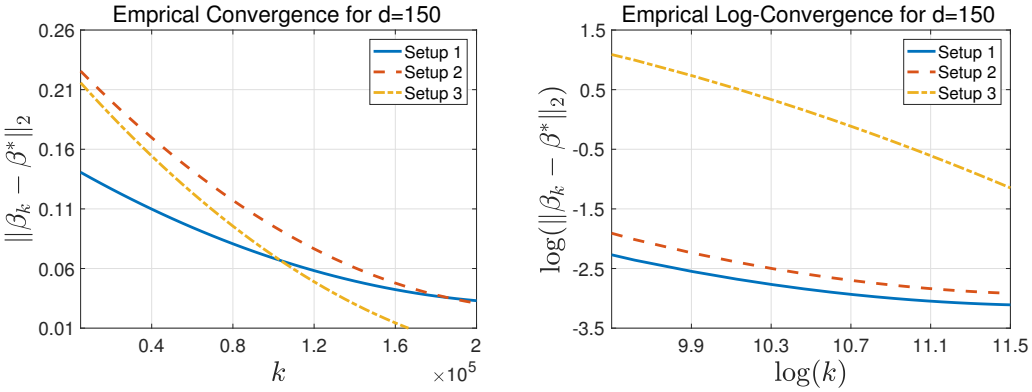

Figure 4: Averaged difference between generated solution and the optimal solution and empirical convergence rate when $d = 150$.

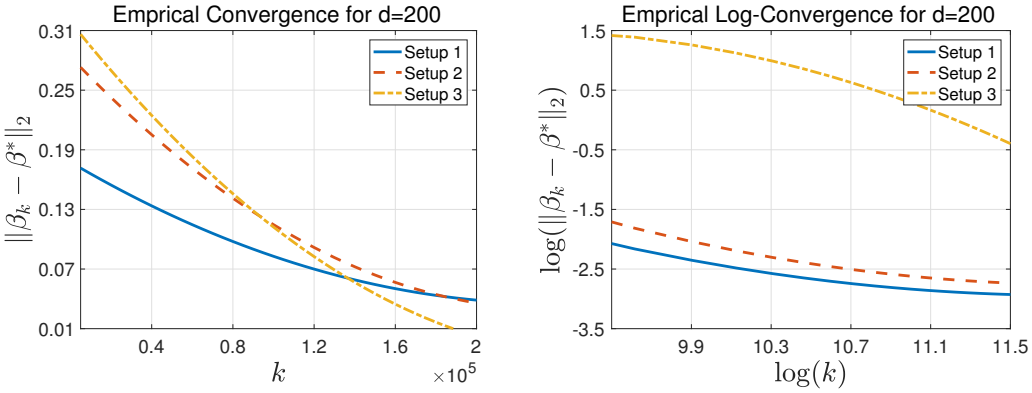

Figure 5: Averaged difference between generated solution and the optimal solution and empirical convergence rate when $d = 200$.

not necessarily the optimal solution), we take the solution after 500,000 iterations as the optimal solution. Meanwhile, in all setups, we draw the random variables $\epsilon \sim N(0, 0.2)$, and generate each component of $\beta^* \in \mathbb{R}^d$ independently from a standard normal distribution. We set the stepsizes to be $\alpha_k = k^{-3/5}$, $\beta_{2,k} = 2k^{-1/2}$, and $\beta_{1,k} = 2k^{-2/5}$. The samples of X are generated independently by the distribution specified in corresponding setup. In each iteration of the algorithm, we draw a new sample of (X, Y), and update the solution using Algorithm 2.

We have experimented the proposed algorithm with multiple values of the dimension, i.e., $d \in \{50, 100, 150, 200\}$. For each instance, we run 100 replications and plot the averaged differences between the solution at the k-th iteration β_k and the optimal solution $\widehat{\beta}$ in Figures 2, 3, 4 and 5. Let us study the performance of the algorithm when d varies. Comparing to case where $d = 50$, we notice that it takes approximately twice many iterations to reach the same accuracy in the case where $d = 200$. This is mainly because the variance of stochastic gradient increases as the dimension grows, due to the fact that the noise in our experiment is Gaussian with unit variance per entry. As a result, the hidden constant inside the error bound, involving sums and products of multi-level variances and Lipschitz constants, also grows polynomially as d grows. Therefore it takes more iterations for the algorithm to converge to the same accuracy level when the overall variance increases with growing dimensions.

Besides, to further investigate empirical rates of convergence under all different settings, we plot the averaged $\log(k)$ vs. $\log(\|\beta_k - \widehat{\beta}\|)$ after 100 replications in the figures, where $\widehat{\beta}$ is the optimal solution. As we can see from the figures, the slopes remain the same regardless of the dimension d. We find that for all cases, the slopes of the lines are close to $-2/5$, which matches our theoretical analysis that our algorithm converges at a rate of $\mathcal{O}(k^{-2/5})$ for three-level problems.

5 Conclusion

In this paper, we propose the first gradient-type algorithms for a class of multi-level stochastic compositional optimization problems. We provide strong theoretical guarantees for our algorithms. In particular, we prove almost sure convergence results that when the problem is convex, our algorithm converges to an optimal solution, and when the problem is nonconvex, every limiting point of the sequence of solutions is an stationary point. Under various assumptions, we further characterize the rates of convergence of our algorithms. In the case where $T = 2$, our convergence rate result matches and strictly generalizes the best known result by [28]. In the case where $T \geq 3$, our results provide the first few benchmarks on the sample complexity for solving multi-level stochastic optimization problems.

There are several interesting future research questions. First, our convergence rate result requires that the inner-level functions $f^{(2)}, \cdots, f^{(T)}$ be smooth. It is unclear how to achieve fast convergence when some of these functions are non-smooth. Second, it is not clear whether the convergence rate can be improved or not. We are not aware of any sample complexity lower bound for the multi-level stochastic optimization problem. Third, it is of practical interest to consider the special case where all expectations are finite sums. In this case, one may conjecture that variance reduction can be used to further improve the algorithms' efficiency.

References

[1] S. Ahmed, U. Çakmak, and A. Shapiro, *Coherent risk measures in inventory problems*, European Journal of Operational Research, 182 (2007), pp. 226–238.

[2] F. Bach and E. Moulines, *Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$*, in Advances in Neural Information Processing Systems, 2013, pp. 773–781.

[3] S. Bruno, S. Ahmed, A. Shapiro, and A. Street, *Risk neutral and risk averse approaches to multistage renewable investment planning under uncertainty*, European Journal of Operational Research, 250 (2016), pp. 979–989.

[4] S. Cole, X. Giné, and J. Vickery, *How does risk management influence production decisions? Evidence from a field experiment*, Review of Financial Studies, (2016), p. hhw080.

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

[6] D. Dentcheva, S. Penev, and A. Ruszczyński, *Statistical estimation of composite risk functionals and risk optimization problems*, Annals of the Institute of Statistical Mathematics, 69 (2017), pp. 737–760.

[7] J. Duchi, E. Hazan, and Y. Singer, *Adaptive subgradient methods for online learning and stochastic optimization*, Journal of Machine Learning Research, 12 (2011), pp. 2121–2159.

[8] Y. Ermoliev, *Methods of Stochastic Programming*, Monographs in Optimization and OR, Nauka, Moscow, 1976.

[9] R. Ge, F. Huang, C. Jin, and Y. Yuan, *Escaping from saddle points-online stochastic gradient for tensor decomposition.*, in Conference of Learning Theory, 2015, pp. 797–842.

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

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

[12] J. Konečnỳ and P. Richtárik, *Semi-stochastic gradient descent methods*, Frontiers in Applied Mathematics and Statistics, 3 (2017), p. 9.

[13] G. Lan, A. Nemirovski, and A. Shapiro, *Validation analysis of mirror descent stochastic approximation method*, Mathematical Programming, 134 (2012), pp. 425–458.

[14] Y. LeCun, Y. Bengio, and G. Hinton, *Deep learning*, Nature, 521 (2015), pp. 436–444.

[15] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, *Gradient descent only converges to minimizers*, in Conference on Learning Theory, 2016, pp. 1246–1257.

[16] C. J. Li, M. Wang, H. Liu, and T. Zhang, *Near-optimal stochastic approximation for online principal component estimation*, arXiv preprint arXiv:1603.05305, (2016).

[17] X. Lian, M. Wang, and J. Liu, *Finite-sum composition optimization via variance reduced gradient descent*, arXiv preprint arXiv:1610.04674, (2016).

[18] C. Lim and B. Yu, *Estimation stability with cross-validation (escv)*, Journal of Computational and Graphical Statistics, 25 (2016), pp. 464–492.

[19] D. Needell, R. Ward, and N. Srebro, *Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm*, in Advances in Neural Information Processing Systems, 2014, pp. 1017–1025.

[20] A. Rakhlin, O. Shamir, and K. Sridharan, *Making gradient descent optimal for strongly convex stochastic optimization*, in International Conference on Machine Learning, 2012, pp. 449–456.

[21] B. Recht and C. Ré, *Parallel stochastic gradient algorithms for large-scale matrix completion*, Mathematical Programming Computation, 5 (2013), pp. 201–226.

[22] A. Ruszczyński and A. Shapiro, *Optimization of convex risk functions*, Mathematics of Operations Research, 31 (2006), pp. 433–452.

[23] M. Schmidt, N. Le Roux, and F. Bach, *Minimizing finite sums with the stochastic average gradient*, Mathematical Programming, 162 (2017), pp. 83–112.

[24] O. Shamir and T. Zhang, *Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes*, in International Conference on Machine Learning, 2013, pp. 71–79.

[25] W. W. Sun, X. Qiao, and G. Cheng, *Stabilized nearest neighbor classifier and its statistical properties*, Journal of the American Statistical Association, 111 (2016), pp. 1254–1265.

[26] M. Wang, E. X. Fang, and H. Liu, *Stochastic compositional gradient descent: Algorithms for minimizing compositions of expected-value functions*, Mathematical Programming, 161 (2017), pp. 419–449.

[27] M. Wang and J. Liu, *A stochastic compositional gradient method using markov samples*, in Proceedings of the 2016 Winter Simulation Conference, IEEE Press, 2016, pp. 702–713.

[28] M. Wang, J. Liu, and E. X. Fang, *Accelerating stochastic composition optimization*, in Advances in Neural Information Processing Systems, 2016, pp. 1714–1722.

[29] S. Wiesler, A. Richard, R. Schluter, and H. Ney, *Mean-normalized stochastic gradient for large-scale deep learning*, in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, IEEE, 2014, pp. 180–184.

[30] L. Xiao and T. Zhang, *A proximal stochastic gradient method with progressive variance reduction*, SIAM Journal on Optimization, 24 (2014), pp. 2057–2075.