Revisiting Stochastic Multi-Level Compositional Optimization

Wei Jiang, Sifan Yang, Yibo Wang, Tianbao Yang, *Senior Member, IEEE*, and Lijun Zhang, *Senior Member, IEEE*

***Abstract*—This paper explores stochastic multi-level compositional optimization, where the objective function is a composition of multiple smooth functions. Traditional methods for solving this problem suffer from either sub-optimal sample complexities or require huge batch sizes. To address these limitations, we introduce the Stochastic Multi-level Variance Reduction (SMVR) method. In the expectation case, our SMVR method attains the optimal sample complexity of $\mathcal{O}(1/\epsilon^3)$ to find an ϵ-stationary point for non-convex objectives. When the function satisfies convexity or the Polyak-Łojasiewicz (PL) condition, we propose a stage-wise SMVR variant. This variant improves the sample complexity to $\mathcal{O}(1/\epsilon^2)$ for convex functions and $\mathcal{O}(1/(\mu\epsilon))$ for functions meeting the μ-PL condition or μ-strong convexity. These complexities match the lower bounds not only in terms of ϵ but also in terms of μ (for PL or strongly convex functions), without relying on large batch sizes in each iteration. Furthermore, in the finite-sum case, we develop the SMVR-FS algorithm, which can achieve a complexity of $\mathcal{O}(\sqrt{n}/\epsilon^2)$ for non-convex objectives, $\mathcal{O}(\sqrt{n}/\epsilon\log(1/\epsilon))$ for convex functions and $\mathcal{O}(\sqrt{n}/\mu\log(1/\epsilon))$ for objectives satisfying the μ-PL condition, where n denotes the number of functions in each level. To make use of adaptive learning rates, we propose the Adaptive SMVR method, which maintains the same complexities while demonstrating faster convergence in practice.**

***Index Terms*—Stochastic compositional optimization, multi-level optimization, nested variance reduction, finite-sum optimization.**

I. Introduction

THIS paper investigates the stochastic multi-level compositional optimization problem, formulated as:

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) = f_K \circ \cdots \circ f_2 \circ f_1(\mathbf{w}), \tag{1}$$

where $f_i : \mathbb{R}^{d_{i-1}} \mapsto \mathbb{R}^{d_i}$, for $i = 1, \ldots, K$ (with $d_K = 1$ and $d_0 = d$). In the expectation case, we assume that only noisy evaluations of each layer function $f_i(\cdot;\xi_i)$ and its gradient $\nabla f_i(\cdot;\xi_i)$ can be accessed, where ξ_i denotes a sample drawn from the oracle such that:

$$\mathbb{E}_{\xi_i}\left[f_i(\cdot;\xi_i)\right] = f_i(\cdot), \quad \mathbb{E}_{\xi_i}\left[\nabla f_i(\cdot;\xi_i)\right] = \nabla f_i(\cdot).$$

In machine learning, $\mathbf{w}$ often represents the parameters of a predictive model, and F denotes the loss of that model, with ξ_i representing a training sample. A special case to be considered separately is when each ξ_i has finite support $\{1, \ldots, n_i\}$ and is uniformly distributed. In such finite-sum cases, the problem is expressed as:

$$\min_{\mathbf{w}\in\mathbb{R}^d} F(\mathbf{w}) = \frac{1}{n_K}\sum_{j=1}^{n_K} f_{K,j}\left(\cdots \frac{1}{n_1}\sum_{j=1}^{n_1} f_{1,j}\left(\mathbf{w}\right)\cdots\right). \tag{2}$$

Problems (1) and (2) have significant applications in many tasks, such as reinforcement learning [1], robust learning [2], multi-step model-agnostic meta-learning [3], risk-averse portfolio optimization [4], [5] and risk management [6], [7].

Our goal is to solve the problem with the optimal sample complexity, which is a commonly used metric in stochastic optimization. This metric measures the number of samples needed to reach an ϵ-stationary point for non-convex functions, i.e., $\|\nabla F(\mathbf{w})\| \le \epsilon$, or an ϵ-optimal point for convex or PL functions, i.e., $F(\mathbf{w}) - \inf_{\mathbf{w}} F(\mathbf{w}) \le \epsilon$. Problems (1) and (2) reduce to the standard one-level stochastic optimization problem when $K = 1$, and are known as the two-level compositional optimization for $K = 2$.

In the expectation case for one-level and two-level non-convex problems, single-loop algorithms such as STORM [8] and RECOVER [9] have been shown to achieve the optimal complexity of $\mathcal{O}(1/\epsilon^3)$ for attaining an ϵ-stationary solution without using large batch sizes. However, for multi-level problems, the errors of gradient and function value estimators accumulate with the level becoming deeper, making the problem much harder. Existing multi-level methods either suffer from sub-optimal complexities [10], [11], [12] or require huge and increasing batch sizes [13]. When the objective function is convex or strongly convex, [14] has been established a sample complexity of $\mathcal{O}(1/\epsilon^2)$ or $\mathcal{O}(1/(\mu^2\epsilon))$. However, their analysis requires that each layer function f_i is monotone and convex, and their complexity for μ-strongly convex function is non-optimal with respect to μ [15].

Received 19 July 2024; revised 26 February 2025; accepted 13 March 2025. Date of publication 18 March 2025; date of current version 6 June 2025. This work was supported in part by NSFC under Grant U23A20382, and in part by the Collaborative Innovation Center of Novel Software Technology and Industrialization. Recommended for acceptance by S. Gould. *(Corresponding author: Lijun Zhang.)*

Wei Jiang is with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: jiangw@lamda.nju.edu.cn).

Sifan Yang, Yibo Wang, and Lijun Zhang are with the National Key Laboratory for Novel Software Technology, School of Artiicial Intelligence, Nanjing University, Nanjing 210023, China (e-mail: yangsf@lamda.nju.edu.cn; wangyb@lamda.nju.edu.cn; zhanglj@lamda.nju.edu.cn).

Tianbao Yang is with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: tianbao-yang@tamu.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2025.3552197, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2025.3552197

In the finite-sum case, only one paper [13] addresses the multi-level finite-sum problem, achieving an optimal complexity of $\mathcal{O}(n+\sqrt{n}/\epsilon^2)$ for non-convex functions. Yet, this approach also requires large batch sizes of $\mathcal{O}(\sqrt{n})$ per step and full batch sizes of $\mathcal{O}(n)$ at each checkpoint. Moreover, the complexities for convex/PL/strongly convex objectives have not been explored in this context.

Hence, a fundamental question to be addressed is: *Is it possible to solve stochastic multi-level problems with optimal complexities for non-convex, convex, and strongly convex functions without large batch sizes?* We give an affirmative answer to this question by introducing an innovative algorithm named Stochastic Multi-level Variance Reduction (SMVR). By using the variance reduction techniques to estimate Jacobians and function values at each level, SMVR achieves the optimal sample complexity of $\mathcal{O}(1/\epsilon^3)$ for non-convex functions in the expectation case, aligning with the established lower bound [16]. Central to the algorithmic design and analysis are: (i) the variance reduction is applied concurrently to Jacobians and function values, which is different from most existing works [10], [11], [12]; (ii) the Jacobian estimators are updated with a projection to ensure that errors of gradient estimators can be bounded regardless of the depth of the problem. Then, by only estimating the overall gradient in each step and using the normalization technique, we further show that we are able to remove the projection operation and do not require problem-dependent parameters to set hyper-parameters, which are more practical to use in real-world scenarios. When handling convex functions or those satisfy the μ-PL condition (weaker than strong convexity), we propose stage-wise SMVR methods and improve the complexity to $\mathcal{O}(1/\epsilon^2)$ and $\mathcal{O}(1/(\mu\epsilon))$ respectively, matching the corresponding lower bounds [15]. A crucial aspect of our analysis is demonstrating that the errors of gradient and function value estimators decrease in a stage-wise manner.

For the finite-sum structure, we adopt the framework of SAG algorithm [17] to avoid computing on the full batches and incorporate variance reduction technique with an additional term recording the gradient history. By employing such a design, we obtain an optimal complexity of $\mathcal{O}(\sqrt{n}/\epsilon^2)$ for non-convex objectives. Following the similar stage-wise approach, we can also achieve a complexity of $\mathcal{O}(\sqrt{n}/\epsilon\log(1/\epsilon))$ for convex functions and $\mathcal{O}(\sqrt{n}/\mu\log(1/\epsilon))$ for μ-PL/strongly convex objectives. Finally, to take advantage of adaptive learning rates, we design an adaptive version of SMVR methods and prove the same rates. Adaptive SMVR performs better in practice and avoids tuning the learning rate manually. Compared with existing multi-level methods, this paper enjoys the following advantages:

1) We achieve the optimal complexity of $\mathcal{O}(1/\epsilon^3)$ for non-convex functions, which is better than existing multi-level methods [10], [11], [12]. Although [13] attains the same rate, their approach relies on a large and increasing batch size of $\mathcal{O}(1/\epsilon)$, which is impractical to use.
2) For convex and strongly convex functions, we obtain an optimal complexity of $\mathcal{O}(1/\epsilon^2)$ and $\mathcal{O}(1/(\mu\epsilon))$, respectively. This is an improvement over [14], as our method does not require each layer function f_i to be monotone and convex, and exhibits better dependence on μ for μ-strongly convex functions.
3) We introduce the Adaptive SMVR method to make use of adaptive learning rates, which enjoys the same complexity but shows faster convergence in practice.

A preliminary version of this paper was presented at the 39th International Conference on Machine Learning in 2022 [18]. In this paper, we have significantly expanded the conference version by adding the following extensions.

1) We develop a simpler version of the original SMVR algorithm, named SMVR-NP, which preserves optimal convergence but does not need the projection operation anymore. This is achieved by estimating the overall gradient instead of evaluating the gradient in each level separately. By further employing the normalization technique, we also avoid requiring problem-dependent parameters to set hyper-parameters, making the newly proposed method much more practical.
2) We also investigate the stochastic multi-level optimization for the finite-sum structure and propose the SMVR-FS algorithm to obtain the optimal sample complexity of $\mathcal{O}(\sqrt{n}/\epsilon^2)$ for non-convex functions. Compared with [13], our method supports a constant batch size, which is much easier to implement. In contrast, [13] requires to use a large batch of $\mathcal{O}(\sqrt{n})$ in each step and of $\mathcal{O}(n)$ in the checkpoint step.
3) We further improve the complexity to $\mathcal{O}(\sqrt{n}/\epsilon\log(1/\epsilon))$ for convex functions and to $\mathcal{O}(\sqrt{n}/\mu\log(1/\epsilon))$ for μ-PL/strongly convex objectives in the finite-sum case. These results are new in the multi-level finite-sum literature, and the linear convergence rate $\mathcal{O}(\log(1/\epsilon))$ is optimal under the PL condition, matching the current result in the single-level finite-sum problem [19].
4) We compare the newly proposed methods, i.e., SMVR-NP and SMVR-FS, with other multi-level algorithms in the experiments of three different tasks, validating the effectiveness of our proposed methods.

A comparison between our results and existing multi-level methods is shown in Tables I and II.

II. Related Work

This section provides an overview of related work on stochastic two-level and multi-level compositional optimization, as well as finite-sum compositional optimization.

A. Two-Level Compositional Optimization

Ref. [20] first introduces the stochastic compositional gradient descent (SCGD) method to minimize a composition of two-level expected-value functions. This method uses two step size sequences in different time scales to update the decision variable and inner function separately. When the inner function is smooth, this approach yields a complexity of $\mathcal{O}(1/\epsilon^7)$ for non-convex objectives, $\mathcal{O}(1/\epsilon^{3.5})$ for convex functions, and $\mathcal{O}(1/(\mu^{14/4}\epsilon^{5/4}))$ for μ-strongly convex functions. In a subsequent work [21], the accelerated stochastic compositional proximal gradient (ASC-PG) is proposed to improve the complexity to $\mathcal{O}(1/\epsilon^{4.5})$, $\mathcal{O}(1/\epsilon^2)$ and $\mathcal{O}(1/\epsilon)$ for non-convex, convex and strongly convex functions, respectively.

TABLE I
SUMMARY OF RESULTS FOR ATTAINING AN ϵ-STATIONARY OR ϵ-OPTIMAL POINT IN THE EXPECTATION CASE

Method	Assumptions	Complexity	Batch size
A-TSCGD [10]	Smooth	$\mathcal{O}\left(1/\epsilon^{(7+K)/2}\right)$	$\mathcal{O}\left(1\right)$
A-TSCGD [10]	Smooth + SC	$\mathcal{O}\left(1/\epsilon^{(3+K)/4}\right)$	$\mathcal{O}\left(1\right)$
NLASG [11]	Smooth	$\mathcal{O}\left(1/\epsilon^{4}\right)$	$\mathcal{O}\left(1\right)$
SCSC [12]	Smooth	$\mathcal{O}\left(1/\epsilon^{4}\right)$	$\mathcal{O}\left(1\right)$
Nested-SPIDER [13]	Smooth	$\mathcal{O}\left(1/\epsilon^{3}\right)$	$\mathcal{O}\left(1/\epsilon\right)$
SSD [14]	Smooth + Mono. & CVX	$\mathcal{O}\left(1/\epsilon^{2}\right)$	$\mathcal{O}\left(1\right)$
SSD [14]	Smooth + SC	$\mathcal{O}\left(1/(\mu^2\epsilon)\right)$	$\mathcal{O}\left(1\right)$
SMVR/SMVR-NP (This work)	Smooth	$\mathcal{O}\left(1/\epsilon^{3}\right)$	$\mathcal{O}\left(1\right)$
Stage-wise SMVR (This work)	Smooth + CVX	$\mathcal{O}\left(1/\epsilon^{2}\right)$	$\mathcal{O}\left(1\right)$
Stage-wise SMVR (This work)	Smooth + PL	$\mathcal{O}\left(1/(\mu\epsilon)\right)$	$\mathcal{O}\left(1\right)$

Notations: CVX for convex, Mono. & CVX indicating each layer function is monotone and convex, SC for μ-strongly convex, and PL for the μ-PL condition (weaker than μ-strongly convex).

TABLE II
SUMMARY OF RESULTS FOR FINDING AN ϵ-STATIONARY OR ϵ-OPTIMAL POINT IN THE FINITE-SUM CASE

Method	Assumptions	Complexity	Batch size
Nested-SPIDER [13]	Smooth	$\mathcal{O}\left(\sqrt{n}/\epsilon^{2}\right)$	$\mathcal{O}\left(\sqrt{n}\right)$
SMVR-FS (This work)	Smooth	$\mathcal{O}\left(\sqrt{n}/\epsilon^{2}\right)$	$\mathcal{O}\left(1\right)$
Stage-wise SMVR-FS (This work)	Smooth + CVX	$\mathcal{O}\left(\sqrt{n}/\epsilon\log\left(1/\epsilon\right)\right)$	$\mathcal{O}\left(1\right)$
Stage-wise SMVR-FS (This work)	Smooth + PL	$\mathcal{O}\left(\sqrt{n}/\mu\log\left(1/\epsilon\right)\right)$	$\mathcal{O}\left(1\right)$

Instead of using two-timescale step sizes, a single-timescale method called Nested Averaged Stochastic Approximation (NASA) has been developed by [22] which achieves a complexity of $\mathcal{O}(1/\epsilon^4)$ for non-convex objectives. With the emergence of variance reduction techniques in one-level stochastic optimization such as SARAH [23], SPIDER [24], SpiderBoost [25] and STORM [8], variance reduced algorithms are also developed for two-level compositional problems with improved rates under a slightly stronger smoothness assumption [26], [27], [28], [29]. Notably, [27] and [28] achieve the optimal $\mathcal{O}(1/\epsilon^3)$ sample complexity, leveraging SARAH and SPIDER with large batch sizes, respectively. Later, [30] develops an algorithm named STORM-Compositional, attaining the same complexity using mini-batches. To avoid using batches, [9] proposes a STORM-based method and obtains the same optimal rate. However, these two-level approaches are not directly extendable to multi-level optimization problems.

B. Multi-Level Compositional Optimization

The pioneering work by [10] marks the beginning of research into stochastic multi-level optimization. They introduced an accelerated T-level stochastic compositional gradient descent (A-TSCGD) algorithm, which, through an extrapolation-interpolation technique, achieved a sample complexity of $\mathcal{O}(1/\epsilon^{(7+K)/2})$ for K-level problems. This complexity is further improved to $\mathcal{O}(1/\epsilon^{(3+K)/4})$ for strongly convex functions. Building on this, [11] proposes the Nested Linearized Averaging Stochastic Gradient method (NLASG), extending the NASA algorithm to a more general $K \geq 1$ setting, achieving a sample complexity of $\mathcal{O}(1/\epsilon^4)$. Concurrently, [12] develops the Stochastically Corrected Stochastic Compositional gradient method (SCSC), which adopts a STORM-like technique for function value estimation at each level, also achieving a sample complexity of $\mathcal{O}(1/\epsilon^4)$.

Later, [13] introduces the Nested-SPIDER method, which employs nested variance reduction for gradient approximation, improving the sample complexity to $\mathcal{O}(1/\epsilon^3)$. However, this method necessitates a large and increasing batch size at the order of $\mathcal{O}(1/\epsilon)$ and even $\mathcal{O}(1/\epsilon^2)$ in the first iteration of each stage. The method also does not specify complexities for convex and strongly convex functions. Later, [14] proves that the sample complexity can be improved to $\mathcal{O}(1/\epsilon^2)$ when every layer function f_i is monotone and convex, using a general Stochastic Sequential Dual (SSD) method. The complexity is further reduced to $\mathcal{O}(1/(\mu^2\epsilon))$ for μ-strongly convex functions. However, their method requires strong assumptions, i.e., layer-wise convexity and monotonicity. In contrast, our method only requires the overall objective function to be convex or strongly convex to achieve the same complexity for convex functions and an even better complexity for strongly convex functions.

More recently, multi-level optimization is also widely investigated in the distributed environment. [31] further introduced the decentralized stochastic multi-level optimization algorithm, which achieves the level-independent convergence rate under the decentralized setting. At the same time, [32] studied distributed multi-level optimization with the smooth and strongly convex objective, attaining an optimal communication complexity while maintaining an almost optimal sample complexity.

C. Finite-Sum Compositional Optimization

For the two-level finite-sum optimization problem in the form of $\frac{1}{n_2}\sum_{j=1}^{n_2} f_{2,j}(\frac{1}{n_1}\sum_{j=1}^{n_1} f_{1,j}(\mathbf{w}))$, [33] first combines the SCGD [20] and SVRG [34] techniques and achieve a complexity of $\mathcal{O}((n_1+n_2)\log(1/\epsilon))$ for strongly convex functions. To deal with the general non-convex objectives, [26] proposes an algorithm named VRSC-PG, which can obtain a complexity of $\mathcal{O}((n_1+n_2)^{2/3}/\epsilon^2)$ by employing the variance reduction technique to estimate the inner function values. This rate is also

achieved by [28] using a composite randomized incremental gradient method.

When it comes to the multi-level finite-sum optimization, [13] obtains a sample complexity of $\mathcal{O}(n + \sqrt{n_{\max}}/\epsilon^2)$, where $n_{\max} = \max\{n_1, \ldots, n_K\}$ and $n = \sum_{i=1}^{K} n_i$. Since this method is based on SPIDER, it still has to use large batch sizes of $\mathcal{O}(\sqrt{n_{\max}})$ and require computing over the full batches at certain checkpoint steps. Furthermore, the sample complexity for convex/PL/strongly convex functions remains unexplored in this setting, highlighting an opportunity for future research to investigate these specific function types within the multi-level finite-sum framework.

III. Multi-Level Variance Reduction for the Expectation Case

We first discuss the main challenge in solving multi-level compositional optimization problems. Then, we develop an optimal method for non-convex objectives in the expectation case. Finally, we explore additional conditions to further improve the sample complexity.

A. Notations and Assumptions

Let ξ denote some random variable and $\|\cdot\|$ denote the euclidean norm of a vector. We use Π_{L_f} to represent the projection onto the ball with radius L_f, i.e.,

$$\Pi_{L_f}(\mathbf{x}) = \underset{\|\mathbf{w}\| \leqslant L_f}{\operatorname{argmin}} \|\mathbf{w} - \mathbf{x}\|^2.$$

We further give the definition of sample complexity below.

Definition 1: The sample complexity refers to the number of samples needed to find a point satisfying $\mathbb{E}[\|\nabla F(\mathbf{w})\|] \leq \epsilon$ (ϵ-stationary) or $\mathbb{E}[F(\mathbf{w}) - \inf_{\mathbf{w}} F(\mathbf{w})] \leq \epsilon$ (ϵ-optimal).

Moreover, we make the following assumptions in this section, which are commonly adopted in the studies of stochastic compositional optimization [13], [20], [21], [27], [28], [35], [36].

Assumption 1: (Bounded Variance) For $1 \leq i \leq K$, the following conditions hold:

$$\begin{aligned}
\mathbb{E}_{\xi_t^i}\left[f_i(\mathbf{x}; \xi_t^i)\right] &= f_i(\mathbf{x}), \\
\mathbb{E}_{\xi_t^i}\left[\nabla f_i(\mathbf{x}; \xi_t^i)\right] &= \nabla f_i(\mathbf{x}), \\
\mathbb{E}_{\xi_t^i}\left[\left\|f_i(\mathbf{x}; \xi_t^i) - f_i(\mathbf{x})\right\|^2\right] &\leq \sigma_f^2, \\
\mathbb{E}_{\xi_t^i}\left[\left\|\nabla f_i(\mathbf{x}; \xi_t^i) - \nabla f_i(\mathbf{x})\right\|^2\right] &\leq \sigma_J^2,
\end{aligned}$$

where $\{\xi_t^i\}_{i=1}^K$ are mutually independent.

Assumption 2: (Mean-Squared Smoothness)

$$\begin{aligned}
\mathbb{E}_{\xi_t^i}\left[\left\|f_i(\mathbf{x}; \xi_t^i) - f_i(\mathbf{y}; \xi_t^i)\right\|^2\right] &\leq L_f^2 \|\mathbf{x} - \mathbf{y}\|^2, \\
\mathbb{E}_{\xi_t^i}\left[\left\|\nabla f_i(\mathbf{x}; \xi_t^i) - \nabla f_i(\mathbf{y}; \xi_t^i)\right\|^2\right] &\leq L_J^2 \|\mathbf{x} - \mathbf{y}\|^2.
\end{aligned}$$

Assumption 3: $F_* = \inf_{\mathbf{w}} F(\mathbf{w}) \geq -\infty$ and $F(\mathbf{w}_1) - F_* \leq \Delta_F$ for the initial solution $\mathbf{w}_1$.

Remark: Note that Assumptions 1 and 2 can imply that the overall objective function F is L_F-smooth, where the smooth constant is defined as $L_F := L_f^{2K-1} L_J \sum_{i=1}^{K} L_f^{-i}$.

B. The Challenge in Multi-Level Optimization

Compared with single-level problems, the main dilemma in multi-level optimization lies in the difficulty of obtaining an unbiased gradient of the function F. Consider a two-level compositional problem, where the objective function is expressed as $F(\mathbf{w}) = f \circ g(\mathbf{w})$. The gradient of this function is given by:

$$\nabla F(\mathbf{w}) = \nabla g(\mathbf{w}) \cdot \nabla f(g(\mathbf{w})).$$

Although we have access to unbiased estimations of each layer function and its gradient, i.e., $\mathbb{E}_{\xi_1}[g(\mathbf{x}; \xi_1)] = g(\mathbf{x})$, $\mathbb{E}_{\xi_2}[f(\mathbf{x}; \xi_2)] = f(\mathbf{x})$ and $\mathbb{E}_{\xi_2}[\nabla f(\mathbf{x}; \xi_2)] = \nabla f(\mathbf{x})$, it is still challenging to obtain an unbiased estimation of the gradient $\nabla f(g(\mathbf{w}))$. This is because the expectation over ξ_1 cannot be moved inside of ∇f such that:

$$\mathbb{E}_{\xi_1, \xi_2}\left[\nabla f(g(\mathbf{w}; \xi_1); \xi_2)\right] \neq \nabla f(g(\mathbf{w})).$$

Similarly, it is also difficult to obtain an unbiased estimation of the function value:

$$\mathbb{E}_{\xi_1, \xi_2}\left[f(g(\mathbf{w}; \xi_1); \xi_2)\right] \neq f(g(\mathbf{w})).$$

These challenges motivate us to adopt the variance reduced estimator to have a better evaluation of function values and Jacobians at each level, ensuring that the estimation errors can be reduced over time.

However, variance reduced estimators used in two-level optimization problems [9] can not be applied to multi-level directly, because the error might blow up as the depth increases if the estimators of Jacobians are not bounded. To handle this issue, [13] proposes to use an extremely small step size and periodically re-evaluate the function values and Jacobians at all levels with a large batch size after several iterations. However, this approach inevitably necessitates the use of large batches (as large as $\mathcal{O}(1/\epsilon^2)$) at the beginning of each stage, and since they use SPIDER [24] as their estimator, their method requires a batch size of $\mathcal{O}(1/\epsilon)$ at other iterations. To avoid using large batches, our method uses STORM [8] estimator and projects gradients onto a ball to ensure the Jacobians can be well bounded so that the error of the gradient estimator does not blow up.

C. Stochastic Multi-Level Variance Reduction Method

Now, we introduce the proposed Stochastic Multi-level Variance Reduction (SMVR) method for solving problem (1). As mentioned before, the main difficulty is that we can not obtain an unbiased estimation of the gradients and inner function values in the multi-level setting. We note that, in the one-level problems, the STORM method employs a momentum-based variance reduction technique for gradient estimation, represented as:

$$\begin{aligned}
\mathbf{d}_t = (1 - \beta_t)\mathbf{d}_{t-1} + \beta_t \nabla f\left(\mathbf{x}_t; \xi_t\right) \\
+ (1 - \beta_t)\left(\nabla f\left(\mathbf{x}_t; \xi_t\right) - \nabla f\left(\mathbf{x}_{t-1}; \xi_t\right)\right).
\end{aligned}$$

This method effectively reduces the variance of the estimated values and achieves the optimal rate. Inspired by STORM, we apply similar variance reduction estimators at each level to approximate the gradient more accurately.

The proposed method is described in Algorithm 1. At each time step t, we employ two sequences, $\mathbf{u}_t^i$ and $\mathbf{v}_t^i$, to estimate

Algorithm 1: SMVR Method.

Input: time step T, initial points $(\mathbf{w}_1, \mathbf{u}_1, \mathbf{v}_1)$, parameter c, and learning rate sequence $\{\eta_t\}$
for time step $t = 1$ **to** T **do**
  Set $\mathbf{u}_t^0 = \mathbf{w}_t$, $\beta_t = c\eta_{t-1}^2$
  for level $i = 1$ **to** K **do**
    Sample ξ_t^i
    Compute the function estimator $\mathbf{u}_t^i$ according to (3)
    Compute the Jacobian estimator $\mathbf{v}_t^i$ according to (4)
  end for
  Update gradient estimation: $\mathbf{v}_t = \prod_{i=1}^K \mathbf{v}_t^i$
  Update the decision variable: $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{v}_t$
end for
Choose τ uniformly at random from $\{1, \ldots, T\}$
Return $(\mathbf{w}_\tau, \mathbf{u}_\tau, \mathbf{v}_\tau)$

Algorithm 2: SMVR-NP.

Input: time step T, initial points $(\mathbf{w}_1, \mathbf{u}_1, \mathbf{v}_1)$, parameter c, and learning rate sequence $\{\eta_t\}$
for time step $t = 1$ **to** T **do**
  Set $\mathbf{u}_t^0 = \mathbf{w}_t$
  for level $i = 1$ **to** K **do**
    Sample ξ_t^i
    Compute the function estimator $\mathbf{u}_t^i$ according to (3)
  end for
  Compute the gradient estimator $\mathbf{v}_t$ according to (5)
  Update the decision variable: $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{v}_t$
end for
Choose τ uniformly at random from $\{1, \ldots, T\}$
Return $(\mathbf{w}_\tau, \mathbf{u}_\tau, \mathbf{v}_\tau)$

the function value and the gradient at level i, respectively. For function value estimation, we use a nested STORM estimator, i.e.,

$$\mathbf{u}_t^i = (1-\beta_t)\mathbf{u}_{t-1}^i + \beta_t f_i(\mathbf{u}_t^{i-1}; \xi_t^i) + (1-\beta_t)\left(f_i(\mathbf{u}_t^{i-1}; \xi_t^i) - f_i(\mathbf{u}_{t-1}^{i-1}; \xi_t^i)\right). \quad (3)$$

This formulation can be interpreted as that $\mathbf{u}_t^i$ is a STORM estimator of $f_i(\mathbf{u}_t^{i-1})$. For estimating the Jacobians, we apply a nested STORM estimator, followed by a projection:

$$\mathbf{v}_t^i = \Pi_{L_f}\left[(1-\beta_t)\mathbf{v}_{t-1}^i + \beta_t \nabla f_i(\mathbf{u}_t^{i-1}; \xi_t^i) + (1-\beta_t)\left(\nabla f_i(\mathbf{u}_t^{i-1}; \xi_t^i) - \nabla f_i(\mathbf{u}_{t-1}^{i-1}; \xi_t^i)\right)\right]. \quad (4)$$

The projection operation ensures that the error of the stochastic gradient estimator can be bounded; otherwise, they may blow up as the level becomes deeper. Note that $\mathbf{v}_t^i$ tracks the value of $\nabla f_i(\mathbf{u}_t^{i-1})$, and the overall gradient estimation error can be bounded as:

$$\begin{aligned}
&\left\|\prod_{i=1}^K \nabla f_i(\mathbf{u}_t^{i-1}) - \prod_{i=1}^K \mathbf{v}_t^i\right\|^2 \\
&\leq K\left\|\prod_{i=1}^K \nabla f_i(\mathbf{u}_t^{i-1}) - \mathbf{v}_t^1 \prod_{i=2}^K \nabla f_i(\mathbf{u}_t^{i-1})\right\|^2 \\
&\quad + \ldots + K\left\|\prod_{i=1}^{K-1} \mathbf{v}_t^i \cdot \nabla f_K(\mathbf{u}_t^{K-1}) - \prod_{i=1}^K \mathbf{v}_t^i\right\|^2 \\
&\leq K\left(\sum_{i=1}^K L_f^{2(K-1)} \left\|\nabla f_i(\mathbf{u}_t^{i-1}) - \mathbf{v}_t^i\right\|^2\right),
\end{aligned}$$

where the last inequality holds since $\mathbf{v}_t^i$ is bounded by L_f. That is to say, on the one hand, we aim to leverage the benefits of variance reduction in the estimator (we require that the true gradients are in the projected domain and thus projection does not hinder the analysis); on the other hand, we do not want the variance of estimator accumulates too fast over multiple levels ($\mathbf{v}_i$ is bounded after projection). Hence, projection on the Jacobian estimator is a perfect solution. Once the gradient at each level is evaluated, we apply the chain rule to calculate the estimated gradient of the objective function, i.e., $\mathbf{v}_t = \mathbf{v}_t^1 \mathbf{v}_t^2 \cdots \mathbf{v}_t^K$ and employ gradient descent to update the variable $\mathbf{w}_t$ at the end of each time step.

Note that in the first iteration of our algorithm, we evaluate the function value and gradient at each level simply as $\mathbf{u}_1^i = f(\mathbf{u}_1^{i-1}; \xi_1^i)$ and $\mathbf{v}_1^i = \nabla f_i(\mathbf{u}_1^{i-1}; \xi_1^i)$. Our algorithm does not need to use large batches in any iterations, though it is fully compatible with mini-batch techniques. Here, ξ_t^i within the algorithm can represent either a single training sample or a batch of samples. Next, we present the sample complexity of our method.

Theorem 1: If we set $c = 10L_1^2$, $\eta_t = (a+t)^{-1/3}/L_1$ and $a = (20L_1^3)^{3/2}$, where $L_1 = \mathcal{O}(KL_F)$ is a positive constant, our Algorithm 1 ensures that $\mathbb{E}[\|\nabla F(\mathbf{w}_\tau)\|] \leq \mathcal{O}(\frac{KL_F}{T^{1/3}})$.

Remark: The complexity of our approach is on the order of $\mathcal{O}(1/\epsilon^3)$, which matches the lower bound in one-level setting [16]. The SMVR method avoids using large batches in each iteration, which is more practical to implement compared with the existing method which requires huge batch sizes and changing the batch size over time [13].

D. Stochastic Multi-Level Variance Reduction Without Projection Operation

In the previous subsection, we introduce a projection operation to prevent the estimation error from escalating with the level becoming deeper. This is necessary because the SMVR method separately estimates the gradient at each level, and then combines them using the chain rule. As a result, if each $\mathbf{v}_t^i$ is unbounded, it becomes challenging to decompose the overall error of the whole gradient estimator $\mathbf{v}_t$ as the errors in each level, resulting the error blowing up. Therefore, the projection is crucial for bounding the gradient estimator at each level. However, implementing the projection (as well as setting hyper-parameters in SMVR) requires the knowledge of the upper bound of the gradient at each level, which is often hard to know in practice.

To overcome this limitation, we propose an alternative approach that only estimates the overall gradient at each time step. Specifically, rather than evaluating the gradient at each

Algorithm 3: Stage-Wise SMVR.

1: **Input:** initial points $(\mathbf{w}_0, \mathbf{u}_0, \mathbf{v}_0)$, parameter c
2: **for** stage $s = 1$ **to** S **do**
3: Set η_s and T_s according to Lemma 1
4: $\mathbf{w}_s, \mathbf{u}_s, \mathbf{v}_s =$ SMVR (with T_s, $(\mathbf{w}_{s-1}, \mathbf{u}_{s-1}, \mathbf{v}_{s-1})$, c, η_s)
5: **end for**
6: Return $\mathbf{w}_S$

level as $\mathbf{v}_t^i$ through the STORM estimator and then multiplying these evaluations as $\mathbf{v}_t = \prod_{i=1}^K \mathbf{v}_t^i$, we directly apply variance reduction estimation on the overall gradient. That is to say, we update the overall gradient estimator $\mathbf{v}_t$ as:

$$\mathbf{v}_t = (1-\beta_t)\mathbf{v}_{t-1} + \beta_t \prod_{i=1}^K \nabla f_i(\mathbf{u}_t^{i-1}; \xi_t^i) + (1-\beta_t)\left(\prod_{i=1}^K \nabla f_i(\mathbf{u}_t^{i-1}; \xi_t^i) - \prod_{i=1}^K \nabla f_i(\mathbf{u}_{t-1}^{i-1}; \xi_t^i)\right). \tag{5}$$

With this modification, we eliminate the need for a projection operation while still ensuring that the error does not accumulate as the level becomes deeper. This is because we no longer multiply $\mathbf{v}_t^i$ together, thus avoiding the need to bound each $\mathbf{v}_t^i$ and compute its error in the analyses. Instead, we can analyze the overall gradient estimation error directly without decomposition. By further employing a normalization technique, we can also avoid requiring problem-dependent parameters such as $L_f, L_J, \sigma_f, \sigma_J$ to set hyper-parameters η_t and β_t for our algorithm. We present this revised approach in Algorithm 2, named SMVR-NP (SMVR with No Projection). We demonstrate that SMVR-NP achieves a similar optimal complexity as stated below.

Theorem 2: By setting $\eta_t = \frac{\eta}{\|\mathbf{v}_t\|}$ and $\eta = \beta_t = T^{-2/3}$, our Algorithm 2 can guarantee that $\mathbb{E}[\|\nabla F(\mathbf{w}_\tau)\|] \leq \mathcal{O}(\frac{KL_F}{T^{1/3}})$.

Remark: SMVR-NP maintains the same optimal complexity as the original SMVR, and it is more practical than the initial SMVR method since it does not need the projection operation in each level or require to know problem-dependent parameters to set hyper-parameters η_t and β_t. When $\|\mathbf{v}_t\| = 0$, we set $\mathbf{v}_t/\|\mathbf{v}_t\| = \mathbf{0}$ such that we do not update $\mathbf{w}_t$ in this case.

E. Faster Convergence Under Stronger Conditions

Next, we explore whether additional assumptions could be used to further improve the complexity of our approach. We develop a variant of our original method, named Stage-wise SMVR, which achieves better complexity when the objective function satisfies the PL condition or convexity.

The new algorithm is a multi-stage adaptation of the original SMVR method, summarized in Algorithm 3. Instead of decreasing the learning rate η_t polynomially, in Stage-wise SMVR, we decrease the learning rate η and the parameter β after each stage, while concurrently increasing the iteration number for each stage. At the end of each stage, the algorithm saves the output $\mathbf{w}_s, \mathbf{u}_s, \mathbf{v}_s$, which are used as starting points for the next stage. With these modifications, we can obtain a better convergence guarantee under the PL condition or dealing with convex objective functions.

First, we investigate the case that the objective function satisfies the PL condition, which is a commonly used condition in the literature [37], [38], [39], [40]. We introduce the definition of the PL condition below.

Definition 2: The function $F(\mathbf{w})$ satisfies the μ-PL condition if there exists a positive constant μ such that:

$$2\mu\left(F(\mathbf{w}) - F_*\right) \leq \|\nabla F(\mathbf{w})\|^2.$$

With this condition, we can prove that the error of function estimator $\mathbf{u}_s$ and gradient estimator $\mathbf{v}_s$ decreases after each stage.

Lemma 1: Define that $\epsilon_1 = \frac{8L_1}{\mu}$ and $\epsilon_s = \frac{\epsilon_1}{2^{s-1}}$. Then, by setting that $T_1 = \max\{4L_1K(\sigma_f^2 + \sigma_J^2), 2\sqrt{2L_1}\Delta_F\}$, $\beta_1 = \frac{1}{2L_1}$, $T_s = \max\{\frac{4L_2^{3/2}}{\mu\epsilon_{s-1}}, \frac{4L_2}{\mu^{3/2}\sqrt{\epsilon_{s-1}}}\}$, $\beta_s = \frac{\mu\epsilon_{s-1}}{L_2}$, $c = 16L_1^2$, $\eta_s = \sqrt{\beta_s/c}$ and $L_2 = 64L_1^2$, the output of Algorithm 3 satisfies:

$$\mathbb{E}\left[F\left(\mathbf{w}_s\right) - F_*\right] \leq \epsilon_s;$$

$$\sum_{i=1}^K \mathbb{E}\left[\left\|f_i(\mathbf{u}_s^{i-1}) - \mathbf{u}_s^i\right\|^2 + \left\|\mathbf{v}_s^i - \nabla f_i(\mathbf{u}_s^{i-1})\right\|^2\right] \leq \mu\epsilon_s.$$

This lemma indicates the objective gap $\mathbb{E}[F(\mathbf{w}_s) - F_*]$ is reduced by half after each stage. As a result, after $S = \log_2(2\epsilon_1/\epsilon)$ stages, the output satisfies $\mathbb{E}[F(\mathbf{w}_S) - F_*] \leq \epsilon$. Based on Lemma 1, we can establish the convergence of our method in the following theorem.

Theorem 3: Assume $F(\mathbf{w})$ satisfies the μ-PL condition. The Stage-wise SMVR algorithm achieves an ϵ-optimal point with a sample complexity of $\mathcal{O}(K^3L_F^3/(\mu\epsilon))$.

Moreover, if the objective function satisfies the convexity rather than the PL condition, our method can still use this property to improve the sample complexity, as indicated in the following theorem.

Theorem 4: Assume $F(\mathbf{w})$ is convex and the optimal solution is bounded by $\|x^*\| \leq D$. The proposed algorithm attains an ϵ-optimal point with a complexity of $\mathcal{O}(K^3L_F^3/\epsilon^2)$.

Remark: The Stage-wise SMVR method behaves optimally when the objective function enjoys the PL condition or convexity. For smooth and convex functions, our method aligns with the $\mathcal{O}(1/\epsilon^2)$ lower bound for this problem [15]. When it comes to the PL condition, there exists $\mathcal{O}(1/(\mu\epsilon))$ lower bound for the μ-strongly convex setting [15], which is a special case of the PL condition, thus proving our method is optimal. Compared with existing results [14], our analysis requires weaker assumptions and enjoys a better and optimal dependence in terms of μ.

IV. Multi-Level Variance Reduction for the Finite-Sum Structure

In this section, we investigate the case for the finite-sum structure, where the function in each level is in the form of the

finite-sum, i.e.,

$$\frac{1}{n_K}\sum_{j=1}^{n_K} f_{K,j}\left(\cdots \frac{1}{n_2}\sum_{j=1}^{n_2} f_{2,j}\left(\frac{1}{n_1}\sum_{j=1}^{n_1} f_{1,j}\left(\mathbf{w}\right)\right)\cdots\right).$$

In this case, we may have the chance to compute the exact gradient in certain iterations, as a result, we can obtain improved complexity in terms of ϵ. First, we introduce the following assumption in this section, which is also used in the previous multi-level finite-sum literature [13].

Assumption 4: Each function $f_{i,j}$ is L_f-Lipschitz continuous and its Jacobian $\nabla f_{i,j}$ is L_J-Lipschitz continuous.

It is well-known that the optimal sample complexity for non-convex objectives in the single-level finite-sum setting is $\mathcal{O}(n+\frac{\sqrt{n}}{\epsilon^2})$ [24]. To achieve this optimal complexity, a straightforward approach is to integrate the existing SVRG [34], [41] technique with our SMVR method. This strategy is also used in the previous multi-level finite-sum literature [13], which incorporates SVRG into the SPIDER algorithm. However, SVRG is a two-loop algorithm and requires computing the full version of the function value and the gradient periodically at "checkpoint steps", which is not practical in real-world scenarios.

To avoid this limitation, we propose a novel single-loop variance reduction technique for the finite-sum structures. In each time step t, for level i, we first sample i_t randomly from $\{1,\ldots,n_i\}$, and then we estimate the function value as:

$$\begin{aligned}\mathbf{u}_t^i &= (1-\beta)\mathbf{u}_{t-1}^i + \beta\mathbf{r}_t^i \\ &\quad + (1-\beta)\left[f_{i_t}(\mathbf{u}_t^{i-1}) - f_{i_t}(\mathbf{u}_{t-1}^{i-1})\right]. \end{aligned} \tag{6}$$

By setting $\mathbf{r}_t = f_{i_t}(\mathbf{u}_t^{i-1})$, this estimation reduces to our original SMVR method. To achieve the optimal rate in the finite-sum structure, we adopt the similar design of SAG method [17], and set $\mathbf{r}_t^i = f_{i_t}(\mathbf{u}_t^{i-1}) - h_t^{i_t} + \frac{1}{n_i}\sum_{j=1}^{n_i} h_t^j$, where h_t represents the historical record of past function values, updated as $h_{t+1}^j = \begin{cases} f_{i_t}(\mathbf{u}_t^{i-1}) & j = i_t \\ h_t^j & j \neq i_t \end{cases}$. This formulation is a combination of the SAG algorithm and STORM method, and effectively ensures that the estimation error decreases over time without using the checkpoint technique or the two-loop design. Similarly, we compute the gradient for each level as follows:

$$\begin{aligned}\mathbf{v}_t^i &= \Pi_{L_f}\left[(1-\beta)\mathbf{v}_{t-1}^i + \beta\mathbf{z}_t^i\right. \\ &\quad \left.+(1-\beta)\left(\nabla f_{i_t}(\mathbf{u}_t^{i-1}) - \nabla f_{i_t}(\mathbf{u}_{t-1}^{i-1})\right)\right], \end{aligned} \tag{7}$$

where $\mathbf{z}_t^i = \nabla f_{i_t}(\mathbf{u}_t^{i-1}) - g_{i,t}^{i_t} + \frac{1}{n_i}\sum_{j=1}^{n_i} g_{i,t}^j$ and we can set that $g_{i,t+1}^j = \begin{cases} \nabla f_{i_t}(\mathbf{u}_t^{i-1}) & j = i_t \\ g_{i,t}^j & j \neq i_t \end{cases}$. Note that in each time step, we first calculate $\mathbf{v}_t^i$ and then update $g_{i,t}$, which helps to avoid the dependency issues in the analyses. Finally, we estimate the overall gradient by multiplying $\mathbf{v}_1\cdots\mathbf{v}_K$ together and apply gradient descent for updating. The whole algorithm, named SMVR-FS (SMVR for Finite-Sum structure), is summarized in Algorithm 4.

Next, we present the theoretical result for non-convex functions as follows:

Theorem 5: Setting $\eta = \mathcal{O}(1/\sqrt{n_{\max}})$ and $\beta = \mathcal{O}(1/n_{\max})$, our method can ensure that $\mathbb{E}[\|F(\mathbf{w}_\tau)\|] \leq \mathcal{O}(\frac{n_{\max}^{1/4}K^{1/4}L_F}{T^{1/2}})$.

Algorithm 4: SMVR-FS.

Input: time step T, initial points $(\mathbf{w}_1, \mathbf{u}_1, \mathbf{v}_1)$, parameter c, and learning rate sequence $\{\eta_t\}$
for time step $t = 1$ to T **do**
  Set $\mathbf{u}_t^0 = \mathbf{x}$
  for level $i = 1$ to K **do**
    random select i_t from $\{1,\ldots,n_i\}$
    Compute the function estimator $\mathbf{u}_t^i$ according to (6)
    Compute the gradient estimator $\mathbf{v}_t^i$ according to (7)
  end for
  Update gradient estimation: $\mathbf{v}_t = \Pi_{i=1}^K \mathbf{v}_t^i$
  Update the decision variable: $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\mathbf{v}_t$
end for
Choose τ uniformly at random from $\{1,\ldots,T\}$
Return $(\mathbf{w}_\tau, \mathbf{u}_\tau, \mathbf{v}_\tau)$

Remark: The complexity of SMVR-FS method is on the order of $\mathcal{O}(\sqrt{n_{\max}}/\epsilon^2)$, where $n_{\max} = \max\{n_1,\ldots,n_K\}$, matching the lower bound for the one-level setting [19]. Our single-loop method avoids using huge batches and checkpoint steps, which is more practical to implement compared with the existing method which requires large batch sizes and the use of checkpoints [13].

Remark: We also have to note that there is a trade-off between sample complexity and storage complexity. To obtain optimal sample compelxity, we borrow the idea from the SAG algorithm, and thus require storing past gradients estimators. This storage requirement is the common issue for SAG or SAGA type variance reduction methods.

Moreover, by adopting a similar stage-wise design as in Algorithm 3, but with constant values for η_s, β_s and T_s in each stage, we can achieve improved complexities for convex, PL, or strongly convex functions:

Theorem 6: Assuming that $F(\mathbf{w})$ satisfies the μ-PL condition or is μ-strongly convex, our stage-wise SMVR-FS algorithm can achieve an ϵ-optimal point with a sample complexity of $\mathcal{O}(\sqrt{n_{\max}K}L_F^2/\mu\log 1/\epsilon)$ by setting that $\eta_s = \mathcal{O}(1/\sqrt{n_{\max}})$, $\beta_s = \mathcal{O}(1/n_{\max})$ and $T_s = 4/\mu\eta_s$.

Theorem 7: Assuming that $F(\mathbf{w})$ is convex and the norm of the optimal solution x^* is bounded by $\|x^*\| \leq D$, our stage-wise SMVR-FS algorithm attains an ϵ-optimal point with a complexity of $\mathcal{O}(\sqrt{n_{\max}K}L_F^2/\epsilon\log(1/\epsilon))$.

Remark: We achieve linear convergence $\mathcal{O}(\log(1/\epsilon))$ for the PL condition, aligning with the current results for the single-level finite-sum problem [19]. It is also the first time that we obtain such complexities for convex, PL, or strongly convex objectives under the multi-level finite-sum setting.

V. Multi-Level Variance Reduction Method With Adaptive Learning Rates

In this section, we demonstrate that the proposed method can be effectively adapted to incorporate adaptive learning rates and maintain the same sample complexity. Adaptive learning rates are widely used in stochastic optimization problems, and many successful methods have been proposed, such as AdaGrad [42], Adam [43], AMSGrad [44], AdaBound [45], etc. Despite their

prevalence, their application in stochastic multi-level setting remains less explored. Inspired by the above methods, we introduce an adaptive version of our method, named Adaptive SMVR. To use adaptive learning rates, we modify the decision variable update step from $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{v}_t$ to:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{\eta_t}{\sqrt{\mathbf{h}_t} + \delta}\mathbf{v}_t, \tag{8}$$

where $\delta > 0$ is a parameter to prevent dividing by zero, and $\mathbf{h}_t$ can take following forms:

$$\begin{aligned}
&\text{AdaGrad -type:} && \mathbf{h}_t = \frac{1}{t}\sum_{i=1}^{t}\mathbf{v}_i^2 \\
&\text{Adam -type:} && \mathbf{h}_t = (1-\beta_t')\,\mathbf{h}_{t-1} + \beta_t'\mathbf{v}_t^2 \\
&\text{AMSGrad -type:} && \mathbf{h}_t' = (1-\beta_t')\,\mathbf{h}_{t-1}' + \beta_t'\mathbf{v}_t^2, \\
& && \mathbf{h}_t = \max\left(\mathbf{h}_{t-1}, \mathbf{h}_t'\right).
\end{aligned} \tag{9}$$

Inspired by the recent study of Adam-style methods [46], we establish the sample complexity of the Adaptive SMVR in Theorem 8 using similar analyses.

Theorem 8: By setting $c = 10L_3^2$, $\eta_t = (a+t)^{-1/3}/L_3$ and $a = (20L_3^3)^{3/2}$, Adaptive SMVR with learning rates defined in (8) and (9) can ensure that $\mathbb{E}[\|\nabla F(\mathbf{w}_\tau)\|] \leq \mathcal{O}(\frac{KL_F}{T^{1/3}})$, where L_3 is a constant indicated in the proof.

VI. Experiments

In this section, we conduct a series of numerical experiments to evaluate the performance of the proposed methods over three different tasks. We compare our method with existing multi-level algorithms, including A-TSCGD [10], NLASG [11], Nested-SPIDER [13] and SCSC [12]. For the SMVR method, hyper-parameters β_t and η_t are set up according to Theorem 1, and the parameter L_1 is searched from the set $\{0.5, 1, 5, 10\}$. When it comes to SMVR-FS, the parameter $n_{\max}$ is searched from the set $\{1e1, 1e2, 1e3, 1e4, 1e5\}$. For other methods, we choose the hyper-parameters recommended in their original papers or conduct a grid search to select the best hyper-parameters. As for the projection operation Π_{L_f}, we simply set L_f as a large value and provide a sensitivity analysis in terms of tuning L_f in the first experiment. All the curves in the experiment part are averaged over 20 runs.

A. Risk-Averse Portfolio Optimization

We first consider the risk-averse portfolio optimization problem. Suppose we have d assets to invest during each time step $\{1, \ldots, T\}$, and $r_t \in \mathbb{R}^d$ denotes the payoff of d assets in the time step t. The objective is to maximize investment returns and minimize the risk simultaneously. A useful formulation is the mean-deviation risk-averse optimization model [5], where the risk is defined as the standard deviation. This mean-deviation model is widely used in practice and often used for experimental validation in multi-level optimization research [10], [14]. The problem can be formulated as:

$$\max_{x\in\mathcal{X}} \frac{1}{T}\sum_{t=1}^{T}\langle r_t, x\rangle - \lambda\sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\langle r_t, x\rangle - \langle \bar{r}, x\rangle\right)^2},$$

where $\bar{r} = \sum_{t=1}^{T} r_t$, decision variable x denotes the investment quantity vector in d assets. Note that the domain $\mathcal{X}$ is a simplex, and we use a projection operation to ensure that the variable $\mathbf{x}$ is within the domain. The above problem is a three-level stochastic compositional optimization problem, and each layer can be represented as:

$$\begin{aligned}
f_1(x) &= \left(\frac{1}{T}\sum_{t=1}^{T}\langle r_t, x\rangle, x\right), \\
f_2(y, x) &= \left(y, \frac{1}{T}\sum_{t=1}^{T}\left(\langle r_t, x\rangle - y\right)^2\right), \\
f_3(z_1, z_2) &= -z_1 + \lambda\sqrt{z_2}.
\end{aligned}$$

In the experiment, we test different methods on real-world datasets Industry-10, Industry-12, Industry-17, and Industry-30 from Keneth R. French Data Library.[1] These datasets consist of 10, 12, 17, and 30 industrial assets payoff over 25105 consecutive periods, respectively. Following [13], we set the parameter $\lambda = 0.2$.

Fig. 1 shows a comparison of the loss values and the gradient norms against the number of samples drawn by each method. We can find that our methods (including SMVR, SMVR-NP, and SMVR-FS) converge much faster than other algorithms across all tasks. More specifically, both the loss and the gradient norms of SMVR and its variants show a more rapid decrease, demonstrating the low sample complexity of the proposed method.

We also conduct experiments to investigate the impact of tuning the parameter L_f for the projection operation Π_{L_f} in the SMVR method. For the theoretical analysis, setting L_f above the actual upper bound of the gradient should not alter the order of the convergence rate, although it may affect the size of the constant factor in the rate. Here, we adjust the L_f from the set $\{5, 10, 50, 100\}$, and the results are depicted in Fig. 2, where L_f = NA indicates that the projection operation is not used, equivalent to assigning L_f an extremely large value, such as $1e7$. We find that the method performs very closely as long as L_f is set as a large number and would perform worse when L_f is small. This finding suggests that in practical applications, setting L_f to a high value is a viable strategy.

B. Hierarchical Tilted Empirical Risk Minimization

Hierarchical Tilted Empirical Risk Minimization (TERM) is a method proposed by [2], [47], which can deal with noisy and imbalanced machine learning problems simultaneously. The TERM objective is given by $\widetilde{R}(w) := \frac{1}{t}\log(\frac{1}{N}\sum_{i\in[N]} e^{tl(w;z_i)})$, where $l(w; z_i)$ denotes the loss for sample z_i from data $\{z_1, \ldots, z_N\}$. It can mitigate outliers when $t < 0$ and handle class imbalance when $t > 0$. When the task

[1] https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/

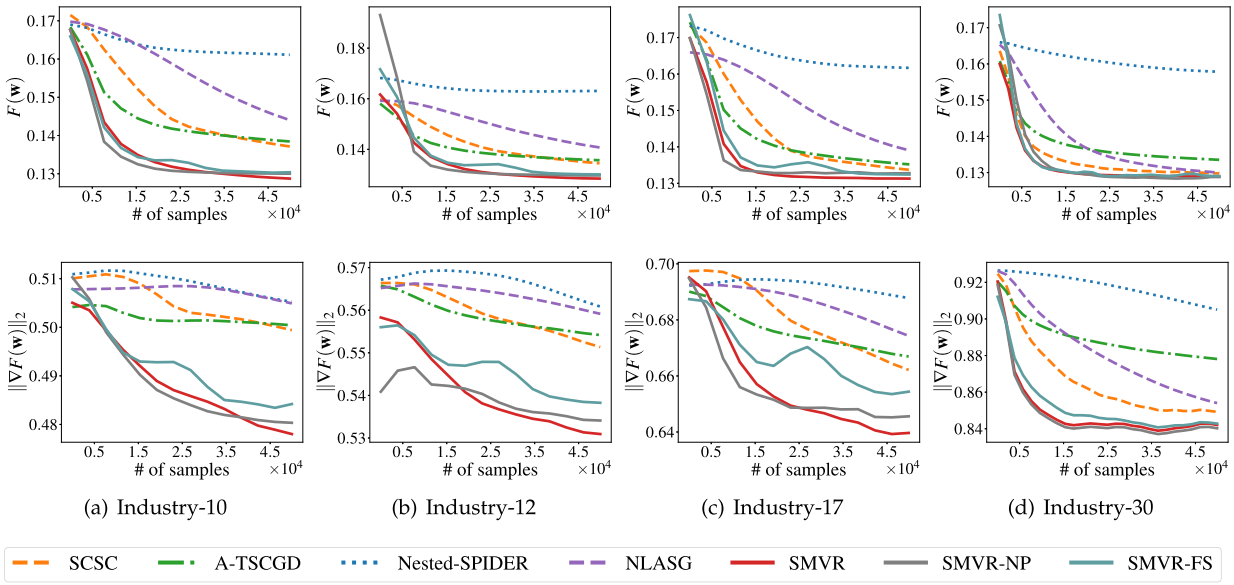

Fig. 1. Results for Risk-Averse Portfolio Optimization.

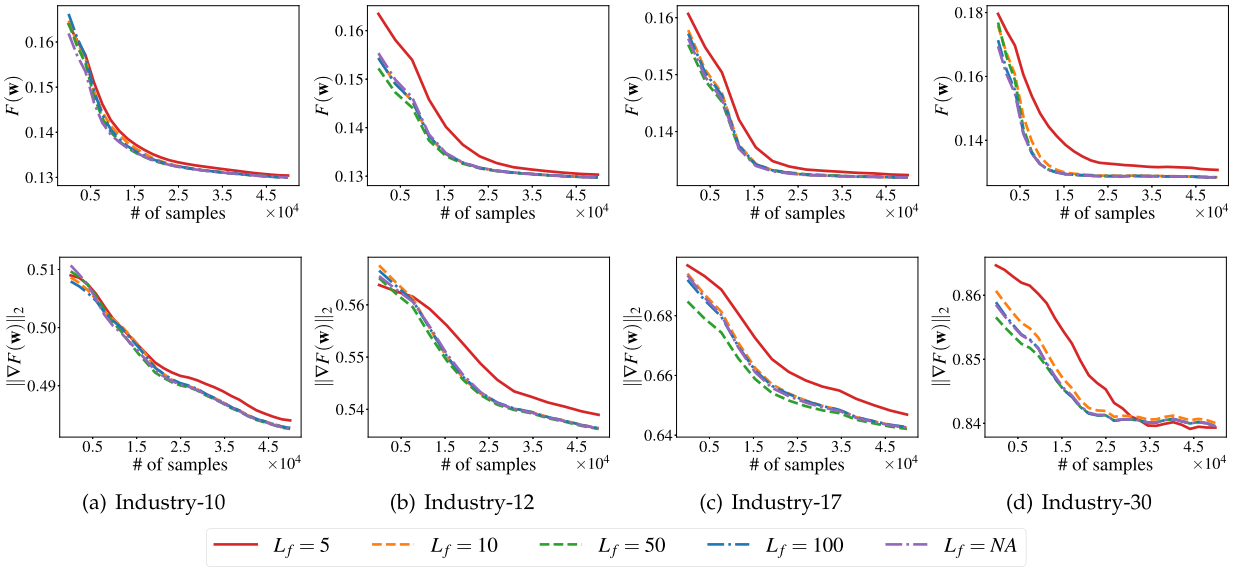

Fig. 2. Results for Risk-Averse Portfolio Optimization.

involves outliers and class imbalance at the same time, the Hierarchical TERM approach can be used:

$$\tilde{J}(w) := \frac{1}{t}\log\left(\frac{1}{|\mathcal{D}|}\sum_{\mathcal{G}\subseteq\mathcal{D}}|\mathcal{G}|e^{t\tilde{R}_{\mathcal{G}}(w)}\right),$$

$$\text{with } \widetilde{R}_{\mathcal{G}}(w) := \frac{1}{\tau}\log\left(\frac{1}{|\mathcal{G}|}\sum_{z\in\mathcal{G}}e^{\tau\ell(w;z)}\right),$$

where $\mathcal{D}$ represents all training samples and $\mathcal{G}$ denotes samples for one specific class. The parameters t and τ are constants dealing with different goals (i.e., outliers and class imbalance). This framework is a four-level stochastic compositional optimization, with each layer represented as:

$$f_1(w) = \frac{1}{|\mathcal{G}|}\sum_{z\in\mathcal{G}}e^{\tau\ell(w;z)},\ f_2(x) = \frac{1}{\tau}\log(x),$$

$$f_3(y) = \frac{1}{|\mathcal{D}|}\sum_{\mathcal{G}\subseteq\mathcal{D}}|\mathcal{G}|e^{ty},\ f_4(z) = \frac{1}{t}\log(z).$$

In the experiment, we use the "HIV-1",[2] "Australian"[3], "Breast-cancer"[3] and "svmguide1"[3] datasets, and make the training data noisy and imbalanced, where nearly 30% of the labels are

[2]https://archive.ics.uci.edu/ml/datasets.php
[3]https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

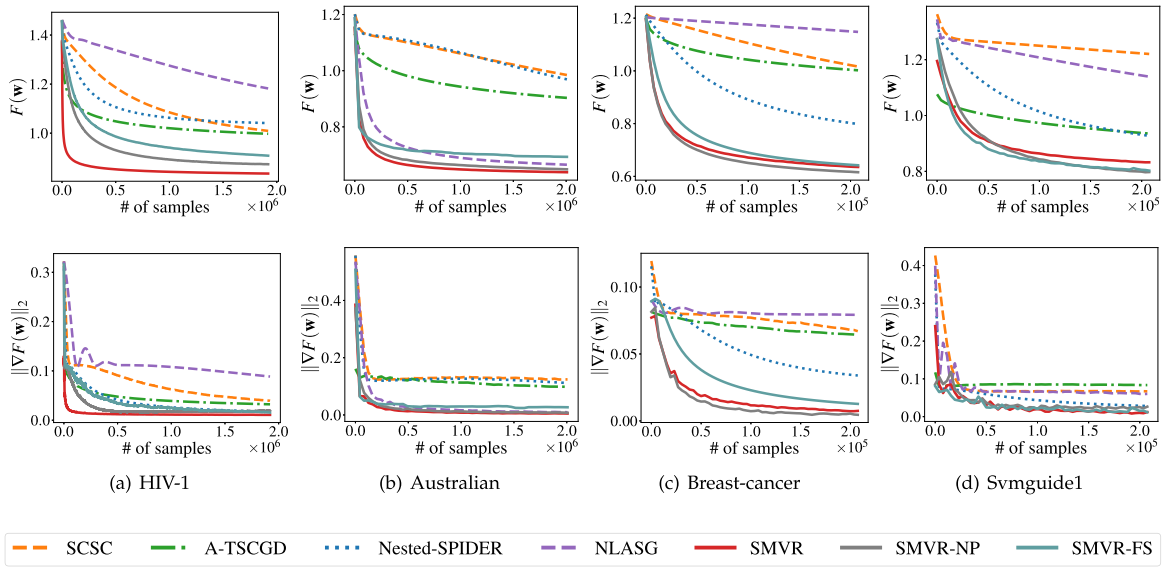

Fig. 3. Results for Hierarchical Tilted Empirical Risk Minimization.

TABLE III
CLASSIFICATION ACCURACIES (%) FOR HIERARCHICAL TILTED EMPIRICAL RISK MINIMIZATION

Method	Hiv-1		Australian scale		Breast-cancer		Svmguide1	
	rare	overall	rare	overall	rare	overall	rare	overall
A-TSCGD	71.3±2.9	88.1±1.2	62.7±8.9	72.5±5.2	55.1±8.5	80.6±4.4	76.5±4.6	85.8±2.1
SCSC	76.7±2.4	88.7±1.2	61.8±8.4	72.5±5.2	55.6±8.0	80.7±4.1	75.8±4.7	85.5±2.1
Nested-SPIDER	47.3±9.2	63.0±6.7	68.5±9.7	72.9±4.8	61.8±8.6	83.3±6.6	74.3±5.4	84.8±2.4
NLASG	69.9±3.0	88.1±1.3	79.4±8.4	78.7±5.8	42.0±8.5	76.7±5.2	73.3±5.2	85.0±2.2
SMVR-NP	77.4±2.8	89.2±1.1	80.2±7.8	80.5±4.3	67.6±5.4	85.5±3.8	78.9±2.6	86.1±2.2
SMVR-FS	78.9±2.6	88.1±2.0	79.5±6.8	79.7±3.5	67.8±7.6	85.2±2.8	80.8±3.4	86.8±1.8
SMVR	**79.3±2.1**	**89.9±1.1**	**83.2±8.0**	**82.7±4.5**	**72.8±8.1**	**86.8±4.2**	**81.2±2.7**	**87.8±2.1**

reshuffled and the number of rare class versus common class is 1:20. We set $\tau = -2$, $t = 10$ according to the origin paper and repeat each experiment 20 times.

As shown in Fig. 3, our methods perform best among all other algorithms. Both the loss value and the norm of the gradient converge more rapidly to a small value compared to other methods. We also report the classification accuracy in Table III. It shows that SMVR and its variants achieve the highest accuracy rates on the rare class and the overall task simultaneously, indicating the effectiveness of our methods.

C. Multi-Step Model-Agnostic Meta-Learning

Finally, we conduct experiments on Multi-step Model-Agnostic Meta-Learning (MAML). Multi-step MAML aims to find a good initialization point that performs well in different tasks after taking a few steps of gradient descent. Classical one-step MAML is formed as:

$$\min_{\mathbf{x}} F(\boldsymbol{\theta}) := \frac{1}{M}\sum_{m=1}^{M} F_m\left(\mathbf{x} - \alpha \nabla F_m(\mathbf{x})\right),$$
$$\text{with } F_m(\boldsymbol{\theta}) := \mathbb{E}_{\xi_m}\left[f\left(\boldsymbol{\theta}; \xi_m\right)\right],$$

where α is the learning rate, F_m denotes the loss for task m and ξ_m represents the training samples for task m. One-step MAML is a two-level problem, involving a single update to the initial point followed by evaluation across different tasks. In practice, it's common to update the initial point multiple times to enhance results, such as the five-step updates used by [48], which is a six-level compositional problem.

Following [48], we conduct experiments on 5-way 1-shot and 5-shot tasks on Omniglot dataset [49]. Each task is a 5-class classification problem, with only 1 or 5 training samples for each class. We conduct a 5-step MAML and report the accuracy of different methods against the number of training samples in Fig. 4. Since adaptive learning rates are widely used in neural networks, which are also applied in Multi-step MAML, we implement Adaptive SMVR methods in these tasks, denoted as SMVR-ADAM. We use the adaptive learning rate defined in (8) and (9) and choose the commonly used Adam-type. As can be seen, the accuracy of SMVR (and its variants), as well as SMVR-ADAM, increases rapidly in both training and testing sets, and outperforms other methods dramatically. Although SMVR and SMVR-ADAM enjoy the same sample complexity, SMVR-ADAM demonstrates faster convergence in practice due to the adaptive learning rate used.

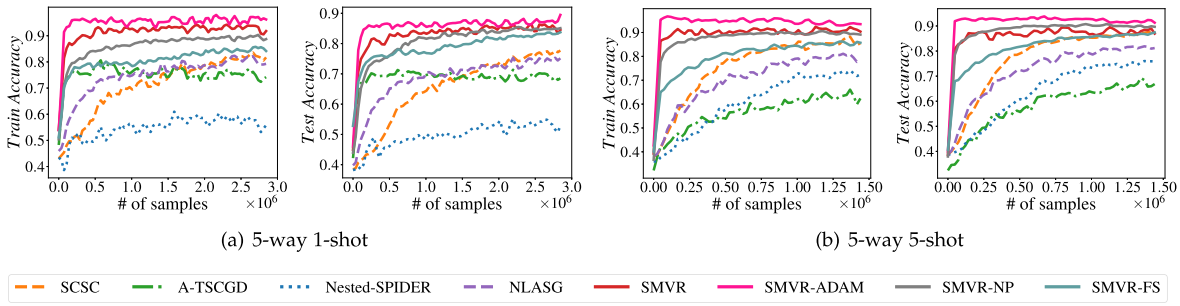

Fig. 4. Results for Multi-step Model-Agnostic Meta-Learning.

VII. Conclusion

In this paper, we propose an optimal algorithm named SMVR for stochastic multi-level composition optimization. We prove that the proposed algorithm, by using variance reduced estimator of function values and Jacobians, coupled with a projection operation, achieves a sample complexity of $\mathcal{O}(1/\epsilon^3)$ for finding an ϵ-stationary point. This complexity aligns with the lower bound even in the one-level setting, and our method avoids using batches in each iteration. Later, we demonstrate that by directly estimating the overall gradient and employing the normalization technique, we are able to remove the projection operation and avoid requiring problem-dependent parameters. When the objective function further satisfies the convexity or PL condition, we develop a stage-wise version of SMVR to obtain the optimal complexity of $\mathcal{O}(1/\epsilon^2)$ or $\mathcal{O}(1/\epsilon)$. For the finite-sum structure, we propose the SMVR-FS method. By utilizing the past gradients and function values, SMVR-FS attains the complexity of $\mathcal{O}(\sqrt{n}/\epsilon^2)$ for non-convex functions, $\mathcal{O}(\sqrt{n}/\epsilon \log(1/\epsilon))$ for convex functions, and $\mathcal{O}(\sqrt{n}/\mu \log(1/\epsilon))$ for μ-PL or μ-strongly convex functions. Finally, to take advantage of adaptive learning rates, we also propose Adaptive SMVR, which can achieve the same complexity with the learning rate changing adaptively. Experiments on three real-world tasks demonstrate the superiority of the proposed methods.

References

[1] C. Dann, G. Neumann, and J. Peters, "Policy evaluation with temporal differences: A survey and comparison," *J. Mach. Learn. Res.*, vol. 15, pp. 809–883, 2014.

[2] T. Li, A. Beirami, M. Sanjabi, and V. Smith, "On tilted losses in machine learning: Theory and applications," 2021, *arXiv:2109.06141*.

[3] K. Ji, J. Yang, and Y. Liang, "Multi-step model-agnostic meta-learning: Convergence and improved algorithms," 2020, *arXiv: 2002.07836*.

[4] S. Bruno, S. Ahmed, A. Shapiro, and A. Street, "Risk neutral and risk averse approaches to multistage renewable investment planning under uncertainty," *Eur. J. Oper. Res.*, vol. 250, no. 3, pp. 979–989, 2016.

[5] A. Shapiro, D. Dentcheva, and A. Ruszczynski, *Lectures on Stochastic Programming: Modeling and Theory*, 3rd ed. Philadelphia, PA, USA: SIAM, 2021.

[6] S. Cole, X. Giné, and J. Vickery, "How does risk management influence production decisions? Evidence from a field experiment," *Rev. Financial Stud.*, vol. 30, no. 6, pp. 1935–1970, 2017.

[7] D. Dentcheva, S. I. Penev, and A. Ruszczynski, "Statistical estimation of composite risk functionals and risk optimization problems," *Ann. Inst. Stat. Math.*, vol. 69, no. 4, pp. 737–760, 2017.

[8] A. Cutkosky and F. Orabona, "Momentum-based variance reduction in non-convex SGD," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 15210–15219.

[9] Q. Qi, Z. Guo, Y. Xu, R. Jin, and T. Yang, "An online method for a class of distributionally robust optimization with non-convex objectives," 2021, *arXiv: 2006.10138*.

[10] S. Yang, M. Wang, and E. X. Fang, "Multilevel stochastic gradient methods for nested composition optimization," *SIAM J. Optim.*, vol. 29, no. 1, pp. 616–659, 2019.

[11] K. Balasubramanian, S. Ghadimi, and A. Nguyen, "Stochastic multi-level composition optimization algorithms with level-independent convergence rates," 2021, *arXiv: 2008.10526*.

[12] T. Chen, Y. Sun, and W. Yin, "Solving stochastic compositional optimization is nearly as easy as solving stochastic optimization," *IEEE Trans. Signal Process.*, vol. 69, pp. 4937–4948, 2021.

[13] J. Zhang and L. Xiao, "Multilevel composite stochastic optimization via nested variance reduction," *SIAM J. Optim.*, vol. 31, no. 2, pp. 1131–1157, 2021.

[14] Z. Zhang and G. Lan, "Optimal algorithms for convex nested stochastic composite optimization," 2021, *arXiv: 2011.10076*.

[15] A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright, "Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3235–3249, May 2012.

[16] Y. Arjevani, Y. Carmon, J. C. Duchi, D. J. Foster, N. Srebro, and B. E. Woodworth, "Lower bounds for non-convex stochastic optimization," 2019, *arXiv: 1912.02365*.

[17] N. L. Roux, M. Schmidt, and F. R. Bach, "A stochastic gradient method with an exponential convergence rate for finite training sets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2672–2680.

[18] W. Jiang, B. Wang, Y. Wang, L. Zhang, and T. Yang, "Optimal algorithms for stochastic multi-level compositional optimization," in *Proc. 39th Int. Conf. Mach. Learn.*, 2022, pp. 10195–10216.

[19] Z. Li, H. Bao, X. Zhang, and P. Richtarik, "Page: A simple and optimal probabilistic gradient estimator for nonconvex optimization," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 6286–6295.

[20] M. Wang, E. X. Fang, and H. Liu, "Stochastic compositional gradient descent: Algorithms for minimizing compositions of expected-value functions," *Math. Program.*, vol. 161, no. 1-2, pp. 419–449, 2017.

[21] M. Wang, J. Liu, and E. X. Fang, "Accelerating stochastic composition optimization," *J. Mach. Learn. Res.*, vol. 18, pp. 105:1–105:23, 2017.

[22] S. Ghadimi, A. Ruszczynski, and M. Wang, "A single timescale stochastic approximation method for nested stochastic optimization," *SIAM J. Optim.*, vol. 30, no. 1, pp. 960–979, 2020.

[23] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takác, "Sarah: A novel method for machine learning problems using stochastic recursive gradient," 2017, *arXiv: 1703.00102*.

[24] C. Fang, C. J. Li, Z. Lin, and T. Zhang, "Spider: Near-optimal non-convex optimization via stochastic path integrated differential estimator," 2018, *arXiv: 1807.01695*.

[25] Z. Wang, K. Ji, Y. Zhou, Y. Liang, and V. Tarokh, "Spiderboost: A class of faster variance-reduced algorithms for nonconvex optimization," 2018, *arXiv: 1810.10690*.

[26] Z. Huo, B. Gu, J. Liu, and H. Huang, "Accelerated method for stochastic composition optimization with nonsmooth regularization," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3287–3294.

[27] H. Yuan, X. Lian, C. J. Li, J. Liu, and W. Hu, "Efficient smooth non-convex stochastic compositional optimization via stochastic recursive gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14905–14916.

[28] J. Zhang and L. Xiao, "A stochastic composite gradient method with incremental variance reduction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 9075–9085.
[29] L. Liu, J. Liu, and D. Tao, "Variance reduced methods for non-convex composition optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5813–5825, Sep. 2022.
[30] H. Yuan and W. Hu, "Stochastic recursive momentum method for non-convex compositional optimization," 2020, *arXiv: 2006.01688*.
[31] H. Gao, "Decentralized multi-level compositional optimization algorithms with level-independent convergence rate," in *Proc. 27th Int. Conf. Artif. Intell. Statist.*, 2024, pp. 4402–4410.
[32] S. Yang and F. Li, "A communication-efficient algorithm for federated multilevel stochastic compositional optimization," *IEEE Trans. Signal Process.*, vol. 72, pp. 2333–2347, 2024.
[33] X. Lian, M. Wang, and J. Liu, "Finite-sum composition optimization via variance reduced gradient descent," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1159–1167.
[34] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 315–323.
[35] W. Jiang, G. Li, Y. Wang, L. Zhang, and T. Yang, "Multi-block-single-probe variance reduced estimator for coupled compositional optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 32499–32511.
[36] W. Jiang, J. Qin, L. Wu, C. Chen, T. Yang, and L. Zhang, "Learning unnormalized statistical models via compositional optimization," in *Proc. 40th Int. Conf. Mach. Learn.*, 2023, pp. 15105–15124.
[37] Z. Charles and D. Papailiopoulos, "Stability and generalization of learning algorithms that converge to global optima," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 745–754.
[38] M. Nouiehed, M. Sanjabi, T. Huang, J. Lee, and M. Razaviyayn, "Solving a class of non-convex min-max games using iterative first order methods," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14905–14916.
[39] Y. Xie, X. Wu, and R. A. Ward, "Linear convergence of adaptive stochastic gradient descent," in *Proc. 23rd Int. Conf. Artif. Intell. Statist.*, 2020, pp. 1475–1485.
[40] S. Chewi, T. Maunu, P. Rigollet, and A. Stromme, "Gradient descent algorithms for Bures-Wasserstein barycenters," in *Proc. 33rd Conf. Learn. Theory*, 2020, pp. 1276–1304.
[41] L. Zhang, M. Mahdavi, and R. Jin, "Linear convergence with condition number independent access of full gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 980–988.
[42] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," in *Proc. 23rd Annu. Conf. Learn. Theory*, 2010, pp. 257–269.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, *arXiv:1412.6980*.
[44] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of Adam and beyond," 2019, *arXiv:1904.09237*.
[45] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive gradient methods with dynamic bound of learning rate," 2019, *arXiv:1902.09843*.
[46] Z. Guo, Y. Xu, W. Yin, R. Jin, and T. Yang, "On stochastic moving-average estimators for non-convex optimization," 2021, *arXiv:2104.14840*.
[47] T. Li, A. Beirami, M. Sanjabi, and V. Smith, "Tilted empirical risk minimization," 2021, *arXiv:2007.01162*.
[48] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.
[49] B. M. Lake, R. Salakhutdinov, J. Gross, and J. B. Tenenbaum, "One shot learning of simple visual concepts," in *Proc. Annu. Meeting Cogn. Sci. Soc.*, 2011, pp. 2568–2573.

Wei Jiang received the BE degree from the Department of Computer Science and Technology, Xi'an Jiaotong University, China, in 2019. He is currently working toward the PhD degree with the Department of Computer Science and Technology, Nanjing University, China. His research interests include machine learning and stochastic optimization.

Sifan Yang received the BE degree in artificial intelligence from Nanjing University, Nanjing, China, in 2023. He is currently working toward the MS degree with the School of Artificial Intelligence, Nanjing University. His research interests include machine learning and optimization.

Yibo Wang received the BE degree from the Department of Computer Science and Technology from Nanjing University, China, in 2021. He is currently working toward the PhD degree with the School of Artificial Intelligence, Nanjing University. His research interests include machine learning and optimization.

Tianbao Yang (Senior Member, IEEE) is an associate professor and Herbert H. Richardson Faculty fellow with the CSE department of Texas A&M University, where he directs the lab of Optimization for Machine learning and AI (OptMAI Lab). His research interests center around optimization, machine learning and AI with applications in computer vision, NLP, trustworthy AI and medicine. Before joining TAMU, he was an assistant professor and then tenured Dean's Excellence associate professor with the Computer Science Department, the University of Iowa from 2014 to 2022. Before that, he worked in Silicon Valley as Machine Learning researcher for two years with GE Research and NEC Labs.

Lijun Zhang (Senior Member, IEEE) received the BE and PhD degrees in software engineering and computer science from Zhejiang University, China, in 2007 and 2012, respectively. He is currently a professor with the School of Artificial Intelligence, Nanjing University, China. Prior to joining Nanjing University, he was a postdoctoral researcher with the Department of Computer Science and Engineering, Michigan State University, USA. His research interests include machine learning and optimization.