Machine Learning (2025) 114:42
https://doi.org/10.1007/s10994-024-06631-x

Optimal large-scale stochastic optimization of NDCG surrogates for deep learning

Zi-Hao Qiu[1] · Quanqi Hu[2] · Yongjian Zhong[3] · Wei-Wei Tu[4] · Lijun Zhang[1] · Tianbao Yang[2]

Received: 15 May 2023 / Revised: 18 August 2024 / Accepted: 9 December 2024

Abstract

In this paper, we introduce principled stochastic algorithms to efficiently optimize Normalized Discounted Cumulative Gain (NDCG) and its top-K variant for deep models. To this end, we first propose novel compositional and bilevel compositional objectives for optimizing NDCG and top-K NDCG, respectively. We then develop two stochastic algorithms to tackle these non-convex objectives, achieving an iteration complexity of $\mathcal{O}(\epsilon^{-4})$ for reaching an ϵ-stationary point. Our methods employ moving average estimators to track the crucial inner functions for gradient computation, effectively reducing approximation errors. Besides, we introduce practical strategies such as initial warm-up and stop-gradient techniques to enhance performance in deep learning. Despite the advancements, the iteration complexity of these two algorithms does not meet the optimal $\mathcal{O}(\epsilon^{-3})$ for smooth non-convex optimization. To address this issue, we incorporate variance reduction techniques in our framework to more finely estimate the key functions, design new algorithmic mechanisms for solving multiple lower-level problems with parallel speed-up, and propose two types of algorithms. The first type directly tracks these functions with the variance reduced estimators, while the

Editor: Lam M Nguyen.

✉ Zi-Hao Qiu
qiuzh@lamda.nju.edu.cn

Quanqi Hu
quanqi-hu@tamu.edu

Yongjian Zhong
yongjian-zhong@uiowa.edu

Wei-Wei Tu
tuweiwei@4paradigm.com

Lijun Zhang
zhanglj@lamda.nju.edu.cn

Tianbao Yang
tianbao-yang@tamu.edu

1 National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

2 Department of Computer Science and Engineering, Texas A&M University, College Station, USA

3 Department of Computer Science, The University of Iowa, Iowa City, USA

4 4Paradigm Inc., Beijing, China

Published online: 27 January 2025

second treats these functions as solutions to minimization problems and employs variance reduced estimators to construct gradient estimators for solving these problems. We manage to establish the optimal $\mathcal{O}(\epsilon^{-3})$ complexity for both types of algorithms. It is important to highlight that our algorithmic frameworks are versatile and can optimize a wide spectrum of metrics, including Precision@K/Recall@K, Average Precision (AP), mean Average Precision (mAP), and their top-K variants. We further present efficient stochastic algorithms for optimizing these metrics with convergence guarantees. We conduct comprehensive experiments on multiple ranking tasks to verify the effectiveness of our proposed algorithms, which consistently surpass existing strong baselines.

Keywords Stochastic optimization · NDCG · Non-convex optimization · Information retrieval

1 Introduction

NDCG is a key performance metric for learning to rank in information retrieval (Liu, 2011) and other machine learning tasks where ranking plays a critical role (Liu & Yang, 2008; Bhatia et al., 2015). In this paper, we utilize the terminology from information retrieval to discuss NDCG and our methods. For a given query q with n items, the ranking model assigns scores to each item, which are then sorted in descending order to create a ranked list. The NDCG score for q is computed by:

$$\text{NDCG}_q = \frac{1}{Z_q} \sum_{i=1}^{n} \frac{2^{y_i} - 1}{\log_2(1 + \text{r}(i))}, \tag{1}$$

where y_i denotes the relevance score of the i-th item, $\text{r}(i)$ indicates the rank of the i-th item in the ordered list, and Z_q is a normalization factor known as the Discounted Cumulative Gain (DCG) score (Järvelin & Kekäläinen, 2002) of the optimal ranking for q. The top-K NDCG is defined by summing over items whose ranks are in the top K positions of the ordered list. This measure is particularly relevant in real-world applications such as recommendation systems, where the objective typically involves selecting a small set of K items from a large pool (Cremonesi et al., 2010), making top-K NDCG a common choice in such scenarios.

Optimizing NDCG and its top-K variant poses several challenges. Firstly, ranking all n items is computationally intensive. Secondly, the rank operator is non-differentiable with respect to model parameters. To address the non-differentiability, surrogate functions have been developed to approximate NDCG (Taylor et al., 2008; Qin et al., 2010; Swezey et al., 2021; Pobrotyn & Bialobrzeski, 2021). However, to the best of our knowledge, the computational challenge of calculating the gradient of equation (1), which involves sorting n items, has not yet been addressed. All existing gradient-based methods have a complexity of $\mathcal{O}(nd)$ per-iteration, where d is the number of model parameters, which is prohibitive for deep learning tasks with big n and big d. A naive approach updates the parameters by the gradient over a mini-batch of samples, but given the complexity and non-convex nature of the NDCG surrogate, these methods do not reliably compute an unbiased stochastic gradient, thus lacking theoretical guarantees.

In this paper, we first propose stochastic algorithms with a per-iteration complexity of $\mathcal{O}(Bd)$, where B is the mini-batch size, for optimizing NDCG and its top-K variant. For NDCG, we formulate a novel *finite-sum coupled compositional optimization (FCCO)* problem. Then, inspired by a recent work on average precision maximization (Qi et al., 2021),

we develop an efficient algorithm named **SONG**. This algorithm utilizes moving average estimators to track the inner functions of the compositional objectives, enabling us to control the optimization error effectively. Unlike the SGD-style or Adam-style updates used by Qi et al. (2021) in their algorithm, we employ a simple yet effective momentum-style update, conduct a more detailed analysis, and establish an iteration complexity of $\mathcal{O}(\epsilon^{-4})$ for finding an ϵ-level stationary solution, which is better than that proved by Qi et al. (2021), i.e., $\mathcal{O}(\epsilon^{-5})$[1]. To optimize top-$K$ NDCG that involves a selection operator, we propose a novel *bilevel optimization* problem, which contains a lower-level problem for top-K selection of each query. Then we smooth the non-smooth functions in the selection operator, and propose an algorithm named **K-SONG** with the iteration complexity of $\mathcal{O}(\epsilon^{-4})$. The algorithm leverages recent advances in bilevel optimization (Guo et al., 2021a) but introduces novel algorithm design and proof techniques to enable parallel processing of multiple lower-level problems in top-K NDCG optimization, while establishing a convergence guarantee. To further improve the effectiveness of optimizing the NDCG surrogates, we implement two practical strategies: initial warm-up to find a good initial solution and stop gradient operator to *simplify* the optimization of the top-K NDCG surrogate.

Although SONG/K-SONG systematically maximize NDCG and its top-K variant, their convergence rates still exhibit a gap compared to the optimal rate for smooth non-convex optimization, i.e., $\mathcal{O}(\epsilon^{-3})$ (Arjevani et al., 2022). To bridge this gap, we propose two types of improved algorithms named **Faster SONG$^{\mathbf{v1}}$/K-SONG$^{\mathbf{v1}}$** and **Faster SONG$^{\mathbf{v2}}$/K-SONG$^{\mathbf{v2}}$**, which are able to estimate those crucial functions for gradient computation more accurately by using advanced variance reduction techniques in different ways. Specifically, Faster SONG$^{\text{v1}}$/K-SONG$^{\text{v1}}$ employ an advanced variance reduced estimator named MSVR (Jiang et al., 2022) to track the functions involving randomness from compositional structures, and the STORM estimators (Cutkosky & Orabona, 2019) to manage stochastic gradient variance. For Faster SONG$^{\text{v2}}$/K-SONG$^{\text{v2}}$, we further explore the idea of converting the estimation of inner functions into solving elaborated minimization problems, where the gradient estimators are updated using the previously mentioned advanced variance reduced estimators. However, when optimizing the complex bilevel problem of top-K NDCG, using these variance-reduced estimators alone is insufficient to achieve the optimal convergence rate. Therefore, we design a novel batch update mechanism for lower-level problems that not only precisely controls the estimation error of lower-level solutions but also achieves parallel speed-up, surpassing the similar algorithm by Guo et al. (2021a). We also establish the optimal convergence rate using new proof techniques. Moreover, our algorithmic frameworks are versatile for optimizing a broad spectrum of metrics such as Precision@K/Recall@K, Average Precision (AP), mean Average Precision (mAP), and their top-K variants. To demonstrate this, we design efficient and theoretically guaranteed stochastic algorithms for optimizing these metrics.

Experiments on recommender systems and learning to rank tasks reveal that SONG and K-SONG significantly outperform prior baseline methods, with further in-depth analyses affirming the effectiveness of our algorithmic designs. Additionally, the enhanced algorithms Faster SONG$^{\text{v1/v2}}$/K-SONG$^{\text{v1/v2}}$ demonstrate improvements over their original counterparts. We also observe that Faster SONG$^{\text{v2}}$/K-SONG$^{\text{v2}}$ typically outperform Faster SONG$^{\text{v1}}$/K-SONG$^{\text{v1}}$. Further, our experiments on graph classification tasks demonstrate that our algorithms effectively optimize Precision@K and top-K mAP, which

[1] From Theorem 1 in Qi et al. (2021), it is evident that reaching an ϵ-stationary point requires $\mathcal{O}\left(\frac{n_+^{1/5}}{T^{1/5}}\right)$ iterations, thus the iteration complexity is $\mathcal{O}(\epsilon^{-5})$.

highlights the versatility and efficacy of our optimization frameworks across a variety of deep learning tasks. The code for replicating these experiments is available at https://github.com/zhqiu/NDCG-Optimization. Our work on SONG and K-SONG has been published at ICML 2022 (Qiu et al., 2022), with the paper's novel contributions summarized as follows:

- To improve the convergence rates of SONG/K-SONG, we propose two types of novel algorithms that not only use advanced variance-reduced estimators but also incorporate innovative algorithm designs to achieve parallel speed-up in optimizing lower-level problems.
- We prove that our algorithms enjoy the optimal convergence rate. To achieve this, we tailor the existing variance-reduced optimization proof framework to our problem and develop novel techniques to control the errors of solving multiple lower-level problems.
- We show that our algorithmic frameworks can optimize a wide range of metrics, such as Precision/Recall@K, mAP, and top-K mAP. We develop provable stochastic algorithms for these metrics and validate them through experiments.
- We employ more baseline methods and datasets from various domains, where our experimental results not only confirm the effectiveness of our methods but also reveal the algorithms' intrinsic mechanisms.

2 Related work

2.1 Listwise LTR approaches

We mainly review the listwise learning to rank (LTR) (Liu, 2011) methods that are close to this work. The listwise approaches fall into three categories. The first category uses ranking metrics to re-weight instances during training. For example, LambdaRank algorithms (Burges et al., 2005a; Burges, 2010) compute a weight ΔNDCG by the NDCG difference when a pair of items in the list is swapped, and use it to re-weight the pair during training. These approaches are then generalized by LambdaLoss (Wang et al., 2018), which achieves the best performance in this family. Although these algorithms consider NDCG, their theoretical relations to NDCG remain ambiguous. Methods in the second category, e.g., ListNet (Cao et al., 2007), RankCosine (Qin et al., 2008), and ListMLE (Xia et al., 2008), define loss functions to optimize the agreement between predictions and ground truth rankings. However, optimizing these loss functions do not necessarily maximize NDCG. In addition, efficient stochastic algorithms for these losses are still underdeveloped. The third category directly optimizes ranking metrics, and mostly focuses on NDCG, as reviewed below.

2.2 NDCG optimization

Earlier works employ traditional optimization techniques, e.g., genetic algorithm (Yeh et al., 2007), boosting (Xu & Li, 2007; Valizadegan et al., 2009), and SVM framework (Chakrabarti et al., 2008). However, these methods are not scalable to big data. A popular approach approximates ranks in NDCG with smooth functions and then optimize the resulting surrogates. For example, SoftRank (Taylor et al., 2008) uses rank distributions to smooth NDCG, but it suffers from a high computational complexity of $\mathcal{O}(n^3)$. ApproxNDCG (Qin et al., 2010) approximates the rank function and the top-K selector for the top-K variant by a generalized sigmoid function. Thonet et al. (2022) introduce SmoothI, a novel differentiable approximation of the rank indicator function that can be applied to various ranking metrics. Recent efforts like

PiRank (Swezey et al., 2021) and NeuralNDCG (Pobrotyn & Bialobrzeski, 2021) propose smoothing NDCG by approximating the non-continuous sorting operator based on NeuralSort (Grover et al., 2019). However, their per-iteration complexities remain $\mathcal{O}(nd)$. Moreover, little attention has been paid to the convergence guarantees for optimizing these surrogates. Recently, we have formulated the NDCG and top-K NDCG optimization problems as FCCO and bilevel optimization problems, respectively, and introduced the first algorithms with an iteration complexity of $\mathcal{O}(\epsilon^{-4})$ (Qiu et al., 2022). In this paper, we propose novel algorithm designs and incorporate advanced variance-reduced estimators, resulting in algorithms that achieve state-of-the-art $\mathcal{O}(\epsilon^{-3})$ complexity with parallel speed-up.

2.3 Stochastic compositional optimization

The optimization of two-level compositional functions in the form of $\mathbb{E}_\xi[f(\mathbb{E}_\zeta[g(\mathbf{w};\zeta)];\xi)]$, where ξ and ζ are independent random variables, or its finite-sum variant has been studied extensively (Wang et al., 2017; Balasubramanian et al., 2022; Chen et al., 2021). In this paper, we formulate the surrogate of NDCG into a similar but more complicated *finite-sum couples compositional optimization (FCCO)* problem of the form $\mathbb{E}_\xi[f(\mathbb{E}_\zeta[g(\mathbf{w};\zeta,\xi))]$, where ξ and ζ are independent and the inner function $g(\mathbf{w};\zeta,\xi)$ also depends on the random variable ξ of the outer level. We borrow a technique from Qi et al. (2021) by using moving average estimators to track the inner functions and make the approximation error controllable, but establish a better complexity of $\mathcal{O}(\epsilon^{-4})$. Recently, Jiang et al. (2022) propose a novel variance reduced estimator named MSVR for FCCO and achieve a better complexity of $\mathcal{O}(\epsilon^{-3})$. To improve the convergence rate, we use the MSVR estimator within a complex *bilevel* optimization framework comprising numerous low-level problems to manage the estimation error of compositional functions, significantly complicating our analysis.

2.4 Stochastic bilevel optimization

Stochastic bilevel optimization (SBO) has a long history in the literature (Colson et al., 2007; Kunisch & Pock, 2013; Liu et al., 2020). Recent works focus on algorithms with provable convergences (Ghadimi & Wang, 2018; Hong et al., 2023; Chen et al., 2022). However, most of them do not explicitly consider the challenge of many lower-level problems. Guo et al. (2021a) consider SBO with many lower-level problems and develop a stochastic algorithm with a convergence guarantee. However, their algorithm is not applicable to our problem with a compositional objective and does not achieve a parallel speed-up when using a mini-batch of samples to estimate the gradients. In this study, we apply advanced variance-reduced estimators within a complex bilevel optimization framework that includes multiple lower-level problems. We introduce two novel types of algorithms, demonstrating that they not only achieve the optimal convergence rate but also achieve parallel speed-up by utilizing multiple queries and multiple items from sample queries.

2.5 Variance reduced methods

Variance reduction has emerged as an important technique for non-convex optimization problems, providing faster convergence upon stochastic gradient descent. Many stochastic variance-reduced gradient algorithms have been proposed and they have improved the convergence rate from $\mathcal{O}(\epsilon^{-4})$ to $\mathcal{O}(\epsilon^{-3})$ (Fang et al., 2018; Zhou et al., 2020). Despite the

improvement, these methods rely heavily on *giant batch size* to construct checkpoint gradients, which limits the use of these algorithms. Cutkosky and Orabona (2019) address this issue and present a new estimator called STORM, achieving the optimal convergence rate by a variant of the momentum term. Recently, inspired by STORM, Jiang et al. (2022) propose a variance reduced estimator called MSVR, which employs a similar update as STORM but with a customized error correction term for FCCO. In this paper, we integrate STORM and MSVR in a complicated bilevel optimization problem with a compositional objective and many lower-level problems, achieving both the optimal iteration complexity of $\mathcal{O}(\epsilon^{-3})$ and parallel speed-up.

3 Preliminaries

Let $\mathcal{Q}$ represent a query set of size N, with each query denoted by $q \in \mathcal{Q}$. For each query q, $\mathcal{S}_q$ is a set of N_q items (e.g., documents, movies) to be ranked. Each item $\mathbf{x}_i^q \in \mathcal{S}_q$ has an associated relevance score $y_i^q \in \mathbb{R}^+$, indicating the relevance between query q and item $\mathbf{x}_i^q$. Define $\mathcal{S}_q^+ \subseteq \mathcal{S}_q$ as the subset containing N_q^+ items relevant to q, characterized by non-zero relevance scores. The set of all relevant query-item (Q-I) pairs is denoted by $\mathcal{S} = \{(q, \mathbf{x}_i^q) : q \in \mathcal{Q}, \mathbf{x}_i^q \in \mathcal{S}_q^+\}$. Let $h_q(\mathbf{x}; \mathbf{w})$ represent the predictive function for an item $\mathbf{x}$ with respect to the query q, where $\mathbf{w} \in \mathbb{R}^d$ denotes the parameters (e.g., a deep neural network). Furthermore, let $\mathbb{I}(\cdot)$ denote the indicator function, and $(x)_+$ represent the function $\max\{x, 0\}$. Let

$$r(\mathbf{w}; \mathbf{x}, \mathcal{S}_q) = \sum_{\mathbf{x}' \in \mathcal{S}_q} \mathbb{I}(h_q(\mathbf{x}'; \mathbf{w}) - h_q(\mathbf{x}; \mathbf{w}) \geq 0)$$

denote the rank of $\mathbf{x}$ with respect to the set $\mathcal{S}_q$, where we simply ignore the tie.

According to the definition in (1), the averaged NDCG over all queries is

$$\text{NDCG} = \frac{1}{N} \sum_{q=1}^{N} \frac{1}{Z_q} \sum_{\mathbf{x}_i^q \in \mathcal{S}_q^+} \frac{2^{y_i^q} - 1}{\log_2(r(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q) + 1)},$$

where Z_q is the maximum DCG of the perfect ranking for the items in $\mathcal{S}_q$. An important variant of NDCG is its top-K variant, which is defined over the items $\mathbf{x}_i^q \in \mathcal{S}_q$ whose prediction scores are in the top-K positions, i.e.,

$$\text{Top-}K\text{NDCG} = \frac{1}{N} \sum_{q=1}^{N} \frac{1}{Z_q^K} \sum_{\mathbf{x}_i^q \in \mathcal{S}_q^+} \mathbb{I}(\mathbf{x}_i^q \in \mathcal{S}_q[K]) \frac{2^{y_i^q} - 1}{\log_2(r(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q) + 1)},$$

where $\mathcal{S}_q[K]$ represents the items within $\mathcal{S}_q$ whose prediction scores rank in the top-K positions, and Z_q^K denotes the top-K DCG score of the ideal ranking. It is worth mentioning that when K is set equal to $|\mathcal{S}_q|$ for each q, top-K NDCG simplifies to NDCG, thereby making **NDCG a special case of top-K NDCG**.

Finally, we introduce the concept of iteration complexity, a commonly used metric to assess the efficiency of stochastic algorithms.

Definition 1 A stochastic algorithm is said to achieve an **ϵ-stationary point** if $\mathbb{E}[\|\nabla F(\mathbf{x}_t)\|] \leq \epsilon$, where $\mathbf{x}_t$ is the algorithm output at the t-th iteration and the expectation is taken over the randomness of the algorithm until the iteration t. The **iteration complexity** of the algorithm is the number of iterations needed to find the ϵ-stationary point.

Algorithm 1 Stochastic Optimization of NDCG: SONG

Require: $\eta, \gamma_0, \beta_1, \mathbf{u}^1 = 0, \mathbf{m}_1 = 0$
Ensure: $\mathbf{w}_{T+1}$
1: **for** $t = 1, ...T$ **do**
2: Draw some relevant Q-I pairs $\mathcal{B} = \{(q, \mathbf{x}_i^q)\} \subset \mathcal{S}$
3: For each sampled q draw a batch of items $\mathcal{B}_q \subset \mathcal{S}_q$
4: **for** each sampled Q-I pair $(q, \mathbf{x}_i^q) \in \mathcal{B}$ **do**
5: Let $\hat{g}_{q,i}(\mathbf{w}_t) = \frac{1}{|\mathcal{B}_q|} \sum_{\mathbf{x}' \in \mathcal{B}_q} \ell(\mathbf{w}_t; \mathbf{x}', \mathbf{x}_i^q, q)$
6: Compute $\mathbf{u}_{q,i}^{t+1} = (1 - \gamma_0)\mathbf{u}_{q,i}^t + \gamma_0 \hat{g}_{q,i}(\mathbf{w}_t)$
7: **end for**
8: Compute the stochastic gradient estimator $G(\mathbf{w}_t)$ according to (5)
9: Compute $\mathbf{m}_{t+1} = \beta_1 \mathbf{m}_t + (1 - \beta_1)G(\mathbf{w}_t)$
10: update $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{m}_{t+1}$
11: **end for**

4 Optimizing a smooth NDCG surrogate

To address the non-differentiability of the rank function $r(\mathbf{w}; \mathbf{x}, \mathcal{S}_q)$, we approximate it by a continuous and differentiable surrogate function

$$\bar{g}(\mathbf{w}; \mathbf{x}, \mathcal{S}_q) = \sum_{\mathbf{x}' \in \mathcal{S}_q} \ell(h_q(\mathbf{x}'; \mathbf{w}) - h_q(\mathbf{x}; \mathbf{w})),$$

where $\ell(\cdot)$ is a surrogate loss of $\mathbb{I}(\cdot \geq 0)$. Here we use a convex and non-decreasing smooth surrogate loss, e.g., squared hinge loss $\ell(x) = \max(0, x + c)^2$, where c is a margin parameter. Below, we abuse the notation $\ell(\mathbf{w}; \mathbf{x}', \mathbf{x}, q) = \ell(h_q(\mathbf{x}'; \mathbf{w}) - h_q(\mathbf{x}; \mathbf{w}))$. Using the surrogate loss, we cast NDCG maximization into:

$$\max_{\mathbf{w} \in \mathbb{R}^d} L(\mathbf{w}) := \frac{1}{|\mathcal{S}|} \sum_{q=1}^{N} \sum_{\mathbf{x}_i^q \in \mathcal{S}_q^+} \frac{2^{y_i^q} - 1}{Z_q \log_2(\bar{g}(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q) + 1)}. \tag{2}$$

The following lemma justifies the maximization of $L(\mathbf{w})$ for NDCG maximization:

Lemma 1 *If $\ell(\mathbf{w}; \mathbf{x}', \mathbf{x}, q) \geq \mathbb{I}(h_q(\mathbf{x}'; \mathbf{w}) - h_q(\mathbf{x}; \mathbf{w}) \geq 0)$, $L(\mathbf{w})$ lower bounds NDCG.*

The key challenge for solving the above problem lies at (i) computing $\bar{g}(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ and its gradient is expensive when $|\mathcal{S}_q| = N_q$ is very large; and (ii) an unbiased stochastic gradient of the objective function is not readily available. To highlight the second challenge, let us consider the gradient of the function $\phi(\mathbf{w}) = \frac{1}{\log_2(\bar{g}(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q) + 1)}$, which is given by

$$\nabla \phi(\mathbf{w}) = \frac{-\log_2(e) \cdot \nabla \bar{g}(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)}{\log_2^2(\bar{g}(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q) + 1) \cdot (\bar{g}(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q) + 1)}.$$

We can estimate $\bar{g}(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ by its unbiased estimator using a mini-batch of items $\mathcal{B}_q \subset \mathcal{S}_q$, i.e., $\frac{N_q}{|\mathcal{B}_q|} \sum_{\mathbf{x}' \in \mathcal{B}_q} l(\mathbf{w}, \mathbf{x}', \mathbf{x}_i^q, q)$. However, directly plugging this unbiased estimator into the above expression produces a *biased* estimator of $\nabla \phi(\mathbf{w})$ because $\nabla \phi(\mathbf{w})$ is *non-linear* w.r.t. $\bar{g}$. The optimization error will be large if $|B_q|$ is small (Hu et al., 2020b).

To address this issue, we cast the problem into the following equivalent form:

$$\min_{\mathbf{w} \in \mathbb{R}^d} F(\mathbf{w}) := \frac{1}{|\mathcal{S}|} \sum_{(q, \mathbf{x}_i^q) \in \mathcal{S}} f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)), \tag{3}$$

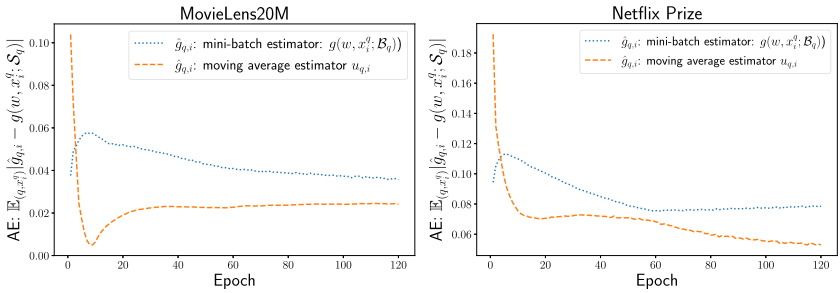

Fig. 1 Comparing the approximation error (AE) of the mini-batch estimator $g(\mathbf{w}, \mathbf{x}_i^q, \mathcal{B}_q)$ and moving average estimator $\mathbf{u}_{q,i}$ for the function $g(\mathbf{w}, \mathbf{x}_i^q, \mathcal{S}_q)$

where $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q) = \frac{1}{N_q}\bar{g}(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ and $f_{q,i}(g) = \frac{1}{Z_q}\frac{1-2^{y_i^q}}{\log_2(N_q g+1)}$. It is a special case of a family of **finite-sum coupled compositional stochastic optimization** problems, which was first studied by Qi et al. (2021) for maximizing average precision. Inspired by their method, we develop a stochastic algorithm for solving (3). The complete procedure is provided in Algorithm 1, which is named as Stochastic Optimization of NDCG (SONG).

To motivate the proposed method, we first derive the gradient of $F(\mathbf{w})$:

$$\nabla F(\mathbf{w}) = \frac{1}{|\mathcal{S}|}\sum_{(q,\mathbf{x}_i^q)\in\mathcal{S}} \nabla f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q))\nabla g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q). \tag{4}$$

The major cost for computing $\nabla F(\mathbf{w})$ lies at computing $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ and its gradient, which involves all items in $\mathcal{S}_q$. To this end, we approximate these quantities by stochastic samples. The gradient $\nabla g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ can be approximated by the stochastic gradient $\nabla \hat{g}_{q,i}(\mathbf{w}) = \frac{1}{|\mathcal{B}_q|}\sum_{\mathbf{x}'\in\mathcal{B}_q} \nabla \ell(\mathbf{w}; \mathbf{x}', \mathbf{x}_i^q, q)$, where $\mathcal{B}_q$ is sampled from $\mathcal{S}_q$. Note that $\nabla f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q))$ is non-linear w.r.t. $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$, thus we need a better way to estimate $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ to control the approximation error.

To this end, we borrow a technique from Qi et al. (2021) by using a **moving average estimator** to track $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ for each $\mathbf{x}_i^q \in \mathcal{S}_q^+$. Specifically, we maintain a scalar $\mathbf{u}_{q,i}$ for each *relevant* query-item pair $(q, \mathbf{x}_i^q)$ and update it by a linear combination of historical one $\mathbf{u}_{q,i}^t$ and an unbiased estimator of $g(\mathbf{w}_t; \mathbf{x}_i^q, \mathcal{S}_q)$ denoted by $\hat{g}_{q,i}(\mathbf{w}_t)$ in Step 5 and 6 in Algorithm 1, where $\gamma_0 \in (0, 1)$ is a parameter. Intuitively, when t increases, $\mathbf{w}_{t-1}$ is getting closer to $\mathbf{w}_t$, hence the previous value of the estimator, i.e., $\mathbf{u}_{q,i}^t$ is useful for estimating $g_{q,i}(\mathbf{w}_t)$.

To examine the effectiveness of the moving average estimator, we conduct an experiment on two popular recommender system datasets: MovieLens20M (Harper & Konstan, 2015) and Netflix Prize (Bennett et al., 2007). We employ the widely used NeuMF model (He et al., 2017), which is trained by calculating $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ across $\mathcal{S}_q$ and minimizing the objective in (3). Further details are provided in Sect. 9.2. During this process, we also compute $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$'s mini-batch estimator $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{B}_q)$ and moving average estimator $\mathbf{u}_{q,i}$, comparing their estimation errors relative to $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$, which is defined as $\mathbb{E}_{(q,\mathbf{x}_i^q)}|\hat{g}_{q,i} - g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)|$. The results, illustrated in Fig. 1, demonstrate that the moving average estimator consistently provides a more accurate estimation of $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$.

With these stochastic estimators, we can compute the gradient of the objective in (3) with controllable error as

$$G(\mathbf{w}_t) = \frac{1}{|\mathcal{B}|} \sum_{(q,\mathbf{x}_i^q)\in\mathcal{B}} \nabla f_{q,i}(\mathbf{u}_{q,i}^t) \nabla \hat{g}_{q,i}(\mathbf{w}_t). \tag{5}$$

We implement the momentum update for $\mathbf{w}_{t+1}$ in Step 9 and 10, where $\beta_1 \in (0, 1)$ is the momentum parameter. The momentum update can be replaced by the Adam-style update (Guo et al., 2021b), where the step size η is replaced by an adaptive step size. We can establish the same convergence rate for the Adam-style update.

We have some remarks about SONG: (i) for $\mathbf{w}_1$, since the NDCG surrogate function is non-convex, we use the initial warm-up strategy described in Sect. 6 to find good initialization parameters for all NDCG optimization algorithms; (ii) the per-iteration complexity of SONG is $\mathcal{O}(Bd+B^2)$, where $\mathcal{O}(Bd)$ and $\mathcal{O}(B^2)$ come from the forward and backward computation of $h_q(\mathbf{x}_i^q; \mathbf{w})$ and $\hat{g}_{q,i}(\mathbf{w})$, $\mathbf{x}_i^q \in \mathcal{B}_q$, respectively. For a large model size $d \gg B$, we have the per-iteration complexity of $\mathcal{O}(Bd)$, which is independent of the length of $\mathcal{S}_q$; and (iii) the additional memory cost is the size of $\mathbf{u}_{q,i}$, i.e., the number of all relevant Q-I pairs. Note that many real-world datasets are very sparse (Yuan et al., 2014; Singh, 2020), hence the cost is acceptable in most cases.

Here we present the convergence guarantee of SONG in the following theorem.

Theorem 1 *Under appropriate conditions and settings of* γ_0, $\eta = \mathcal{O}(\epsilon^2)$, $\beta_1 = 1-\mathcal{O}(\epsilon^2)$, *Algorithm 1 ensures that after* $T = \mathcal{O}(\epsilon^{-4})$ *iterations we can find an* ϵ*-stationary solution of* $F(\mathbf{w})$, *i.e.,* $\mathbb{E}[\|\nabla F(\mathbf{w}_\tau)\|^2] \leq \epsilon^2$ *for a random* $\tau \in \{1, \ldots, T\}$.

Remark Inspired by Qi et al. (2021), we also employ the moving average estimator technique to control the optimization error of FCCO. However, we manage to establish an iteration complexity of $\mathcal{O}(\epsilon^{-4})$, which is the same as the standard SGD for solving standard non-convex losses (Ghadimi & Lan, 2013) and better than that proved by Qi et al. (2021), i.e., $\mathcal{O}(\epsilon^{-5})$. We attribute this improvement to the use of a simple yet effective momentum-style stochastic gradient estimator $\mathbf{m}_t$ (see Algorithm 1, line 9) and a more refined proof process. A detailed analysis is provided in the 'Innovations in Proof Techniques' section in Appendix E.

5 Optimizing a smooth top-*K* NDCG surrogate

In this section, we propose an efficient stochastic algorithm to optimize the top-K variant of NDCG. By using the smooth surrogate loss $\ell(\cdot)$ for approximating the rank function, we have the following objective for top-K NDCG:

$$\frac{1}{N}\sum_{q=1}^{N} \frac{1}{Z_q^K} \sum_{\mathbf{x}_i^q \in \mathcal{S}_q^+} \mathbb{I}(\mathbf{x}_i^q \in \mathcal{S}_q[K]) \frac{2^{y_i^q} - 1}{\log_2(\bar{g}(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q) + 1)},$$

where $\mathcal{S}_q[K]$ denotes the set of top-K items in $\mathcal{S}_q$. Compared with optimizing the NDCG surrogate in (3), there is another level of complexity, i.e., the selection of top-K items from $\mathcal{S}_q$, which is non-differentiable. In the literature, Qin et al. (2010) and Wu et al. (2009) use the relationship $\mathbb{I}(\mathbf{x}_i^q \in \mathcal{S}_q[K]) = \mathbb{I}(K - r(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q) \geq 0)$ and approximate it by $\psi(K - \bar{g}(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q))$, where ψ is a continuous surrogate of the indicator function. However, there are two levels of approximation error, one lies at approximating $r(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ by $\bar{g}(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$

and the other one lies at approximating $\mathbb{I}(\cdot \geq 0)$ by $\psi(\cdot)$. To reduce the error for selecting $\mathbf{x}_i^q \in \mathcal{S}_q[K]$, we propose a more effective method, which relies on the following lemma:

Lemma 2 *Let* $\lambda_q(\mathbf{w}) = \arg\min_\lambda (K+\varepsilon)\lambda + \sum_{\mathbf{x}' \in \mathcal{S}_q} (h_q(\mathbf{x}'; \mathbf{w}) - \lambda)_+$, *where* $\varepsilon \in (0, 1)$, *then* $\lambda_q(\mathbf{w})$ *is the* $(K+1)$*-th largest value among* $h_q(\mathbf{x}', \mathbf{w})$, $\forall \mathbf{x}' \in \mathcal{S}_q$, *and hence* $\mathbf{x}_i^q \in \mathcal{S}_q[K]$ *is equivalent to* $h_q(\mathbf{x}_i^q; \mathbf{w}) > \lambda_q(\mathbf{w})$.

Remark The optimal $\lambda_q(\mathbf{w})$ can be the threshold to select top-K items in $\mathcal{S}_q$.

As a result, the problem can be converted into

$$\min_{\mathbf{w}} \frac{1}{|\mathcal{S}|} \sum_{q=1}^{N} \sum_{\mathbf{x}_i^q \in \mathcal{S}_q^+} \frac{\mathbb{I}(h_q(\mathbf{x}_i^q; \mathbf{w}) - \lambda_q(\mathbf{w}) > 0)(1 - 2^{y_i^q})}{Z_q^K \log_2(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q) + 1)}$$

$$s.t., \lambda_q(\mathbf{w}) = \arg\min_{\lambda} \frac{K+\varepsilon}{N_q}\lambda + \frac{1}{N_q} \sum_{\mathbf{x}' \in \mathcal{S}_q} (h_q(\mathbf{x}'; \mathbf{w}) - \lambda)_+.$$

There are still some challenges that prevent us developing a provable algorithm. In particular, the selection operator $\mathbb{I}(h_q(\mathbf{x}_i^q; \mathbf{w}) - \lambda_q(\mathbf{w}) > 0)$ is non-smooth w.r.t. $\mathbf{w}$ due to (i) the indicator function $\mathbb{I}(\cdot)$ is non-continuous and non-differentiable; and (ii) $\lambda_q(\mathbf{w})$ is non-smooth w.r.t. $\mathbf{w}$ because the lower optimization problem is non-smooth and non-strongly convex. To address these challenges, we first approximate $\mathbb{I}(\cdot > 0)$ by a smooth and Lipschtiz continuous function $\psi(\cdot)$, whose choice can be justified by the following lemma:

Lemma 3 *If* $\psi(h_q(\mathbf{x}_i^q; \mathbf{w}) - \lambda_q(\mathbf{w})) \leq C\mathbb{I}(h_q(\mathbf{x}_i^q; \mathbf{w}) - \lambda_q(\mathbf{w}) > 0)$ *holds for some constant* $C > 0$ *and* $\ell(\mathbf{w}; \mathbf{x}', \mathbf{x}, q) \geq \mathbb{I}(h_q(\mathbf{x}'; \mathbf{w}) - h_q(\mathbf{x}; \mathbf{w}) > 0)$, *then the function*

$\frac{1}{N}\sum_{q=1}^{N} \sum_{\mathbf{x}_i^q \in S_q^+} \frac{\psi(h_q(\mathbf{x}_i^q; \mathbf{w}) - \lambda_q(\mathbf{w}))(2^{y_i^q} - 1)}{C Z_q^K \log_2(\bar{g}(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q) + 1)}$ *is a lower bound of the top-K NDCG.*

Remark When $h_q(\mathbf{x}; \mathbf{w})$ is bounded, it is not hard to find a smooth and Lipschitz continuous function $\psi(\cdot)$ satisfying the above condition, e.g., the sigmoid function.

Next, to smooth $\lambda(\mathbf{w})$, we aim to make the lower level problem smooth and strongly convex, while not affecting the optimal solution $\lambda(\mathbf{w})$ too much. To this end, we replace the lower level problem by

$$\hat{\lambda}_q(\mathbf{w}) = \arg\min_{\lambda} L_q(\lambda; \mathbf{w}) := \frac{K+\varepsilon}{N_q}\lambda + \frac{\tau_2}{2}\lambda^2 + \frac{1}{N_q} \sum_{\mathbf{x}_i \in \mathcal{S}_q} \tau_1 \ln\left(1 + \exp\left(\frac{h_q(\mathbf{x}_i; \mathbf{w}) - \lambda}{\tau_1}\right)\right).$$

The following lemma justifies the above smoothing.

Lemma 4 *Assuming* $h_q(\mathbf{x}, \mathbf{w}) \in (0, c_h]$, *if* $\tau_1 = \tau_2 = \varepsilon$ *for some* $\varepsilon \ll 1$, *then we have* $|\hat{\lambda}_q(\mathbf{w}) - \lambda_q(\mathbf{w})| \leq \mathcal{O}(\varepsilon)$ *for any* $\mathbf{w}$. *In addition,* $L_q(\lambda; \mathbf{w})$ *is a smooth and strongly convex function in terms of* λ *for any* $\mathbf{w}$.

As a result, we propose the following problem for optimizing top-K NDCG:

$$\min F_K(\mathbf{w}) := \frac{1}{|\mathcal{S}|} \sum_{(q, \mathbf{x}_i^q) \in \mathcal{S}} \psi(h_q(\mathbf{x}_i^q; \mathbf{w}) - \hat{\lambda}_q(\mathbf{w})) f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q))$$

$$s.t., \hat{\lambda}_q(\mathbf{w}) = \arg\min_{\lambda} L_q(\lambda; \mathbf{w}), \forall q \in \mathcal{Q}, \tag{6}$$

where we employ $f_{q,i}(g)$ to denote $\frac{1}{Z_q^K}\frac{1-2^{y_i^q}}{\log_2(N_q g+1)}$.

Although (6) is a bilevel optimization problem, existing stochastic algorithms for bilevel optimization are not applicable because there are several differences from the standard bilevel optimization problem studied in the literature. First, an unbiased stochastic gradient of the objective function is not readily computed as we explained before. Second, there are multiple lower level problems in (6), whose solutions cannot be updated at the same time for all $q \in \mathcal{Q}$ when N is large. To address these challenges, we develop a tailored stochastic algorithm for solving (6).

The proposed algorithm is presented in Algorithm 2, to which we refer as K-SONG. To motivate K-SONG, we first consider the gradient of the objective function in (6), i.e., $\nabla F_K(\mathbf{w})$, as follows:

$$\frac{1}{|\mathcal{S}|}\sum_{(q,\mathbf{x}_i^q)\in\mathcal{S}}\Big(\psi'(h_q(\mathbf{x}_i^q;\mathbf{w})-\hat{\lambda}_q(\mathbf{w}))\cdot(\nabla h_q(\mathbf{x}_i^q;\mathbf{w})-\nabla_{\mathbf{w}}\hat{\lambda}_q(\mathbf{w}))\Big)f_{q,i}(g(\mathbf{w};\mathbf{x}_i^q,\mathcal{S}_q))$$
$$+\psi(h_q(\mathbf{x}_i^q;\mathbf{w})-\hat{\lambda}_q(\mathbf{w}))\nabla g(\mathbf{w};\mathbf{x}_i^q,\mathcal{S}_q)\nabla f_{q,i}(g(\mathbf{w};\mathbf{x}_i^q,\mathcal{S}_q)). \tag{7}$$

Similar to SONG, we can estimate $g(\mathbf{w}_t;\mathbf{x}_i^q,\mathcal{S}_q)$ by $\mathbf{u}_{q,i}^t$. An inherent challenge of bilevel optimization is to estimate the implicit gradient $\nabla_{\mathbf{w}}\hat{\lambda}(\mathbf{w})$. According to the optimality condition of $\hat{\lambda}(\mathbf{w})$ (Ghadimi & Wang, 2018), we can derive

$$\nabla_{\mathbf{w}}\hat{\lambda}_q(\mathbf{w}) = -\nabla^2_{\lambda,\mathbf{w}}L_q(\hat{\lambda}_q(\mathbf{w});\mathbf{w})(\nabla^2_{\lambda\lambda}L_q(\hat{\lambda}_q(\mathbf{w});\mathbf{w}))^{-1}. \tag{8}$$

To estimate $\nabla^2_{\lambda,\mathbf{w}}L_q(\hat{\lambda}(\mathbf{w});\mathbf{w})$ at t-th iteration, we use the current estimate λ_q^t in place of $\hat{\lambda}_q(\mathbf{w}_t)$ and use $L_q(\hat{\lambda},\mathbf{w};\mathcal{B}_q)$ defined by a mini-batch samples $\mathcal{B}_q$ in place of $L_q(\hat{\lambda};\mathbf{w})$, i.e.,

$$L_q(\lambda,\mathbf{w};\mathcal{B}_q) = \frac{K+\epsilon}{N_q}\lambda + \frac{\tau_2}{2}\lambda^2 + \frac{1}{|\mathcal{B}_q|}\sum_{\mathbf{x}_i\in\mathcal{B}_q}\tau_1\ln(1+\exp((h_q(\mathbf{x}_i;\mathbf{w})-\lambda)/\tau_1)).$$

Estimating $(\nabla^2_{\lambda\lambda}L_q(\hat{\lambda}_q(\mathbf{w});\mathbf{w}))^{-1}$ is more tricky. In the literature (Ghadimi & Wang, 2018), a common method is to use von Neuman series with stochastic samples to estimate it. However, such method requires multiple samples in the order of $\mathcal{O}(1/\tau_2)$, which is a large number when τ_2 is small. To address this issue, we follow a similar strategy of Guo et al. (2021a) to estimate $\nabla^2_{\lambda\lambda}L_q(\hat{\lambda}_q(\mathbf{w});\mathbf{w})$ directly by using mini-batch samples. In the proposed algorithm, we use a moving average estimator denoted by $\mathbf{s}_q$ as shown in Step 10. Finally, we have the following stochastic gradient estimator:

$$G(\mathbf{w}_t) = \frac{1}{|\mathcal{B}|}\sum_{(q,\mathbf{x}_i^q)\in\mathcal{B}}p_{q,i}\nabla\hat{g}_{q,i}(\mathbf{w}_t)$$
$$+\psi'(h_q(\mathbf{x}_i^q;\mathbf{w}_t)-\lambda_{q,t})\Big[\nabla_{\mathbf{w}}h_q(\mathbf{x}_i^q;\mathbf{w}_t)+\nabla^2_{\lambda,\mathbf{w}}L_q(\lambda_q^t,\mathbf{w}_t;\mathcal{B}_t)(\mathbf{s}_q^t)^{-1}\Big]f(\mathbf{u}_{q,i}^t), \tag{9}$$

where $p_{q,i} = \psi(h_q(\mathbf{x}_i^q;\mathbf{w}_t)-\lambda_q^t)\nabla f_{q,i}(\mathbf{u}_{q,i}^t)$ and is computed in Step 7 in K-SONG. In Appendix C, we detail the specific implementation for the two second-order derivatives with respect to L_q in (9).

It is notable that different from Guo et al. (2021a), we update λ_q^{t+1} with a mini-batch of queries q for parallel speed-up in Step 11, which makes the analysis more challenging. At last, we present the convergence guarantee of K-SONG.

Algorithm 2 Stochastic Optimization of top-K NDCG: K-SONG

Require: $\eta_0, \eta_1, \gamma_0, \gamma_0', \beta_1, \mathbf{u}^1 = 0, \mathbf{s}^1 = 0, \lambda^1 = 0, \mathbf{m}_1 = 0$
Ensure: $\mathbf{w}_{T+1}$
1: **for** $t = 1, ...T$ **do**
2: Draw some relevant Q-I pairs $\mathcal{B} = \{(q, \mathbf{x}_i^q)\} \subset \mathcal{S}$
3: For each $q \in \mathcal{B}$ draw a batch of items $\mathcal{B}_q \subset \mathcal{S}_q$
4: **for** each sampled Q-I pair $(q, \mathbf{x}_i^q) \in \mathcal{B}$ **do**
5: Let $\hat{g}_{q,i}(\mathbf{w}_t) = \frac{1}{|\mathcal{B}_q|} \sum_{\mathbf{x}' \in \mathcal{B}_q} \ell(\mathbf{w}_t; \mathbf{x}', \mathbf{x}_i^q, q)$
6: Let $\mathbf{u}_{q,i}^{t+1} = (1 - \gamma_0)\mathbf{u}_{q,i}^t + \gamma_0 \hat{g}_{q,i}(\mathbf{w}_t)$
7: Let $p_{q,i} = \psi(h_q(\mathbf{x}_i^q; \mathbf{w}_t) - \lambda_q^t) \nabla f_{q,i}(\mathbf{u}_{q,i}^t)$
8: **end for**
9: **for** each sampled query $q \in \mathcal{B}$ **do**
10: Let $\mathbf{s}_q^{t+1} = (1 - \gamma_0')\mathbf{s}_q^t + \gamma_0' \nabla_{\lambda\lambda}^2 L_q(\lambda_q^t, \mathbf{w}_t; \mathcal{B}_q)$
11: Let $\lambda_q^{t+1} = \lambda_q^t - \eta_0 \nabla_\lambda L_q(\lambda_q^t, \mathbf{w}_t; \mathcal{B}_q)$
12: **end for**
13: Compute a stochastic gradient $G(\mathbf{w}_t)$ according to (9) or (10)
14: Compute $\mathbf{m}_{t+1} = \beta_1 \mathbf{m}_t + (1 - \beta_1) G(\mathbf{w}_t)$
15: Update $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_1 \mathbf{m}_{t+1}$
16: **end for**

Theorem 2 *Under appropriate conditions and proper settings of parameters* $\gamma_0, \gamma_0', \eta_0, \eta_1 = \mathcal{O}(\epsilon^2)$, $\beta_1 = 1 - \mathcal{O}(\epsilon^2)$, *after* $T = \mathcal{O}(\epsilon^{-4})$ *iterations K-SONG can find an* ϵ*-stationary solution, i.e.,* $\mathbb{E}[\|\nabla F_K(\mathbf{w}_\tau)\|^2] \leq \epsilon^2$ *for a random* $\tau \in \{1, \ldots, T\}$.

Remark The above theorem indicates that K-SONG also has the iteration complexity of $\mathcal{O}(\epsilon^{-4})$. To achieve this, inspired by Guo et al. (2021a), we maintain variance-reduced estimators for the key functions in our algorithm. However, unlike Guo et al. (2021a), our algorithm implements parallel speed-up in lines 9–12 of Algorithm 2 and introduces new proof techniques to control the optimization error of the lower-level problems. In Appendix E, we provide detailed assumptions, parameter settings, and proofs for Theorems 1 and 2. We also provide a proof sketch to help readers better understand our algorithm. Additionally, we highlight the differences between our proofs and those of similar algorithms in the 'Innovations in Proof Techniques' section, emphasizing our contributions.

6 Practical strategies

In this section, we present two practical strategies for improving the effectiveness of SONG/K-SONG in deep learning applications.

6.1 Initial warm-up

A potential problem of optimizing NDCG is that it may not lead to a good local minimum if a bad initial solution is given. To address this issue, we use warm-up to find a good initial solution by solving a well-behaved objective. Similar strategies have been used in the literature (Yuan et al., 2020; Qi et al., 2021), however, their objectives are not suitable for ranking. Here we choose the listwise cross-entropy loss (Cao et al., 2007), i.e.,

$$\min_{\mathbf{w}} \frac{1}{N}\sum_{q=1}^{N}\frac{1}{N_q}\sum_{\mathbf{x}_i^q\in\mathcal{S}_q^+} -\ln\left(\frac{\exp(h_q(\mathbf{x}_i^q;\mathbf{w})}{\sum_{\mathbf{x}_j^q\in\mathcal{S}_q} h_q(\mathbf{x}_j^q;\mathbf{w}))}\right),$$

which is the cross-entropy between predicted and ground truth top-one probability distributions. We use PyTorch's default methods to initialize $\mathbf{w}$ in practice. The objective can be formulated as a similar finite-sum coupled compositional problem as NDCG, and a similar algorithm to SONG can be used to solve it. We present the formulation and detailed algorithm in Appendix A.

6.2 Stop gradient for the top-K Selector

Given a good initial solution, we justify that the second term in (9) is close to 0 under a reasonable condition, and present the details in Appendix B. Thus, the gradient of the top-K selector $\psi(h(\mathbf{x}_i^q,\mathbf{w}) - \hat{\lambda}_q(\mathbf{w}))$ is not essential. Hence we can apply the stop gradient operator on the top-K selector, and compute the gradient estimator by

$$G(\mathbf{w}_t) = \frac{1}{|\mathcal{B}|}\sum_{(q,\mathbf{x}_i^q)\in\mathcal{B}} p_{q,i}\nabla\hat{g}_{q,i}(\mathbf{w}_t), \tag{10}$$

which **simplifies** K-SONG by avoiding maintaining and updating $s_{q,t}$. We refer to the K-SONG using the gradient in (9) as theoretical K-SONG, and the K-SONG using the gradient in (10) as practical K-SONG.

7 Optimizing NDCG and top-K NDCG with faster convergence

The algorithms SONG/K-SONG are effective for deep learning but do not achieve the optimal $\mathcal{O}(\epsilon^{-3})$ iteration complexity for smooth non-convex optimization (Arjevani et al., 2022). In this section, we present two types of algorithms that achieve the optimal iteration complexity by utilizing advanced variance reduced estimators in different ways to further control gradient approximation errors. We begin by describing the two key estimators employed, STORM (Cutkosky & Orabona, 2019) and MSVR (Jiang et al., 2022). Then, we elaborate two strategies for using these estimators, and conclude with the algorithms and their theoretical guarantees.

STORM is a famous variance reduction estimator that achieves an iteration complexity of $\mathcal{O}(\epsilon^{-3})$ on smooth losses. Assume that we have a target function $\nabla f(\mathbf{w}_t;\mathcal{S})$, its STORM estimator $\mathbf{d}^t$ is updated by

$$\begin{aligned}\mathbf{d}^{t+1} &= (1-\gamma_t)\mathbf{d}^t + \gamma_t\nabla f(\mathbf{w}_t;\mathcal{B}) + \underbrace{(1-\gamma_t)(\nabla f(\mathbf{w}_t;\mathcal{B}) - \nabla f(\mathbf{w}_{t-1};\mathcal{B}))}_{\text{error correction}}\\ &= (1-\gamma_t)(\mathbf{d}^t - \nabla f(\mathbf{w}_{t-1};\mathcal{B})) + \nabla f(\mathbf{w}_t;\mathcal{B}),\end{aligned}$$

where $\mathcal{B}$ is sampled from $\mathcal{S}$. Compared with moving average estimators, it is notable that the STORM estimator employs an additional error correction term to alleviate the noise from sampling $\mathcal{B}$. To improve the iteration complexity, we first design a STORM-style stochastic gradient estimator $\mathbf{m}_t$ for updating the model parameters:

$$\mathbf{m}_t = (1-\gamma_{m,t})(\mathbf{m}_{t-1} - G(\mathbf{w}_{t-1})) + G(\mathbf{w}_t), \tag{11}$$

where $G(\mathbf{w}_t)$ and $G(\mathbf{w}_{t-1})$ are computed using (5) and (9) to optimize NDCG and its top-K variant, respectively, with $\gamma_{m,t}$ serving as a tunable parameter.

However, for the moving average estimators $\mathbf{u}_{q,i}$ and $\mathbf{s}_q$ for tracking $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ and $\nabla^2_{\lambda\lambda} L_q(\hat{\lambda}_q(\mathbf{w}); \mathbf{w})$ in SONG/K-SONG, we cannot simply replace them to the corresponding STORM estimators. The reason is that the updates of $\mathbf{u}$ and $\mathbf{s}$ involve *twofold* randomness: sampling a set of queries $\mathcal{B}$ and sampling items $\mathcal{B}_q$ for each query $q \in \mathcal{B}$. The error correction term in STORM addresses only the randomness associated with sampling queries, and does not account for the noise introduced by sampling items.

To address this issue, Jiang et al. (2022) propose a new variance reduced estimator named MSVR, which is inspired by STORM but introduces a customized error correction term to alleviate the noise from both sampling $\mathcal{B}$ and $\mathcal{B}_q$. Specifically, the MSVR estimator $\mathbf{d}_q^t$ for tracking $\nabla f_q(\mathbf{w}_t; \mathcal{S})$ is updated by

$$\mathbf{d}_q^{t+1} = \begin{cases} (1-\gamma_t)\mathbf{d}_q^t + \gamma_t \nabla f_q(\mathbf{w}_t; \mathcal{B}_q) + \underbrace{\beta_t(\nabla f_q(\mathbf{w}_t; \mathcal{B}_q) - \nabla f_q(\mathbf{w}_{t-1}; \mathcal{B}_q))}_{\text{error correction}} & \text{if } q \in \mathcal{B} \\ \mathbf{d}_q^t & \text{o.w.} \end{cases},$$

where β_t can be set to $\frac{N-|\mathcal{B}|}{|\mathcal{B}|(1-\gamma_t)} + (1-\gamma_t)$ according to the analysis (Jiang et al., 2022), N is the total number of queries. It is notable that if we set β_t to $1-\gamma_t$, then MSVR estimator will reduce to STORM estimator. Jiang et al. (2022) prove that with MSVR estimator, the aforementioned FCCO problems with non-convex objectives can be solved with an improved iteration complexity of $\mathcal{O}(\epsilon^{-3})$.

Next, we need to determine how to employ the MSVR estimator to track $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ and $\nabla^2_{\lambda\lambda} L_q(\hat{\lambda}_q(\mathbf{w}); \mathbf{w})$. An intuitive approach is to directly apply these estimators to track these crucial functions. To this end, we maintain the following MSVR estimator for $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$:

$$\mathbf{u}_{q,i}^{t+1} = \begin{cases} (1-\gamma_{u,t})\mathbf{u}_{q,i}^t + \gamma_{u,t} g(\mathbf{w}_t; \mathbf{x}_i^q, \mathcal{B}_q) \\ \qquad + \beta_{u,t}(g(\mathbf{w}_t; \mathbf{x}_i^q, \mathcal{B}_q) - g(\mathbf{w}_{t-1}; \mathbf{x}_i^q, \mathcal{B}_q)) & \text{if } (q, \mathbf{x}_i^q) \in \mathcal{B}, \\ \mathbf{u}_{q,i}^t & \text{o.w.} \end{cases} \tag{12}$$

where $\gamma_{u,t}$ and $\beta_{u,t}$ are adjustable parameters. Similarly, The MSVR estimator for $\nabla^2_{\lambda\lambda} L_q(\hat{\lambda}_q(\mathbf{w}); \mathbf{w})$ can be updated by

$$\mathbf{s}_q^{t+1} = \begin{cases} (1-\gamma_{s,t})\mathbf{s}_q^t + \gamma_{s,t} \nabla^2_{\lambda\lambda} L_q(\lambda_q^t; \mathbf{w}_t; \mathcal{B}_q) \\ \qquad + \beta_{s,t}(\nabla^2_{\lambda\lambda} L_q(\lambda_q^t; \mathbf{w}_t; \mathcal{B}_q) - \nabla^2_{\lambda\lambda} L_q(\lambda_q^{t-1}; \mathbf{w}_{t-1}; \mathcal{B}_q)) & \text{if } q \in \mathcal{B}. \\ \mathbf{s}_q^t & \text{o.w.} \end{cases} \tag{13}$$

where $\gamma_{s,t}$ and $\beta_{s,t}$ are tunable parameters. We define such straightforward application of MSVR estimators for tracking the target functions in (12) and (13) as the **v1 type update**.

We can also view the functions $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ and $\nabla^2_{\lambda\lambda} L_q(\hat{\lambda}_q(\mathbf{w}); \mathbf{w})$ from another perspective. Inspired by recent work in bilevel optimization (Li et al., 2022; Dagréou et al., 2022), we can consider these two functions as **the solutions to specific quadratic problems**. Therefore, we can approximate these functions by iteratively solving these quadratic problems. We employ quadratic functions due to their smoothness and strongly convexity, enabling us to derive algorithms with fast convergence. Specifically, for the function $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$, we define the following minimization problem:

$$\min_{\mathbf{u}} \tilde{g}_{q,i}(\mathbf{u}, \mathbf{w}) := \frac{1}{2}\|\mathbf{u} - g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)\|^2. \tag{14}$$

It is not difficult to show that $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ is the solution for minimizing $\tilde{g}_{q,i}(\mathbf{u}, \mathbf{w})$. Thus, we can set $\mathbf{u}_{q,i}$ as an estimator for $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$, and update $\mathbf{u}_{q,i}$ using the stochastic gradient estimator computed for solving $\min_{\mathbf{u}} \tilde{g}_{q,i}(\mathbf{u}, \mathbf{w})$, which is given by

$$\nabla_u \tilde{g}_{q,i}(\mathbf{u}, \mathbf{w}; \mathcal{B}_q) = \mathbf{u}_{q,i} - g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{B}_q).$$

However, directly updating with the above stochastic gradient estimator results in poorer iteration complexity. To avoid this, we update all sampled blocks of $\mathbf{u}_{q,i}$ using its MSVR gradient estimator $\mathbf{v}_{q,i}$ as follows:

$$\mathbf{v}_{q,i}^t = \begin{cases} (1-\gamma_{v,t})\mathbf{v}_{q,i}^{t-1} + \gamma_{v,t}\nabla_u \tilde{g}_{q,i}(\mathbf{u}^t, \mathbf{w}_t; \mathcal{B}_q) \\ \qquad +\beta_{v,t}(\nabla_u \tilde{g}_{q,i}(\mathbf{u}^t, \mathbf{w}_t; \mathcal{B}_q) - \nabla_u \tilde{g}_{q,i}(\mathbf{u}^{t-1}, \mathbf{w}_{t-1}; \mathcal{B}_q)) & \text{if } q \in \mathcal{B}, \\ \mathbf{v}_{q,i}^{t-1} & \text{o.w.} \end{cases} \tag{15}$$

and then an update $\mathbf{u}_{q,i}^{t+1} = \mathbf{u}_{q,i}^t - \tau\tau_t \mathbf{v}_{q,i}^t$ for the sampled items can be conducted. Here, $\tau\tau_t$ represents the step size, where the parameter τ_t is primarily set for theoretical analysis (as detailed in the proof of Theorem 5 in Appendix F). In practice, $\tau\tau_t$ can be treated as a tunable learning rate parameter.

A similar approach can also be applied to estimate $\nabla^2_{\lambda\lambda} L_q(\hat{\lambda}_q(\mathbf{w}); \mathbf{w})$. When designing the quadratic problem for this function, which is a Hessian matrix, direct estimation might lead to significant approximation errors. Thus, we consider a minimization problem through *matrix-vector products*. To illustrate our method clearly, we restate the gradient for the top-K NDCG surrogate by substituting (8) into (7):

$$\begin{aligned} &\frac{1}{|\mathcal{S}|}\sum_{(q,\mathbf{x}_i^q)\in\mathcal{S}} \Big\{\psi'(h_q(\mathbf{x}_i^q;\mathbf{w}) - \hat{\lambda}_q(\mathbf{w}))\nabla h_q(\mathbf{x}_i^q;\mathbf{w}) f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)) \\ &+ \psi'(h_q(\mathbf{x}_i^q;\mathbf{w}) - \hat{\lambda}_q(\mathbf{w}))\nabla^2_{\lambda,\mathbf{w}} L_q(\hat{\lambda}_q(\mathbf{w}); \mathbf{w})(\nabla^2_{\lambda\lambda} L_q(\hat{\lambda}_q(\mathbf{w}); \mathbf{w}))^{-1} f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)) \\ &+ \psi(h_q(\mathbf{x}_i^q; \mathbf{w}) - \hat{\lambda}_q(\mathbf{w}))\nabla g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)\nabla f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)\Big\}. \end{aligned} \tag{16}$$

Thus, we estimate the Hessian-vector product $(\nabla^2_{\lambda\lambda} L_q(\hat{\lambda}_q(\mathbf{w}); \mathbf{w}))^{-1} f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q))$. To this end, we define the following problem for query q:

$$\min_{\mathbf{s}} \ \phi_q(\mathbf{s}, \hat{\lambda}_q(\mathbf{w}), \mathbf{w}) := \frac{1}{2}\mathbf{s}^\top \nabla^2_{\lambda\lambda} L_q(\hat{\lambda}_q(\mathbf{w}); \mathbf{w})\mathbf{s} - \mathbf{s}^\top f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)). \tag{17}$$

Note that the optimal $\mathbf{s}$ for (17) is equal to $(\nabla^2_{\lambda\lambda} L_q(\hat{\lambda}_q(\mathbf{w}); \mathbf{w}))^{-1} f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q))$. Similarly, we can approximate $(\nabla^2_{\lambda\lambda} L_q(\hat{\lambda}_q(\mathbf{w}); \mathbf{w}))^{-1} f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q))$ by solving $\min_{\mathbf{s}} \phi_q(\mathbf{s}, \hat{\lambda}_q(\mathbf{w}), \mathbf{w})$. To this end, we first define the stochastic estimator for $\phi_q(\mathbf{s}, \hat{\lambda}_q(\mathbf{w}), \mathbf{w})$ as follows:

$$\nabla_{\mathbf{s}} \phi_q(\mathbf{s}, \mathbf{w}_t; \mathcal{B}_q) = \nabla^2_{\lambda\lambda} L_q(\lambda_q^t; \mathbf{w}_t; \mathcal{B}_q)\mathbf{s}_q - f_{q,i}(\mathbf{u}_{q,i}^t),$$

where we employ $\mathbf{u}_{q,i}^t$ as the estimate for $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$. Then the MSVR estimator for the gradient is given by

$$\mathbf{r}_q^t = \begin{cases} (1-\gamma_{r,t})\mathbf{r}_q^{t-1} + \gamma_{r,t}\nabla_{\mathbf{s}}\phi_q(\mathbf{s}^t, \mathbf{w}_t; \mathcal{B}_q) \\ \qquad +\beta_{r,t}(\nabla_{\mathbf{s}}\phi_q(\mathbf{s}^t, \mathbf{w}_t; \mathcal{B}_q) - \nabla_{\mathbf{s}}\phi_q(\mathbf{s}^{t-1}, \mathbf{w}_{t-1}; \mathcal{B}_q)) & \text{if } q \in \mathcal{B}. \\ \mathbf{r}_q^{t-1} & \text{o.w.} \end{cases} \tag{18}$$

and then an update $\mathbf{s}_q^{t+1} = \mathbf{s}_q^t - \tau\tau_t \mathbf{r}_q^t$ for the sampled queries can be conducted. We refer to these update rules as the **v2 type update**. Intuitively, the v2 type update involves solving

Algorithm 3 Faster SONG$^{\text{v1/v2}}$

Require: $\mathbf{w}_0, \mathbf{w}_1, \mathbf{m}_0 = 0, \mathbf{v}^0 = 0, \mathbf{u}^0 = \mathbf{u}^1 = 0$, update type: v1 or v2
Ensure: $\mathbf{w}_{T+1}$
for $t = 1, 2, \ldots, T$ **do**
  Draw some relevant Q-I pairs $\mathcal{B} = \{(q, \mathbf{x}_i^q)\} \subset \mathcal{S}$
  For each $q \in \mathcal{B}$ draw a batch of items $\mathcal{B}_q \subset \mathcal{S}_q$
  if using v1 type update **then**
    Compute $\mathbf{u}_{q,i}^{t+1}$ according to (12) ▷ Use MSVR update
  else // using v2 type update
    Compute $\mathbf{v}_{q,i}^{t}$ according to (15) ▷ Use MSVR update
    Update $\mathbf{u}_{q,i}^{t+1} = \mathbf{u}_{q,i}^{t} - \tau\tau_t \mathbf{v}_{q,i}^{t}$
  end if
  Compute the gradient estimator $G(\mathbf{w}_{t-1})$ and $G(\mathbf{w}_t)$ by (5)
  Compute $\mathbf{m}_t$ according to (11) ▷ Use STORM update
  $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha\eta_t \mathbf{m}_t$
end for

lower-level problems to estimate crucial functions, essentially remaining a bilevel optimization with multiple lower-level problems. Therefore, the complexity of the problem remains unchanged, and both types of updates exhibit the same convergence rate, which will be demonstrated later.

To clearly demonstrate the two estimators based on the MSVR update rule proposed in this section, we follow the experimental setup from Fig. 1 in Sect. 4, and present the approximation errors of the two new estimators in Fig. 2. It is notable that both estimators outperform the moving average estimator. Additionally, we find that the v2 type update yields better results than the v1 type update.

Last, notice that the stochastic gradient estimator for λ, i.e., $\nabla_\lambda L_q(\lambda_q^t; \mathbf{w}_t; \mathcal{B}_q)$, also involves the randomness from both sampling $\mathcal{B}$ and $\mathcal{B}_q$, thus we also have to use the technique of MSVR for it. Let $\mathbf{z}_q$ denote the gradient estimator for $\nabla_\lambda L_q(\lambda_q^t; \mathbf{w}_t; \mathcal{B}_q)$, and it is updated as follows:

$$\mathbf{z}_q^t = \begin{cases} (1-\gamma_{z,t})\mathbf{z}_q^{t-1} + \gamma_{z,t}\nabla_\lambda L_q(\lambda_q^t; \mathbf{w}_t; \mathcal{B}_q) \\ \qquad +\beta_{z,t}(\nabla_\lambda L_q(\lambda_q^t; \mathbf{w}_t; \mathcal{B}_q) - \nabla_\lambda L_q(\lambda_q^{t-1}; \mathbf{w}_{t-1}; \mathcal{B}_q)), & \text{if } q \in \mathcal{B}, \\ \mathbf{z}_q^{t-1}, & \text{o.w.} \end{cases} \tag{19}$$

where $\gamma_{z,t}$ and $\beta_{z,t}$ are tunable parameters. Notably, the use of the variance-reduced gradient estimator $\mathbf{z}_q$ is essential for proving the algorithm's optimal convergence rate. For more details, please refer to the proof in Appendix F.

With these considerations in hand, we propose improved stochastic algorithms named **Faster SONG/K-SONG** for optimizing NDCG and its top-K variant with the iteration complexity of $\mathcal{O}(\epsilon^{-3})$ in Algorithms 3 and 4, respectively. For each algorithm, we provide two variants that utilize variance reduced estimators in v1 or v2 type update. The effect of parameter η_t in the algorithms is similar to that of parameter τ_t, and $\alpha\eta_t$ can be viewed as one parameter in practice. We derive the following guarantee for Faster SONG$^{\text{v1/v2}}$/K-SONG$^{\text{v1/v2}}$:

Theorem 3 *Under appropriate conditions, with* $\eta_t = \tau_t = \Theta(t^{-1/3})$ *and* $\gamma_z = \gamma_u = \gamma_s = \gamma_m = \Theta(\eta_t^2)$, *Algorithm 4 ensures that after* $T = \mathcal{O}(\frac{1}{\epsilon^3})$ *iterations, we can find an* ϵ*-stationary solution of* $F(\mathbf{w}_t)$, *i.e.,* $\mathbb{E}\left[\sum_{t=1}^T \frac{1}{T}\|\nabla F(\mathbf{w}_t)\|^2\right] \leq \mathcal{O}\left(\frac{1}{T^{2/3}}\right)$.

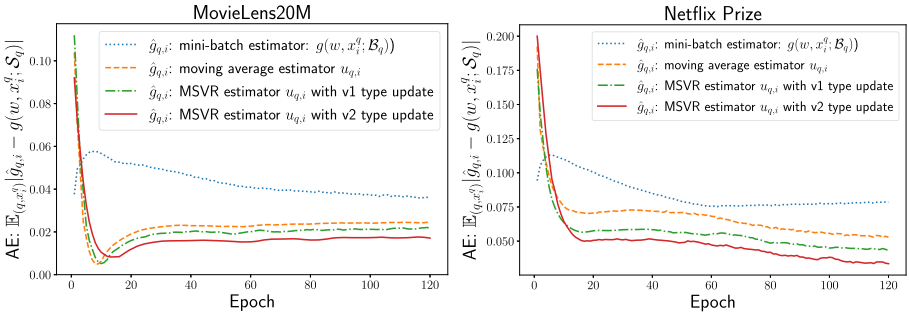

Fig. 2 Comparing the approximation error (AE) of the mini-batch estimator $g(\mathbf{w}, \mathbf{x}_i^q, \mathcal{B}_q)$, moving average estimator $\mathbf{u}_{q,i}$, and two types of estimators with MSVR-style updates for the function $g(\mathbf{w}, \mathbf{x}_i^q, \mathcal{S}_q)$

Algorithm 4 Faster K-SONG$^{\text{v1/v2}}$

Require: $\mathbf{w}_0$, $\mathbf{w}_1$, initialize $\mathbf{m}_0$, λ^0, λ^1, $\mathbf{z}^0$, $\mathbf{u}^0$, $\mathbf{s}^0$, $\mathbf{u}^1$, $\mathbf{s}^1$, $\mathbf{v}^0$, $\mathbf{r}^0$ to 0, update type: v1 or v2
Ensure: $\mathbf{w}_{T+1}$
for $t = 1, 2, \ldots, T$ **do**
  Draw some relevant Q-I pairs $\mathcal{B} = \{(q, \mathbf{x}_i^q)\} \subset \mathcal{S}$
  For each $q \in \mathcal{B}$ draw a batch of items $\mathcal{B}_q \subset \mathcal{S}_q$
  if using v1 type update **then**
    Compute $\mathbf{u}_{q,i}^{t+1}$ according to (12) ▷ Use MSVR update
    Compute $\mathbf{s}_q^{t+1}$ according to (13) ▷ Use MSVR update
  else // using v2 type update
    Compute $\mathbf{v}_{q,i}^t$ and $\mathbf{r}_q^t$ according to (15) and (18) ▷ Use MSVR update
    Update $\mathbf{u}_{q,i}^{t+1} = \mathbf{u}_{q,i}^t - \tau\tau_t \mathbf{v}_{q,i}^t$, $\mathbf{s}_q^{t+1} = \mathbf{s}_q^t - \tau\tau_t \mathbf{r}_q^t$
  end if
  Compute $\mathbf{z}_q^t$ according to (19) ▷ Use MSVR update
  for each sampled query $q \in \mathcal{B}$ **do**
    Update $\lambda_q^{t+1} = \lambda_q^t - \tau\tau_t \mathbf{z}_q^t$
  end for
  Compute stochastic gradient estimator $G(\mathbf{w}_{t-1})$ and $G(\mathbf{w}_t)$ according to (9) or (10)
  Compute gradient estimator $\mathbf{m}_t$ according to (11) ▷ Use STORM update
  $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha\eta_t \mathbf{m}_t$
end for

Remark 1 The achieved iteration complexity (i) matches the optimal iteration complexity of $\mathcal{O}(\epsilon^{-3})$ for standard smooth non-convex stochastic optimization (Arjevani et al., 2022), and (ii) enjoys a parallel speedup by sampling multiple queries and items at each iteration. It's worth noting that, although v1 and v2 employ different methods to estimate functions $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ and $\nabla^2_{\lambda\lambda} L_q(\hat{\lambda}(\mathbf{w}); \mathbf{w})$, both variants still have the same iteration complexity.

Remark 2 Our algorithms are related to previous work on variance reduction (Cutkosky & Orabona, 2019; Jiang et al., 2022) and SBO (Guo et al., 2021a), but have significant differences. Firstly, unlike Cutkosky and Orabona (2019) and Jiang et al. (2022), we study a complex SBO problem that includes multiple lower-level problems. Additionally, in contrast to Guo et al. (2021a), our algorithms are designed to solve multiple lower-level problems in parallel per iteration. To this end, our algorithms use STORM and MSVR in novel ways to better estimate key functions, while introducing new mechanisms for solving the lower-level problems and developing proof techniques for a tighter error bound. We refer the interested

readers to Appendix F for a proof sketch, a comprehensive comparison of our work with similar works, and a detailed proof of Theorem 3.

8 Broadening framework applications to other metrics

In this section, we highlight the versatility of our algorithmic frameworks, which can optimize a broad spectrum of metrics, including Precision@K/Recall@K, Average Precision (AP), mean Average Precision (mAP), and their top-K variants. We begin by defining these metrics, followed by the presentation of provably efficient stochastic algorithms for optimizing Precision@K (equivalent to Recall@K up to a constant) and top-K mAP (with AP, top-K AP, and mAP being its special cases).

Precision@K and Recall@K are key performance metrics commonly used in *binary* classification tasks. They measure Precision and Recall on the top K samples with the highest scores, with their formulas as follows:

$$\text{Precision@}K = \frac{\sum_{\mathbf{x}_i \in \mathcal{S}_+} \mathbb{I}(\mathbf{x}_i \in \mathcal{S}[K])}{K},$$
$$\text{Recall@}K = \frac{\sum_{\mathbf{x}_i \in \mathcal{S}_+} \mathbb{I}(\mathbf{x}_i \in \mathcal{S}[K])}{|\mathcal{S}_+|},$$

where $\mathcal{S}_+$ indicates all relevant (positive) items, and $\mathcal{S}[K]$ denotes the set of top-K items of $\mathcal{S}$. It is notable that **Precision@K is equivalent to Recall@K up to a constant** for a given dataset.

Leveraging Lemma 2 and 4, we can formulate the problem of maximizing Precision@K as the following bilevel optimization problem:

$$\min_{\mathbf{w}} F_{\text{Prec@}K}(\mathbf{w}) := \frac{1}{K} \sum_{\mathbf{x}_i \in \mathcal{S}_+} \ell(h_{\mathbf{w}}(\mathbf{x}_i) - \lambda(\mathbf{w}))$$
$$\lambda(\mathbf{w}) = \arg\min_{\lambda} L(\lambda, \mathbf{w}) := \frac{K+\epsilon}{|\mathcal{S}|}\lambda + \frac{\tau_2}{2}\lambda^2 + \frac{1}{|\mathcal{S}|}\sum_{\mathbf{x}_i \in \mathcal{S}} \tau_1 \ln\left(1 + \exp\left(\frac{h(\mathbf{x}_i; \mathbf{w}) - \lambda}{\tau_1}\right)\right), \quad (20)$$

where $\ell(\cdot)$ is a differentiable non-decreasing surrogate function for replacing the indicator function, e.g., a squared hinge loss. Similar to our previous derivation for top-K NDCG, we can derive the expression for $\nabla F_{\text{Prec@}K}(\mathbf{w})$ as follows:

$$\frac{1}{K} \sum_{\mathbf{x}_i \in \mathcal{S}_+} \ell'(h_{\mathbf{w}}(\mathbf{x}_i) - \lambda(\mathbf{w})) \left(\nabla_{\mathbf{w}} h_{\mathbf{w}}(\mathbf{x}_i) + \nabla^2_{\lambda,\mathbf{w}} L(\hat{\lambda}(\mathbf{w}), \mathbf{w})[\nabla^2_{\lambda\lambda} L(\hat{\lambda}(\mathbf{w}), \mathbf{w})]^{-1}\right),$$

where we employ the fact that $\nabla_{\mathbf{w}} \lambda(\mathbf{w}) = -\nabla^2_{\lambda,\mathbf{w}} L(\hat{\lambda}(\mathbf{w}), \mathbf{w})[\nabla^2_{\lambda\lambda} L(\hat{\lambda}(\mathbf{w}), \mathbf{w})]^{-1}$. Assuming that at the t-th iteration, we sample a mini-batch $\mathcal{B}$ with positive samples $\mathcal{B}_+$, the stochastic gradient estimator can be calculated as follows:

$$G(\mathbf{w}_t) = \frac{|\mathcal{S}_+|}{K} \mathbb{E}_{\mathbf{x}_i \in \mathcal{B}_+} \ell'(h_{\mathbf{w}}(\mathbf{x}_i) - \lambda^t) \left(\nabla_{\mathbf{w}} h_{\mathbf{w}}(\mathbf{x}_i) + \nabla^2_{\lambda,\mathbf{w}} L(\lambda^t, \mathbf{w}_t; \mathcal{B})[\mathbf{s}^t]^{-1}\right), \quad (21)$$

where $\mathbf{s}$ is the moving average estimator for tracking $\nabla^2_{\lambda\lambda} L(\hat{\lambda}(\mathbf{w}), \mathbf{w})$, and λ^t is the current estimate for $\lambda(\mathbf{w}_t)$. In our implementation, we employ the **controlled data sampler** provided by LibAUC[2] (Yuan et al., 2023), which not only controls the number of positive and negative

[2] https://libauc.org/

Algorithm 5 Stochastic Optimization for Precision@K/Recall@K

Require: $\eta_0, \eta_1, \gamma_0, \beta_1, \mathbf{s}^1 = 0, \lambda^1 = 0, \mathbf{m}_1 = 0$
Ensure: $\mathbf{w}_{T+1}$
1: **for** $t = 1, ...T$ **do**
2: Draw a mini-batch $\mathcal{B}$, where the positive samples are denoted by $\mathcal{B}_+$
3: Update $\mathbf{s}^{t+1} = (1-\gamma_0)\mathbf{s}^t + \gamma_0 \nabla^2_{\lambda\lambda} L(\lambda^t, \mathbf{w}_t; \mathcal{B})$
4: Update $\lambda^{t+1} = \lambda^t - \eta_0 \nabla_\lambda L(\lambda^t, \mathbf{w}_t; \mathcal{B})$
5: Compute a stochastic gradient $G(\mathbf{w}_t)$ according to (21)
6: Compute $\mathbf{m}_{t+1} = \beta_1 \mathbf{m}_t + (1-\beta_1) G(\mathbf{w}_t)$
7: Update $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_1 \mathbf{m}_{t+1}$
8: **end for**

samples but also ensures that positive samples precede negative ones in each mini-batch, facilitating the computation of the expectation over $\mathcal{B}_+$.

We present the complete Precision@K optimization algorithm in Algorithm 5. Since for the same task, Precision@K and Recall@K differ only by a constant, this algorithm can also be used to optimize Recall@K. Note that the objective function for Precision@K in (20) can be viewed as an extreme case of optimizing the top-K NDCG surrogate, involving only one single lower-level problem. Therefore, we can analyze Algorithm 5 in a similar manner and establish the same iteration complexity as K-SONG, i.e., $\mathcal{O}(\epsilon^{-4})$.

Average Precision (AP) calculates the average precision each time a new positive (relevant) item is retrieved for binary classification tasks, which is computed as

$$\text{AP} = \sum_{\mathbf{x}_i \in \mathcal{S}_+} \frac{r(\mathbf{w}; \mathbf{x}_i, \mathcal{S}_+)}{r(\mathbf{w}; \mathbf{x}_i, \mathcal{S})},$$

where $r(\mathbf{w}; \mathbf{x}, \mathcal{S})$ denotes the rank of $\mathbf{x}$ w.r.t. the set $\mathcal{S}$. Mean Average Precision (mAP) is defined as the average of the AP scores calculated for all tasks (classes), with its top-K variant calculated based on the items ranked within the top-K positions by their prediction scores. The definitions of mAP and top-K mAP are:

$$\text{mAP} = \frac{1}{N} \sum_{q=1}^{N} \sum_{\mathbf{x}_i^q \in \mathcal{S}_q^+} \frac{r(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q^+)}{r(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)},$$

$$\text{Top-}K \text{ mAP} = \frac{1}{N} \sum_{q=1}^{N} \sum_{\mathbf{x}_i^q \in \mathcal{S}_q^+} \mathbb{I}(\mathbf{x}_i^q \in \mathcal{S}_q[K]) \frac{r(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q^+)}{r(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)},$$

where we consider N tasks, where S_q and S_q^+ represent the total number of samples and the total number of positive samples in the q-th task, respectively. It is worth to mention that mAP provides a holistic view of model performance across multiple tasks and helps in assessing models in various deep learning applications, including object detection (Ren et al., 2015), information retrieval (Kishida, 2005), and natural language processing (Voorhees, 1999).

Noting that AP and top-K AP can be considered special forms of mAP and top-K mAP, we proceed to present an optimization algorithm for mAP and top-K mAP. Similar to optimizing the NDCG surrogate, we first replace $r(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ with a surrogate function, and introduce the following objective for mAP:

$$\min_{\mathbf{w}} -\frac{1}{N} \sum_{q=1}^{N} \sum_{\mathbf{x}_i^q \in \mathcal{S}_q^+} \frac{\sum_{\mathbf{x}' \in \mathcal{S}_q} \mathbb{I}(\mathbf{x}' \in \mathcal{S}_q^+) \ell(h_q(\mathbf{x}'; \mathbf{w}) - h_q(\mathbf{x}_i^q; \mathbf{w}))}{\sum_{\mathbf{x}' \in \mathcal{S}_q} \ell(h_q(\mathbf{x}'; \mathbf{w}) - h_q(\mathbf{x}_i^q; \mathbf{w}))},$$

Algorithm 6 Stochastic Optimization for mAP/top-K mAP

Require: $\eta, \gamma_0, \beta_1, \mathbf{u}^1 = 0, \mathbf{u}^2 = 0, \mathbf{m}_1 = 0$
Ensure: $\mathbf{w}_{T+1}$
for $t = 1, ...T$ **do**
  Draw some positive (relevant) Q-I pairs $\mathcal{B} = \{(q, \mathbf{x}_i^q)\} \subset \mathcal{S}$
  For each sampled q draw a batch of items $\mathcal{B}_q \subset \mathcal{S}_q$
  for each sampled Q-I pair $(q, \mathbf{x}_i^q) \in \mathcal{B}$ **do**
    Compute $g(\mathbf{w}_t; \mathbf{x}_i^q, \mathcal{B}_q)$ according to (22)
    Update $\mathbf{u}_{q,i}^1 = (1-\gamma_0)\mathbf{u}_{q,i}^1 + \gamma_0[g(\mathbf{w}_t; \mathbf{x}_i^q, \mathcal{B}_q)]_1$
    Update $\mathbf{u}_{q,i}^2 = (1-\gamma_0)\mathbf{u}_{q,i}^2 + \gamma_0[g(\mathbf{w}_t; \mathbf{x}_i^q, \mathcal{B}_q)]_2$
  end for
  Compute the stochastic gradient estimator $G(\mathbf{w}_t)$ according to (23) or (24)
  Compute $\mathbf{m}_{t+1} = \beta_1 \mathbf{m}_t + (1-\beta_1)G(\mathbf{w}_t)$
  update $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{m}_{t+1}$
end for

where $\ell(\cdot)$ is a smooth surrogate function. We further employ

$$
\begin{aligned}
g(\mathbf{w}; \mathbf{x}', \mathbf{x}_i^q) &= \left[\mathbb{I}(\mathbf{x}' \in \mathcal{S}_q^+)\ell(h_q(\mathbf{x}'; \mathbf{w}) - h_q(\mathbf{x}_i^q; \mathbf{w})), \ell(h_q(\mathbf{x}'; \mathbf{w}) - h_q(\mathbf{x}_i^q; \mathbf{w}))\right], \\
g(\mathbf{w}; \mathbf{x}_i^q; \mathcal{S}_q) &= \mathbb{E}_{\mathbf{x}' \in \mathcal{S}_q} g(\mathbf{w}; \mathbf{x}', \mathbf{x}_i^q) : \mathbb{R}^d \to \mathbb{R}^2, \\
f_{q,i}(\mathbf{s}) &= -\frac{s_1}{s_2} : \mathbb{R}^2 \to \mathbb{R},
\end{aligned} \tag{22}
$$

and the objective can be converted into

$$
\min_{\mathbf{w} \in \mathbb{R}^d} F_{\text{mAP}}(\mathbf{w}) := \frac{1}{|\mathcal{S}|} \sum_{(q, \mathbf{x}_i^q) \in \mathcal{S}} f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)),
$$

where $\mathcal{S} = \{(q, \mathbf{x}_i^q), q \in [N], \mathbf{x}_i^q \in \mathcal{S}_q^+\}$. Note that the above problem is also an instance of FCCO problem like (3) for optimizing NDCG. Thus, we can employ the previously introduced algorithm frameworks for the NDCG surrogate to optimize mAP. To this end, we first derive the gradient of $\mathbf{w}$ w.r.t. F_{mAP}:

$$
\begin{aligned}
\nabla_{\mathbf{w}} F_{\text{mAP}}(\mathbf{w}) &= \frac{1}{|\mathcal{S}|} \sum_{(q, \mathbf{x}_i^q) \in \mathcal{S}} \nabla_{\mathbf{w}} g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q) \nabla f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q))^\top \\
&= \frac{1}{|\mathcal{S}|} \sum_{(q, \mathbf{x}_i^q) \in \mathcal{S}} \nabla_{\mathbf{w}} g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q) \left(\frac{-1}{[g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)]_2}, \frac{[g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)]_1}{([g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)]_2)^2} \right)^\top.
\end{aligned}
$$

The major cost for computing $\nabla_{\mathbf{w}} F_{\text{mAP}}(\mathbf{w})$ lies at evaluating function g and its gradient. The stochastic estimator for $\nabla_{\mathbf{w}} g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ can be simply computed by:

$$
\nabla_{\mathbf{w}} g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{B}_q) = \begin{pmatrix} \frac{1}{|\mathcal{B}_q|} \sum_{x' \in \mathcal{B}_q} \mathbb{I}(\mathbf{x}' \in \mathcal{B}_q^+) \nabla \ell(h_q(\mathbf{x}'; \mathbf{w}) - h_q(\mathbf{x}_i^q; \mathbf{w})) \\ \frac{1}{|\mathcal{B}_q|} \sum_{x' \in \mathcal{B}_q} \nabla \ell(h_q(\mathbf{x}'; \mathbf{w}) - h_q(\mathbf{x}_i^q; \mathbf{w})) \end{pmatrix}^\top,
$$

where $\mathcal{B}_q$ denotes a mini-batch of samples from $\mathcal{S}_q$. To control the approximation error from g, we employ two moving average estimators $\mathbf{u}_{q,i}^1$ and $\mathbf{u}_{q,i}^2$ for tracking $[g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)]_1$ and

$[g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)]_2$, respectively. Thus, the stochastic gradient estimator is:

$$G_{\text{mAP}}(\mathbf{w}_t) = \frac{1}{|\mathcal{B}|} \sum_{(q,\mathbf{x}_i^q)\in\mathcal{B}} \nabla_{\mathbf{w}} g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{B}_q) \left(\frac{-1}{\mathbf{u}_{q,i}^2}, \frac{\mathbf{u}_{q,i}^1}{(\mathbf{u}_{q,i}^2)^2} \right)^\top \tag{23}$$

where $\mathcal{B}$ is a batch sampled from $\mathcal{S}$. Similar to top-K NDCG, the objective of optimizing top-K mAP is as follows:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{|\mathcal{S}|} \sum_{(q,\mathbf{x}_i^q)\in\mathcal{S}} \mathbb{I}(\mathbf{x}_i^q \in \mathcal{S}_q[K]) f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)),$$

where $\mathcal{S}_q[K]$ denotes the set of top-K items in $\mathcal{S}_q$. Then, for approximating the top-K selector $\mathbb{I}(\mathbf{x}_i^q \in \mathcal{S}_q[K])$, we employ the relationship $\mathbb{I}(\mathbf{x}_i^q \in \mathcal{S}_q[K]) = \mathbb{I}(K - \sum_{\mathbf{x}_j^q\in\mathcal{S}_q} \mathbb{I}(h_q(\mathbf{x}_j^q; \mathbf{w}) \geq h_q(\mathbf{x}_i^q; \mathbf{w})))$, and approximate it by $\sigma(K - \sum_{\mathbf{x}_j^q\in\mathcal{S}_q} \ell(h_q(\mathbf{x}_j^q; \mathbf{w}) - h_q(\mathbf{x}_i^q; \mathbf{w})))$, where $\sigma(\cdot)$ is a surrogate of the indicator function, e.g., the sigmoid function. Therefore, we have the following smooth surrogate for top-K mAP:

$$F_{\text{mAP@}K}(\mathbf{w}) := \frac{1}{|\mathcal{S}|} \sum_{(q,\mathbf{x}_i^q)\in\mathcal{S}} \sigma\left(K - \sum_{\mathbf{x}_j^q\in\mathcal{S}_q} \ell(h_q(\mathbf{x}_j^q; \mathbf{w}) - h_q(\mathbf{x}_i^q; \mathbf{w})) \right) f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)).$$

To derive the stochastic gradient estimator for the above objective, we still employ $\mathbf{u}_{q,i}^1$ and $\mathbf{u}_{q,i}^2$ and use obtain:

$$G_{\text{mAP@}K}(\mathbf{w}_t) = \frac{1}{|\mathcal{B}|} \sum_{(q,\mathbf{x}_i^q)\in\mathcal{B}} \left\{ \sigma\left(K - \frac{|\mathcal{S}_q|}{|\mathcal{B}_q|} \sum_{\mathbf{x}_j^q\in\mathcal{B}_q} \ell(h_q(\mathbf{x}_j^q; \mathbf{w}) - h_q(\mathbf{x}_i^q; \mathbf{w})) \right) \right.$$
$$\left. \cdot \nabla_{\mathbf{w}} g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{B}_q) \left(\frac{-1}{\mathbf{u}_{q,i}^2}, \frac{\mathbf{u}_{q,i}^1}{(\mathbf{u}_{q,i}^2)^2} \right)^\top \right\}, \tag{24}$$

where we employ the previously introduced stop-gradient techniques to avoid the computational overhead associated with calculating the gradient of the top-K selector. We demonstrate the stochastic optimization algorithms for optimizing mAP and top-K mAP in Algorithm 6. Since the objective functions for optimizing mAP and top-K mAP can be converted into FCCO problems akin to that for the NDCG surrogate, we can similarly analyze Algorithm 6 and establish an iteration complexity identical to that of SONG, i.e., $\mathcal{O}(\epsilon^{-4})$.

9 Experiments

In this section, we evaluate our algorithms for optimizing NDCG and its top-K variant across two domains: learning to rank and recommender systems. Experimental results demonstrate that our algorithms surpass existing ranking methods in terms of NDCG. We also conduct experiments to validate our algorithmic designs, including the moving average estimator, initial warm-up, and bilevel formulation for top-K NDCG. A hyperparameter analysis is performed to determine the impact of various parameters within our algorithms. Additionally, to confirm the efficacy of our proposed Faster SONG/K-SONG, we compare them with the variants that incorporate STORM estimators directly into our frameworks. Lastly, we evaluate

Table 1 The test NDCG on two learning to rank datasets

Method	MSLR WEB30K		Yahoo! LTR Dataset	
	NDCG@1	NDCG@5	NDCG@1	NDCG@5
RankNet	0.5138±0.0008	0.5159±0.0003	0.7066±0.0006	0.7368±0.0005
ListNet	0.5105±0.0001	0.5146±0.0002	0.7066±0.0002	0.7352±0.0004
ListMLE	0.5153±0.0012	0.5136±0.0005	0.7067±0.0008	0.7353±0.0007
LambdaRank	0.5173±0.0014	0.5187±0.0003	0.7084±0.0003	0.7352±0.0004
LambdaLoss	0.5182±0.0009	0.5183±0.0007	0.7086±0.0005	0.7354±0.0005
ApproxNDCG	0.5204±0.0007	0.5179±0.0006	0.7085±0.0009	0.7350±0.0006
NeuralNDCG	0.5160±0.0006	0.5155±0.0002	0.7076±0.0003	0.7349±0.0003
SmoothI	0.5236±0.0004	0.5193±0.0005	0.7115±0.0004	0.7364±0.0004
SONG	0.5265±0.0005	0.5206±0.0003	0.7131±0.0002	0.7390±0.0002
K-SONG	0.5271±0.0006	0.5204±0.0003	0.7128±0.0004	0.7394±0.0008
Faster SONG^{v1}	0.5274±0.0007	0.5219±0.0007	0.7130±0.0003	0.7397±0.0005
Faster K-SONG^{v1}	0.5273±0.0005	0.5223±0.0004	**0.7134**±0.0003	0.7392±0.0003
Faster SONG^{v2}	0.5280±0.0006	**0.5231**±0.0003	0.7128±0.0004	0.7406±0.0004
Faster K-SONG^{v2}	**0.5282**±0.0004	0.5230±0.0005	0.7131±0.0006	**0.7408**±0.0003

We report the average NDCG@K ($K \in [1, 5]$) and standard deviation with 3 different random seeds
Bold represent the best performance metrics

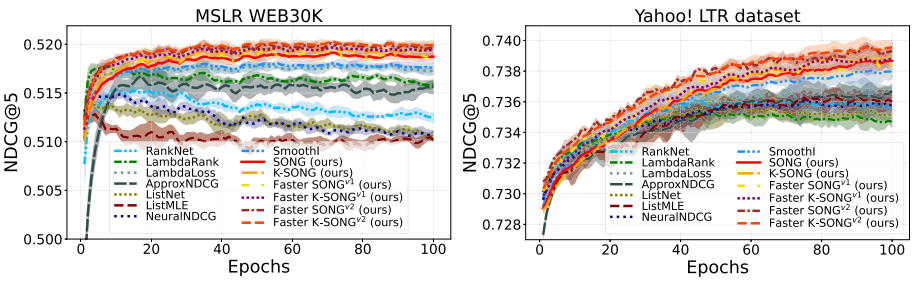

Fig. 3 Convergence of different methods in terms of validation NDCG@5 scores

our algorithms for optimizing Precision@K and top-K mAP on two graph classification tasks to demonstrate the flexibility of our frameworks.

We compare our algorithms against the following NDCG optimization methods: **RankNet** (Burges et al., 2005b), **ListNet** (Cao et al., 2007), **ListMLE** (Xia et al., 2008), **LambdaRank** (Burges et al., 2005a), **ApproxNDCG** (Qin et al., 2010), **LambdaLoss** (Wang et al., 2018), **NeuralNDCG** (Pobrotyn & Bialobrzeski, 2021), and **SmoothI** (Thonet et al., 2022). We do not compare with SoftRank (Taylor et al., 2008), as its $\mathcal{O}(n^3)$ complexity is prohibitive. Similar to NeuralNDCG, PiRank (Swezey et al., 2021) also employs NeuralSort (Grover et al., 2019) to approximate NDCG, so we do not compare with it. For K-SONG, we report its theoretical version results unless specified otherwise. The hyper-parameters of all losses are fine-tuned using grid search with training/validation splits mentioned below. We present the detailed implementation information in Appendix D.

9.1 Learning to rank

9.1.1 Data

Learning to rank (LTR) algorithms aim to rank a set of candidate items for a given search query. We consider two datasets: MSLR-WEB30K (Qin & Liu, 2013) and Yahoo! LTR dataset (Chapelle & Chang, 2011), which are the largest public LTR datasets from commercial search engines. Both datasets contain query-document pairs represented by real-valued feature vectors, and have associated relevance scores on the scale from 0 to 4. Following Ai et al. (2019), we use the training/validation/test sets in the Fold1 of MSLR-WEB30K dataset for evaluation. The Yahoo! LTR dataset splits the queries arbitrarily and uses 19,944 for training, 2,994 for validation and 6,983 for testing. We present the detailed information of these two datasets in Appendix D.

9.1.2 Setup

We adopt the Context-Aware Ranker (Pobrotyn et al., 2020) as the backbone network, which takes raw features of items in a list as input and outputs a real-valued score for each item. We first pre-train models by initial warm-up, and then re-initialize the last layer and train the model by different methods as mentioned before. In both stages, we set the initial learning rate and batch size to 0.001 and 64, respectively. We train the networks for 100 epochs, decaying the learning rate by 0.1 after 50 epochs. Our algorithms employ multiple estimators involving several hyperparameters. For the moving average parameters γ in the moving average estimators, we adjust them within the range $\{0.1, 0.3, 0.5, 0.7\}$. For the additional error correction parameters β in the MSVR estimators, their tuning range is $\{0.001, 0.005, 0.01\}$. Additionally, when optimizing top-K NDCG, we adjust K within the range of $\{50, 100, 300\}$.

9.1.3 Results

The results presented in Table 1 indicate that methods directly optimizing NDCG surrogates exhibit superior performance, aligning with findings from other studies (Qin et al., 2010; Pobrotyn & Bialobrzeski, 2021). Our SONG and K-SONG consistently outperform all prior baseline methods across both datasets, demonstrating their efficacy for LTR tasks. Notably, Faster SONG/K-SONG shows better performance than SONG/K-SONG, suggesting an improved iteration complexity. Moreover, the v2 type update is found to be more effective than the v1 type update. We also present the training curves of our algorithms alongside those of other baseline methods in Fig. 3, highlighting the faster convergence of our algorithms.

9.2 Recommender systems

9.2.1 Data

We use two movie recommendation datasets: MovieLens20M (Harper & Konstan, 2015) and Netflix Prize dataset (Bennett et al., 2007). Both datasets contain large numbers of users and movies represented by integer IDs. All users have rated several movies, with ratings range from 1 to 5. We use the most recent rated item of each user for testing, the second recent item

Table 2 The test NDCG on movie recommendation data

Method	MovieLens20M		Netflix Prize Dataset	
	NDCG@10	NDCG@20	NDCG@10	NDCG@20
RankNet	0.0538±0.0011	0.0744±0.0013	0.0362±0.0002	0.0489±0.0003
ListNet	0.0660±0.0003	0.0875±0.0004	0.0532±0.0002	0.0700±0.0002
ListMLE	0.0588±0.0001	0.0799±0.0001	0.0376±0.0003	0.0508±0.0004
LambdaRank	0.0697±0.0001	0.0913±0.0002	0.0531±0.0002	0.0693±0.0002
LambdaLoss	0.0712±0.0004	0.0929±0.0004	0.0557±0.0004	0.0703±0.0006
ApproxNDCG	0.0735±0.0005	0.0938±0.0003	0.0434±0.0005	0.0592±0.0009
NeuralNDCG	0.0692±0.0003	0.0901±0.0003	0.0554±0.0002	0.0718±0.0003
SmoothI	0.0739±0.0006	0.0952±0.0004	0.0566±0.0003	0.0725±0.0004
SONG	0.0748±0.0002	0.0969±0.0002	0.0571±0.0002	0.0749±0.0002
K-SONG	0.0747±0.0002	0.0973±0.0003	0.0573±0.0003	0.0743±0.0003
Faster SONG^{v1}	0.0761±0.0003	0.0974±0.0004	0.0583±0.0003	0.0762±0.0003
Faster K-SONG^{v1}	0.0760±0.0004	0.0986±0.0003	0.0579±0.0003	0.0759±0.0004
Faster SONG^{v2}	**0.0765**±0.0005	0.0989±0.0005	0.0588±0.0004	**0.0776**±0.0005
Faster K-SONG^{v2}	0.0757±0.0007	**0.0995**±0.0003	**0.0597**±0.0002	0.0765±0.0003

We report the average NDCG@K ($K \in [10, 20]$) and standard deviation with 3 different random seeds
Bold represent the best performance metrics

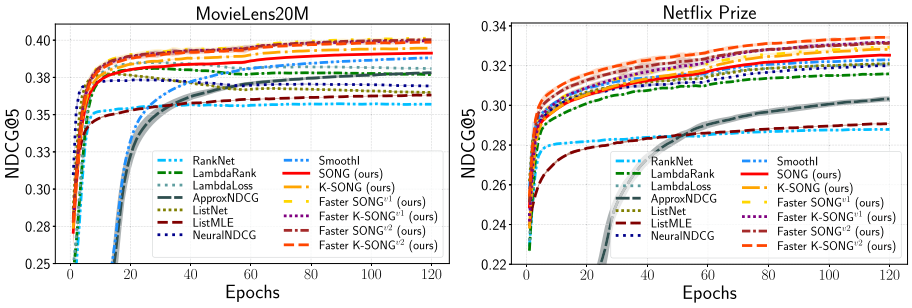

Fig. 4 Convergence of different methods in terms of validation NDCG@5 scores

for validation, and the remaining items for training, which is widely-used in the literature (He et al., 2018; Wang et al., 2020). During training, we rank the relevant (rated) items with 1000 unrated items to compute validation NDCG scores. When testing, however, we adopt the all ranking protocol Wang et al. (2019); He et al. (2020) — all unrated items are used for evaluation.

9.2.2 Setup

We choose NeuMF (He et al., 2017) as the backbone network. All models are first pre-trained by our initial warm-up method for 20 epochs with the learning rate 0.001 and a batch size of 256. Then the last layer is randomly re-initialized and the network is fine-tuned by different methods. At the fine-tuning stage, the initial learning rate and weight decay are set to 0.0004 and 1e-7, respectively. We train the models for 120 epochs with the learning rate multiplied

Table 3 We present the mean values and standard deviations of Precision@K and Top-K mAP, where K is set to 50, 100, and 300, across three different random seeds for the HIV and PCBA datasets

HIV	Precision@50	Precision@100	Precision@300
Cross-entropy Loss	0.43±0.05	0.28±0.03	0.16±0.03
SmoothI P@k Loss	0.46±0.03	0.31±0.07	0.18±0.04
Precision@k Loss (ours)	**0.47**±0.05	**0.33**±0.06	**0.22**±0.02
PCBA	Top-50 mAP	Top-100 mAP	Top-300 mAP
Cross-entropy Loss	0.4331±0.0009	0.3649±0.0011	0.3577±0.0014
Focal Loss	0.4782±0.0015	0.3957±0.0010	0.3814±0.0018
Top-k mAP Loss (ours)	**0.5139**±0.0017	**0.4207**±0.0013	**0.4022**±0.0019

Bold represent the best performance metrics

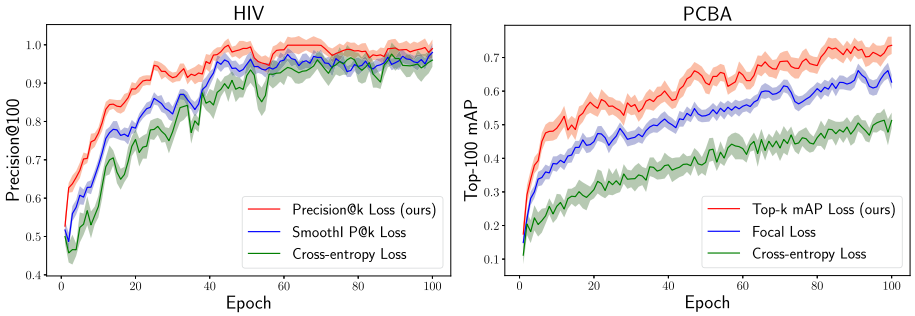

Fig. 5 Training curves on two graph classification tasks

by 0.25 at 60 epochs. The tuning ranges for the hyperparameters γ, β, and K in our algorithm are consistent with those used in previous learning to rank experiments.

9.2.3 Results

We evaluate all methods by calculating NDCG@K ($K \in [10, 20]$) on the test data, with results reported in Table 2. SONG consistently outperformed all previous baselines across both datasets, achieving improvements of 3.30% and 4.32% in NDCG@20 over the best baselines on the MovieLens20M and Netflix Prize datasets, respectively. Moreover, K-SONG generally performs better than SONG. These results clearly demonstrate the effectiveness of our algorithms in optimizing NDCG and its top-K variant. Additionally, Faster SONG/K-SONG show faster convergence rates than SONG/K-SONG, as illustrated in Fig. 4. It is worth to mention that the improvements from our methods on RS datasets are higher than that on LTR datasets. The reason is that RS datasets have about 20,000 items per query, while most queries in LTR datasets have less than 1,000 items (detailed statistics in Appendix D). These results validate that our methods are more advantageous for large-scale data.

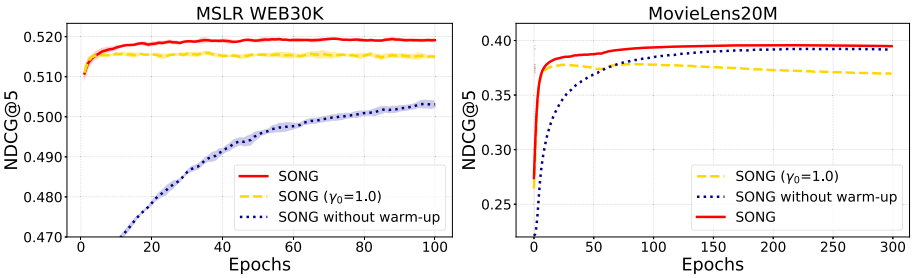

Fig. 6 Ablation study on two variants of SONG on learning to rank data MSLR Web30K and movie recommendation data MovieLens20M

9.3 Graph classification for molecular property prediction

9.3.1 Data

To further demonstrate the advantages of our algorithmic frameworks, we conduct experiments for optimizing Precision@K and top-K mAP on graph classification tasks. We employ the datasets HIV and PCBA from the MoleculeNet (Wu et al., 2018), which is a benchmark for molecular property prediction. The HIV dataset has 41,913 molecules with binary labels, and the positive samples are molecules tested to have inhibition ability to HIV. We employ this dataset to evaluate our Precision@K algorithm. The PCBA dataset contains 437,929 molecules and the task on this dataset is to predict 128 different biological activities. Therefore we use PCBA to evaluate our top-K mAP algorithms. We use the split of training/validation/test sets and node features of graphs provided by OGB (Hu et al., 2020a).

9.3.2 Setup

Many recent studies have shown that graph neural networks (GNNs) are powerful for graph data analysis (Gao et al., 2018; Rong et al., 2020). Hence, we employ the widely used graph isomorphism network (GIN) (Xu et al., 2018) as the backbone network for graph classification. For both tasks, We set the number of layers and hidden state dimensionality to 5 and 300, respectively. We train the models using different methods by Adam with 100 epochs and a learning rate of 0.001. For our algorithms for optimizing Precision@K and top-K mAP, we tune hyper-parameters γ and K in the ranges of $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ and $\{50, 100, 300, 500\}$, respectively.

9.3.3 Results

We compare our method with several baseline methods in Table 3 and present the training curves for these methods in Fig. 5. One can be observe that our algorithms perform well on both tasks. Given the smaller scale of the graph classification task on the HIV dataset, which features binary labels, we find that even simple cross-entropy loss yields satisfactory results. The SmoothI method also demonstrates good performance. On the larger-scale PCBA dataset, which encompasses more tasks, our method demonstrates significant advantages over traditional cross-entropy loss and focal loss (Lin et al., 2017).

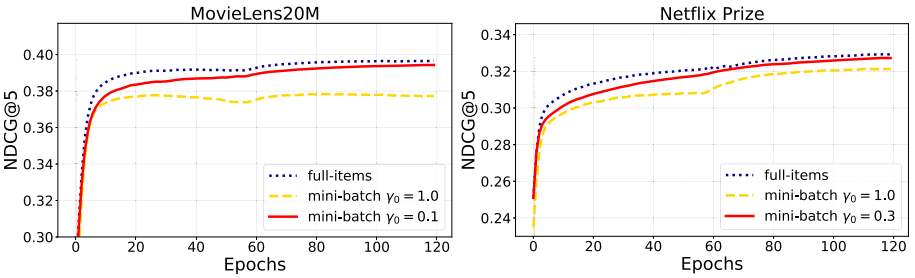

Fig. 7 Comparison of full-items and mini-batch training

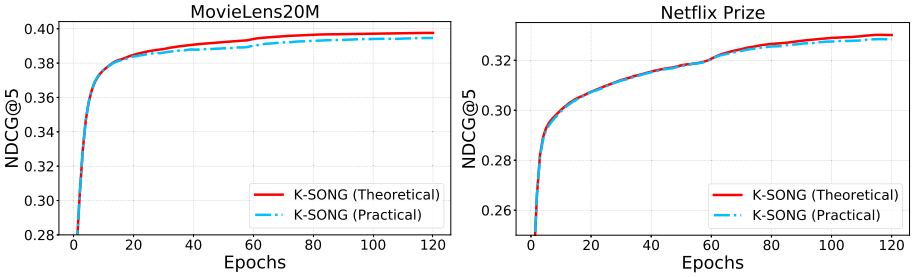

Fig. 8 Comparison of theoretical and practical K-SONG

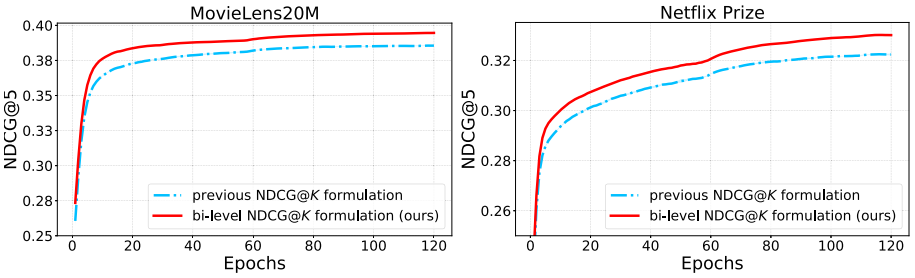

Fig. 9 Comparison of our bilevel NDCG@K formulation and previous NDCG@K formulation

9.4 In-depth analyses

9.4.1 Ablation studies

We now study the effects of the moving average estimators in our methods and initial warm-up. We present partial experimental results on MSLR Web30K and MovieLens20M data in Fig. 6 and full results in Fig. 14 in Appendix D. First, we can observe that maintaining the moving average estimators enables our algorithm perform better. Second, we consistently observe that initial warm-up can bring the model to a good initialization state and improve the final performance of the model.

Table 4 Hyperparameter analyses for γ, β, and K. We report NDCG@5 and NDCG@10 for the results on LTR and recommendation dataset, respectively

γ	0.1	0.3	0.5	0.7
MSLR Web30K	0.5201±0.0007	0.5214±0.0004	**0.5230**±0.0005	0.5222±0.0006
Yahoo! LTR	0.7395±0.0005	0.7404±0.0007	**0.7408**±0.0003	0.7391±0.0004
MovieLens20M	**0.0757**±0.0007	0.0747±0.0004	0.0735±0.0006	0.0721±0.0009
Netflix Prize	0.0592±0.0004	**0.0597**±0.0002	0.0594±0.0003	0.0583±0.0005
β	0	0.001	0.005	0.01
MSLR Web30K	0.5204±0.0003	**0.5230**±0.0005	0.5193±0.0006	0.5177±0.0002
Yahoo! LTR	0.7394±0.0008	0.7402±0.0006	**0.7408**±0.0003	0.7380±0.0003
MovieLens20M	0.0747±0.0002	**0.0757**±0.0007	0.0742±0.0004	0.0731±0.0007
Netflix Prize	0.0573±0.0003	**0.0597**±0.0002	0.0582±0.0008	0.0571±0.0009
K	50	100	300	
MSLR Web30K	**0.5230**±0.0005	0.5226±0.0004	0.5221±0.0007	
Yahoo! LTR	0.7392±0.0003	**0.7408**±0.0003	0.7406±0.0004	
MovieLens20M	0.0741±0.0008	0.0752±0.0006	**0.0757**±0.0007	
Netflix Prize	0.0588±0.0009	**0.0597**±0.0002	0.0593±0.0004	

Bold represent the best performance metrics

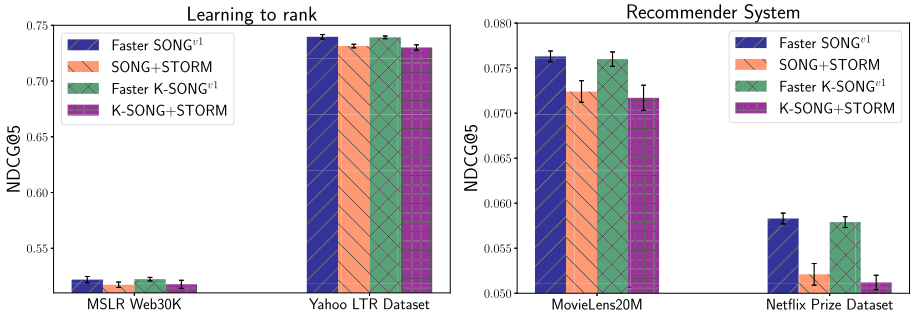

Fig. 10 Comparison with STORM-style Variants

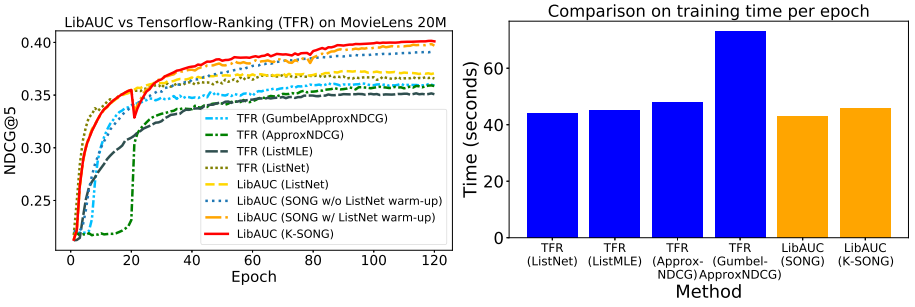

Fig. 11 Comparison of convergence (left) and training time (right) between LibAUC (ours) and TensorFlow Ranking library

9.4.2 Comparison with full-items training

We compare three different training methods: full-items gradient descent that uses *all items* in $\mathcal{S}_q$ to computing $g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)$ and its gradient, biased mini-batch gradient descent (i.e., set $\gamma_0 = 1.0$ in our algorithms), and our algorithms (i.e., with γ_0 tuned). We employ these methods for NDCG maximization on movie recommendation data, where the $|\mathcal{S}_q|$ values are much greater than that of LTR data, and present the results in Fig. 7. We can see that our methods converge to that of full-items gradient descent, which proves the effectiveness of our algorithms. We also provide the negative loglikelihood loss curves of three different training methods for warm-up in Fig. 15 in Appendix D, and similar conclusions can be reached.

9.4.3 Theoretical and practical K-SONG

To verify the effectiveness of stop gradient operator, we present the comparison of theoretical K-SONG and practical K-SONG on the left of Fig. 8. We observe that practical K-SONG and theoretical K-SONG achieve similar performance on both datasets, which indicates that the proposed stop gradient operator is effective in simplifying theoretical K-SONG.

9.4.4 The advantage of the bilevel formulation

To demonstrate the advantage of our bilevel formulation for optimizing the top-K NDCG surrogate, we implement previous NDCG@K formulation by modifying our Algorithm 1 for optimizing the NDCG@K objective with $\psi(K - \bar{g}(\mathbf{w}, \mathbf{x}))$ in place of $\mathbb{I}(K \geq r(\mathbf{w}; \mathbf{x}))$. We compare these two formulations and present the results on the middle of Fig. 9, and we can see that our bilevel formulation is more advantageous.

9.4.5 Comparison with STORM-style variants

To further demonstrate the effectiveness of our methods, we replace the MSVR estimators in our Faster SONG$^{\text{v1}}$/K-SONG$^{\text{v1}}$ algorithms with the STORM estimators, renaming the modified algorithms SONG/K-SONG+STORM. We compare the performance of these algorithms across four datasets, with the results shown in Fig. 10. The figure reveals that simply using the STORM estimator leads to poorer performance. This is due to the reasons explained in Sect. 7: the STORM estimator cannot simultaneously control the twofold errors introduced by sampling queries and the items for sampled queries when optimizing NDCG and top-K NDCG.

9.4.6 The effect of hyperparameters

We study the impact of the following hyperparameters in our algorithms: γ in the moving average estimator, β in the MSVR estimator, and the value of K in top-K NDCG optimization. We choose Faster K-SONG$^{\text{v2}}$ algorithm and present the effects of these hyperparameters across four datasets in Table 4. We observe that γ in the moving average estimator is quite important and greatly affects final performance. Additionally, a smaller value of β in MSVR tends to yield better results. Lastly, our algorithm is not sensitive to the choice of K. In Fig. 16 in Appendix D, we further demonstrate the impact of the parameter γ on the convergence rate during training.

9.4.7 Comparison with tensorflow ranking

We implement our SONG and K-SONG in the LibAUC[3] library, and compare them with four ranking methods, i.e., ListNet, ListMLE, ApproxNDCG, and Gumbel-ApproxNDCG, in TensorFlow Ranking library[4] (Pasumarthi et al., 2019) (TFR). All models are trained for 120 epochs on MovieLens20M with the learning rate 0.001 and a batch size of 256. For SONG and K-SONG, we first train the models by initial warm-up for the first 20 epochs, and then keep training the models by SONG or K-SONG for 100 epochs. We present the comparison of convergence and training time per epoch in Fig. 11. We notice that our implementations of SONG and K-SONG in the LibAUC library converge faster than the algorithms in the TFR library, which indicates the advantages of our implementations in LibAUC.

10 Conclusion

In this work, we first introduce a novel compositional optimization problem to optimize NDCG, and a novel bilevel compositional optimization problem for top-K NDCG. Then, we develop innovative algorithms named SONG/K-SONG for these problems with provable convergence. To overcome SONG/K-SONG's suboptimal convergence rate, we integrate advanced variance reduced estimators into our frameworks and introduce two types of algorithms that utilize these estimators in different ways, referred to as Faster SONG$^{v1/v2}$/K-SONG$^{v1/v2}$. We demonstrate these algorithms achieves both the optimal iteration complexity for smooth non-convex optimization and parallel speed-up. To demonstrate the flexibility of our frameworks, we further design efficient and provable algorithms for other widely-used metrics including Precision/Recall@K, mAP, and top-K mAP. Comprehensive experiments on learning to rank, recommender systems, and graph classification tasks demonstrate the effectiveness of our algorithms.

Appendix A Initial warm-up

The listwise cross-entropy loss can be reformulated as follows:

$$\begin{aligned}
&\min_{\mathbf{w}} \frac{1}{N}\sum_{q=1}^{N}\frac{1}{N_q}\sum_{\mathbf{x}_i^q\in\mathcal{S}_q^+} -\ln\frac{\exp(h_q(\mathbf{x}_i^q;\mathbf{w}))}{\sum_{\mathbf{x}_j^q\in\mathcal{S}_q} h_q(\mathbf{x}_j^q;\mathbf{w})} \\
&= \frac{1}{N}\sum_{q=1}^{N}\frac{1}{N_q}\sum_{\mathbf{x}_i^q\in\mathcal{S}_q^+} \ln\left(\sum_{\mathbf{x}_j^q\in\mathcal{S}_q}\exp(h_q(\mathbf{x}_j^q)-h_q(\mathbf{x}_i^q))\right).
\end{aligned}$$

The above objective has the same structure of the NDCG surrogate, i.e., it is an instance of finite-sum coupled compositional stochastic optimization problem. Hence, we can use a similar algorithm to SONG to solve the above problem. We present the details in Algorithm 7.

[3] https://www.libauc.org

[4] https://www.tensorflow.org/ranking

Algorithm 7 Stochastic Optimization of Listwise CE loss: SOLC

Require: $\eta, \beta_0, \beta_1, \mathbf{u}^1 = 0, \mathbf{m}_1 = 0$
Ensure: $\mathbf{w}_T$
for $t = 1, ..., T$ **do**
Draw a set of queries denoted by $\mathcal{Q}_t$
For each query draw a batches of examples $\{\mathcal{B}_q^+, \mathcal{B}_q\}$, where $\mathcal{B}_q^+$ denote a set of sampled relevant documents for q and $\mathcal{B}_q$ denote a set of sampled documents from $\mathcal{S}_q$
for $\mathbf{x}_i^q \in \mathcal{B}_q^+$ for each $q \in \mathcal{Q}_t$ **do**
$\mathbf{u}_{q,i}^{t+1} = (1-\gamma_0)\mathbf{u}_{q,i}^t + \gamma_0 \frac{1}{|\mathcal{B}_q|} \sum_{\mathbf{x}' \in \mathcal{B}_q} \exp(h_q(\mathbf{x}'; \mathbf{w}) - h_q(\mathbf{x}; \mathbf{w}))$
Compute $p_{q,i} = 1/\mathbf{u}_{q,i}^{t+1}$
end for
Compute gradient

$$G(\mathbf{w}_t) = \frac{1}{|\mathcal{Q}_t|} \frac{1}{|\mathcal{B}_q^+|} \frac{1}{|\mathcal{B}_q|} \sum_{q \in \mathcal{Q}_t} \sum_{\mathbf{x}_i^q \in \mathcal{B}_q^+} \sum_{\mathbf{x}_j^q \in \mathcal{B}_q} p_{q,i} \nabla_{\mathbf{w}} (h_q(\mathbf{x}_j^q; \mathbf{w}_t) - h_q(\mathbf{x}_i^q; \mathbf{w}_t))$$

Compute $\mathbf{m}_{t+1} = \beta_1 \mathbf{m}_t + (1-\beta_1) G(\mathbf{w}_t)$
Update $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{m}_{t+1}$
end for

Appendix B Justification of stop gradient operator

Below, we provide a justification by showing that the second term in (9) is close to 0 under a reasonable condition. For simplicity of notation, we let $\psi_i(\mathbf{w}, \hat{\lambda}_q(\mathbf{w})) = \psi(h(\mathbf{x}_i^q, \mathbf{w}) - \hat{\lambda}_q(\mathbf{w}))$. Its gradient is given by

$$\nabla_{\mathbf{w}} \psi_i = \psi_i'(\mathbf{w}, \hat{\lambda}_q(\mathbf{w})) \Big(\nabla_{\mathbf{w}} h(\mathbf{x}_i^q, \mathbf{w}) - \nabla^2_{\mathbf{w}\lambda} L_q(\mathbf{w}, \hat{\lambda}_q(\mathbf{w})) [\nabla^2_{\lambda} L_q(\mathbf{w}, \hat{\lambda}_q(\mathbf{w}))]^{-1} \Big).$$

For the purpose of justification, we can approximate $\phi(h_q(\mathbf{x}_i; \mathbf{w}) - \lambda) = \tau_1 \log(1 + \exp((h_q(\mathbf{x}_i; \mathbf{w}) - \lambda)/\tau_1))$ by a smoothed hinge loss function, $\kappa(h_q(\mathbf{x}_i; \mathbf{w}) - \lambda) = \max_\alpha \alpha(h_q(\mathbf{x}_i; \mathbf{w}) - \lambda) - \tau_1 \alpha^2/2$, which is equivalent to

$$\kappa(h_q(\mathbf{x}_i; \mathbf{w}) - \lambda) = \begin{cases} 0, & h_q(\mathbf{x}_i; \mathbf{w}) - \lambda \le 0 \\ \frac{(h_q(\mathbf{x}_i; \mathbf{w}) - \lambda)^2}{2\tau_1}, & 0 < h_q(\mathbf{x}_i; \mathbf{w}) - \lambda \le \tau_1 \\ h_q(\mathbf{x}_i; \mathbf{w}) - \lambda - \frac{\tau_1}{2}, & h_q(\mathbf{x}_i; \mathbf{w}) - \lambda > \tau_1 \end{cases}$$

Please refer to Fig. 12 for the curves of $[\cdot]_+$ and $\phi(\cdot)$ and $\kappa(\cdot)$. Below, we assume $L_q(\mathbf{w}, \lambda)$ is defined by using $\kappa(h_q(\mathbf{x}_i; \mathbf{w}) - \lambda)$ in place of $\phi(h_q(\mathbf{x}_i; \mathbf{w}) - \lambda)$.

For any $\mathbf{w}$, let us consider a subset $\mathcal{C}_q = \{\mathbf{x}_i^q \in \mathcal{S}_q^+ : h_{\mathbf{w}}(\mathbf{x}_i^q) - \hat{\lambda}_q(\mathbf{w}) \in (0, \tau_1)\}$. It is not difficult to show that

$$\begin{aligned} \nabla^2_{\mathbf{w}\lambda} L_q(\mathbf{w}, \hat{\lambda}_q(\mathbf{w})) &= \frac{1}{N_q} \sum_{\mathbf{x}_i^q \in \mathcal{C}_q} \frac{-\partial_{\mathbf{w}} h(\mathbf{x}_i^q; \mathbf{w})}{\tau_1}, \nabla^2_{\lambda} L_q(\mathbf{w}, \hat{\lambda}_q(\mathbf{w})) \\ &= \frac{1}{N_q} \sum_{\mathbf{x}_i^q \in \mathcal{C}_q} \frac{1}{\tau_1} + \tau_2 \approx \frac{1}{N_q} \sum_{\mathbf{x}_i^q \in \mathcal{C}_q} \frac{1}{\tau_1} \end{aligned}$$

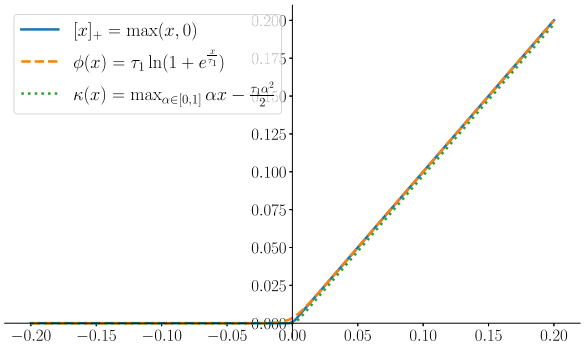

Fig. 12 Curves of $[\cdot]_+$, $\phi(\cdot)$, and $\kappa(\cdot)$

for sufficiently small τ_1, τ_2. Then we have

$$\frac{\nabla^2_{\mathbf{w}\lambda} L_q(\mathbf{w}, \lambda(\mathbf{w}))}{\nabla^2_{\lambda} L_q(\mathbf{w}, \lambda_q(\mathbf{w}))} = \frac{1}{|\mathcal{C}_q|} \sum_{\mathbf{x}_i^q \in \mathcal{C}_q} -\partial h_{\mathbf{w}}(\mathbf{x}_i^q).$$

Assume that ψ is chosen such that $\psi_i'(\mathbf{w}, \lambda_q(\mathbf{w})) \approx 0$ if $h_{\mathbf{w}}(\mathbf{x}_j^q) - \lambda_q(\mathbf{w}) \notin [0, \tau_1]$, and $\psi_i'(\mathbf{w}, \lambda_q(\mathbf{w})) \approx c_1$ and $f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)) \approx c_2$ if $h_{\mathbf{w}}(\mathbf{x}_j^q) - \lambda_q(\mathbf{w}) \in [0, \tau_1]$, then we have

$$\sum_{\mathbf{x}_i^q \in \mathcal{S}_q} \nabla_{\mathbf{w}} \psi_i f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q)) \approx c_1 c_2 \sum_{\mathbf{x}_i^q \in \mathcal{C}_q} \left(\nabla_{\mathbf{w}} h_{\mathbf{w}}(\mathbf{x}_i^q; \mathbf{w}) + \frac{1}{|\mathcal{C}_q|} \sum_{\mathbf{x}_j^q \in \mathcal{C}_q} -\nabla_{\mathbf{w}} h(\mathbf{x}_j^q; \mathbf{w}) \right) = 0$$

As a result, when τ_1 is small enough the condition $\psi_i'(\mathbf{w}, \lambda_q(\mathbf{w})) \approx 0$ if $h_{\mathbf{w}}(\mathbf{x}_j^q) - \lambda_q(\mathbf{w}) \notin [0, \tau_1]$, and $\psi_i'(\mathbf{w}, \lambda_q(\mathbf{w})) \approx c$ if $h_{\mathbf{w}}(\mathbf{x}_j^q) - \lambda_q(\mathbf{w}) \in [0, \tau_1]$ is well justified. An example of such $\psi(\cdot)$ is provided in the Fig. 13. As a result, with initial warm-up, we can compute the gradient estimator by

$$G(\mathbf{w}_t) = \frac{1}{|\mathcal{B}|} \sum_{(q, \mathbf{x}_i^q) \in \mathcal{B}} p_{q,i} \nabla \hat{g}_{q,i}(\mathbf{w}_t),$$

which simplifies K-SONG by avoiding maintaining and updating $s_{q,t}$.

Appendix C Algorithm implementation details

Our algorithms for optimizing top-K NDCG involve two second-order derivatives of the function L_q when computing the stochastic gradient estimators. In this section, we will describe how these quantities are calculated in our implementation.

Recall the formulation of L_q

$$L_q(\lambda_q^t(\mathbf{w}), \mathbf{w}_t; \mathcal{B}_t) = \frac{K + \epsilon}{N_q} \lambda + \frac{\tau_2}{2} \lambda^2 + \frac{1}{|\mathcal{B}_t|} \sum_{\mathbf{x}_i \in \mathcal{B}_t} \tau_1 \ln \left(1 + \exp \left(\frac{h_q(\mathbf{x}_i; \mathbf{w}) - \lambda}{\tau_1} \right) \right).$$

We first compute the first-order derivative of function L_q w.r.t. λ as follows

$$\nabla_\lambda L_q(\lambda_q^t(\mathbf{w}), \mathbf{w}_t; \mathcal{B}_t) = \frac{K+\epsilon}{N_q} + \tau_2\lambda + \frac{1}{|\mathcal{B}_t|}\sum_{\mathbf{x}_i\in\mathcal{B}_t}\left(-\frac{\exp\left(\frac{h_q(\mathbf{x}_i;\mathbf{w})-\lambda}{\tau_1}\right)}{1+\exp\left(\frac{h_q(\mathbf{x}_i;\mathbf{w})-\lambda}{\tau_1}\right)}\right).$$

Let $\sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$ denote the sigmoid function, then the above equation can be simplified to

$$\nabla_\lambda L_q(\lambda_q^t(\mathbf{w}), \mathbf{w}_t; \mathcal{B}_t) = \frac{K+\epsilon}{N_q} + \tau_2\lambda - \frac{1}{|\mathcal{B}_t|}\sum_{\mathbf{x}_i\in\mathcal{B}_t}\sigma\left(\frac{h_q(\mathbf{x}_i;\mathbf{w})-\lambda}{\tau_1}\right).$$

Further differentiating w.r.t. $\mathbf{w}$, we obtain

$$\nabla^2_{\lambda,\mathbf{w}} L_q(\lambda_q^t(\mathbf{w}), \mathbf{w}_t; \mathcal{B}_t) = \frac{1}{|\mathcal{B}_t|}\sum_{\mathbf{x}_i\in\mathcal{B}_t} -\sigma\left(\frac{h_q(\mathbf{x}_i;\mathbf{w})-\lambda}{\tau_1}\right)\left(1-\sigma\left(\frac{h_q(\mathbf{x}_i;\mathbf{w})-\lambda}{\tau_1}\right)\right)\frac{\nabla_\mathbf{w} h_q(\mathbf{x}_i;\mathbf{w})}{\tau_1},$$

where we employ the fact that $\sigma'(x) = \sigma(x)(1-\sigma(x))$.

From the above equation, it can be seen that the original second order derivative $\nabla^2_{\lambda,\mathbf{w}} L_q(\lambda_q^t(\mathbf{w}), \mathbf{w}_t; \mathcal{B}_t)$ equals to the first-order derivative $\nabla_\mathbf{w} h_q(\mathbf{x}_i;\mathbf{w})$ multiplied by a weight $-\sigma\left(\frac{h_q(\mathbf{x}_i;\mathbf{w})-\lambda}{\tau_1}\right)\left(1-\sigma\left(\frac{h_q(\mathbf{x}_i;\mathbf{w})-\lambda}{\tau_1}\right)\right)$. Therefore, in our implementation, we design the following loss function for calculating $\nabla^2_{\lambda,\mathbf{w}} L_q(\lambda_q^t(\mathbf{w}), \mathbf{w}_t; \mathcal{B}_t)$

$$\mathcal{L}_{\nabla^2_{\lambda,\mathbf{w}} L_q} = \frac{1}{|\mathcal{B}_t|}\sum_{\mathbf{x}_i\in\mathcal{B}_t}\mathbf{sg}\left(-\sigma\left(\frac{h_q(\mathbf{x}_i;\mathbf{w})-\lambda}{\tau_1}\right)\left(1-\sigma\left(\frac{h_q(\mathbf{x}_i;\mathbf{w})-\lambda}{\tau_1}\right)\right)\right)\frac{\nabla_\mathbf{w} h_q(\mathbf{x}_i;\mathbf{w})}{\tau_1},$$

where $\mathbf{sg}(\cdot)$ denotes the stop gradient operator. In this loss function, the weight component can be directly calculated, and the stop gradient operation is used to prevent gradient back-propagation. For $\nabla_\mathbf{w} h_q(\mathbf{x}_i;\mathbf{w})$, we set $h_q(\mathbf{x}_i;\mathbf{w})$ as the the differentiable part, allowing the automatic differentiation framework to compute $\nabla_\mathbf{w} h_q(\mathbf{x}_i;\mathbf{w})$. Thus, differentiating such loss function ultimately yields the desired second-order derivative.

The computation for the function $\nabla^2_{\lambda\lambda} L_q(\lambda_q^t(\mathbf{w}), \mathbf{w}_t; \mathcal{B}_t)$ is straightforward. We can directly derive its specific form:

$$\nabla^2_{\lambda\lambda} L_q(\lambda_q^t(\mathbf{w}), \mathbf{w}_t; \mathcal{B}_t) = \tau_2 + \frac{1}{|\mathcal{B}_t|}\sum_{\mathbf{x}_i\in\mathcal{B}_t}\sigma\left(\frac{h_q(\mathbf{x}_i;\mathbf{w})-\lambda}{\tau_1}\right)\left(1-\sigma\left(\frac{h_q(\mathbf{x}_i;\mathbf{w})-\lambda}{\tau_1}\right)\right)\frac{1}{\tau_1}.$$

Appendix D Experiments

MSLR-WEB30K[5] and Yahoo! LTR dataset[6] are the largest public LTR datasets from commercial English search engines. We provide the statistics of these two datasets in Table 5. In MSLR-WEB30K dataset, there are 5 folds containing the same data, and each fold randomly splits to training, validation, and test sets. Due to privacy concerns, these datasets do not disclose any text information and only provide feature vectors for each query-document

[5] https://www.microsoft.com/en-us/research/project/mslr/

[6] https://webscope.sandbox.yahoo.com

Table 5 Statistics of learning to rank datasets

Dataset	MSLR-WEB30K	Yahoo! LTR dataset
Query	30,000	29,921
Q-D pair	3,771,125	709,877
max Q-D pair per query	1,245	135
min Q-D pair per query	1	1

Table 6 Statistics of recommender systems datasets

Dataset	# users	# items	# interactions	sparsity
MovieLens20M	138,493	26,744	20,000,263	99.46%
Netflix Prize dataset	236,117	17,770	89,973,534	97.86%

pair. For these two LTR datasets, we standarize the features, log-transforming selected ones, before feeding them to the learning algorithms. Since the lengths of search results lists in the datasets are unequal, we truncate or pad samples to the length of 40 and 100 for Yahoo! LTR dataset and MSLR-WEB30K when training, respectively, but use the full list for evaluation.

MovieLens20M[7] contains 20 million ratings applied to 27,000 movies by 138,000 users, and all users have rated at least 20 movies. Netflix Prize dataset[8] consists of about 100,000,000 ratings for 17,770 movies given by 480,189 users. We filter the Netflix Prize dataset by retaining users with at least 100 interactions to cater sufficient information for modeling. In both datasets, users and movies are represented with integer IDs, while ratings range from 1 to 5. The statistics of these two datasets are shown in Table 6.

For the experiments on two LTR datasets, we adopt allRank framework Pobrotyn et al. (2020). We implement some baseline methods based on their code. For the recommender systems experiments, we use ReChorus framework Wang et al. (2020), and follow the scripts in ReChorus to preprocess the datasets. We train our models on one Tesla V100 GPU with 32GB memory. The training on the Context-Aware Ranker model takes about 2~3 hours for convergence, while the training of the NeuMF model takes about 8~12 hours for convergence. For the experiments on two molecular datasets, we adopt the code base from OGB[9].

More experimental results are provided in this section. The full ablation studies on four datasets are presented in Fig. 14. We provide the negative log-likelihood loss curves of three different training methods for warm-up in Fig. 15. We also study the effect of varying γ_0 and report the training curves of warm-up and SONG in Fig. 16. We observe that $\gamma_0 = 0.1$ achieves the best performance in most cases. Setting $\gamma_0 = 1.0$ is equivalent to update the model with a biased stochastic gradient, which leads to the worst performance. These results signify the importance of moving average estimators in our methods.

[7] https://grouplens.org/datasets/movielens/20m/

[8] https://www.kaggle.com/netflix-inc/netflix-prize-data

[9] https://github.com/snap-stanford/ogb/tree/master/examples/graphproppred/mol

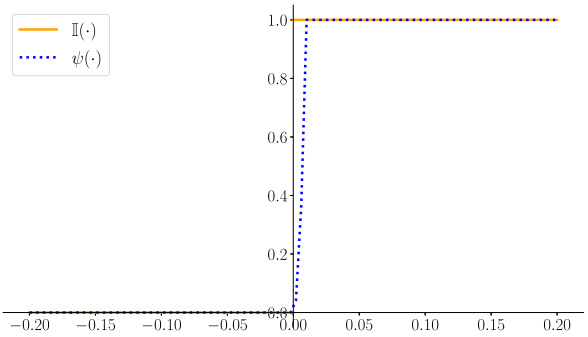

Fig. 13 An example of $\psi(\cdot)$

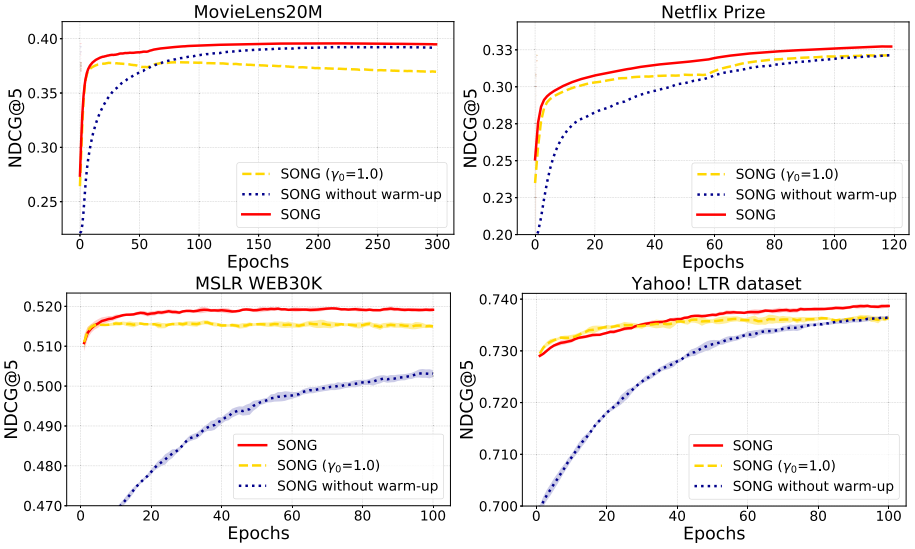

Fig. 14 Ablation study on two variants of SONG on four different datasets

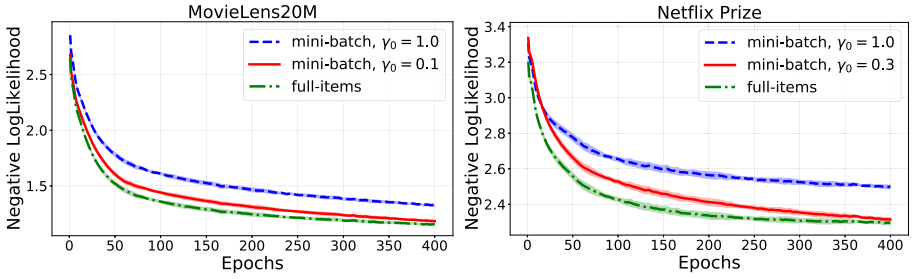

Fig. 15 Comparison of full-items and mini-batch training

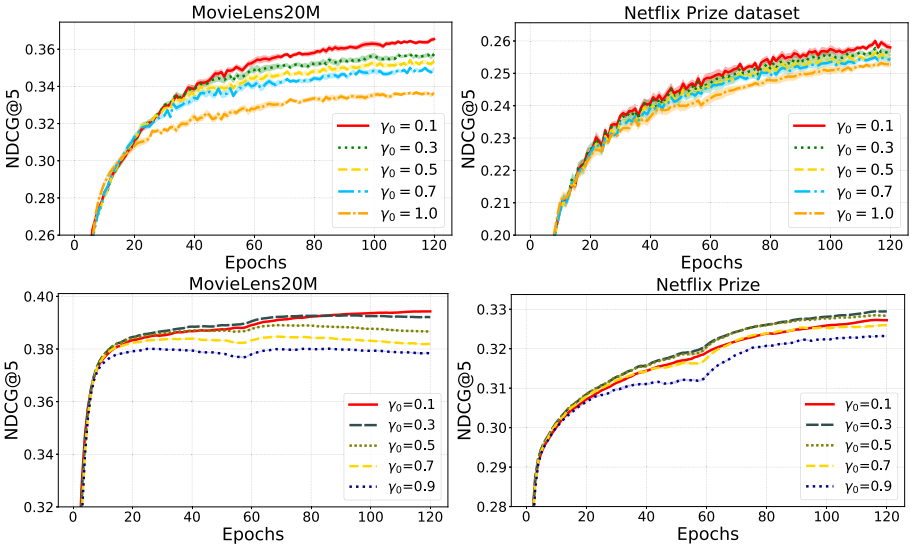

Fig. 16 The effect of varying γ_0 for warm-up (top two) and SONG (bottom two)

Appendix E Convergence analysis for SONG and K-SONG

As we point out in Sect. 3, NDCG can be seen as a special case of top-K NDCG. From the perspective of optimization problem, if we set the $\psi(\cdot)$ function in problem (6) for optimizing top-K NDCG as a constant function, the problem will reduce to problem (3) for optimizing NDCG. Hence, Theorem 1 naturally follows from Theorem 2, of which the proof will be presented in the this section.

Before analyzing the convergence for K-SONG, to simplify the notations, we first reorder the set of $\mathcal{S}$ so that each pair $(q, \mathbf{x}_i^q)$ has a new single index i, and we abuse the notation $\mathcal{S}$ denoting the set of the new indexing . Then we employ $\psi_i(\mathbf{w}, \lambda_q(\mathbf{w}))$ and $f_i(g_i(\mathbf{w}))$ to represent $\psi(h_q(\mathbf{x}_i^q; \mathbf{w}) - \hat{\lambda}_q(\mathbf{w}))$ and $f_{q,i}(g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q))$, respectively. Now the compositional bilevel optimization problem becomes:

$$\begin{aligned} \min_{\mathbf{w}} \quad & F(\mathbf{w}) := \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \psi_i(\mathbf{w}, \lambda_q(\mathbf{w})) f_i(g_i(\mathbf{w})) \\ s.t. \quad & \lambda_q(\mathbf{w}) = \arg\min_{\lambda} L_q(\mathbf{w}, \lambda), \forall q \in \mathcal{Q}, \end{aligned} \tag{25}$$

which allows us to restate K-SONG as Algorithm 8 accordingly. Throughout this convergence analysis section, all subscript q represents the variable or function corresponding to query q. The following auxiliary notations will be used:

$$\delta_{\lambda,t} := \|\lambda(\mathbf{w}_t) - \lambda^t\|^2, \quad \delta_{g,t} := \|g(\mathbf{w}_t) - \mathbf{u}^t\|^2, \quad \delta_{L\lambda\lambda,t} := \|\nabla^2_{\lambda\lambda} L(\lambda(\mathbf{w}_t); \mathbf{w}_t) - \mathbf{s}^t\|^2.$$

Besides, we employ $\mathbb{E}[\cdot]$ to represent the expectation over the randomness of the algorithm until the current iteration, and $\mathbb{E}_t[\cdot]$ to denote the expectation over the randomness at iteration t. We make the following assumptions regarding problem (25).

Assumption 1 (i) Functions ψ_i, f_i, g_i are L_ψ, L_f, L_g-smooth and C_ψ, C_f, C_g-Lipschitz continuous respectively for all i.

Algorithm 8 Restate K-SONG with new indexing

Require: $\mathbf{w}_0, \mathbf{m}_0 = \lambda^0 = \mathbf{u}^0 = \mathbf{s}^0 = 0, \gamma_0, \gamma_0', \beta_1, \eta_0, \eta_1$
Ensure: $\mathbf{w}_T$
for $t = 0, 1, \ldots, T-1$ **do**
  Draw some relevant Q-I pairs $\mathcal{B}^t = \{(q, \mathbf{x}_i^q)\} \subset \mathcal{S}$
  For each $q \in \mathcal{B}^t$ draw a batch of items $\mathcal{B}_q^t \subset \mathcal{S}_q$
  Compute $\mathbf{u}_i^{t+1} = \begin{cases} (1-\gamma_0)\mathbf{u}_i^t + \gamma_0 g_i(\mathbf{w}_t; \mathcal{B}_q^t) & \text{if } i \in \mathcal{B}^t \\ \mathbf{u}_i^t & \text{o.w.} \end{cases}$
  Compute $\lambda_q^{t+1} = \begin{cases} \lambda_q^t - \eta_0 \nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t; \mathcal{B}_q^t) & \text{if } q \in \mathcal{B}^t \\ \lambda_q^t & \text{o.w.} \end{cases}$
  Compute $\mathbf{s}_q^{t+1} = \begin{cases} (1-\gamma_0')\mathbf{s}_q^t + \gamma_0' \nabla^2_{\lambda\lambda} L_q(\mathbf{w}_t, \lambda_q^t; \mathcal{B}_q^t) & \text{if } q \in \mathcal{B}^t \\ \mathbf{s}_q^t & \text{o.w.} \end{cases}$
  Compute stochastic gradient estimator $G(\mathbf{w}_t)$ according to (26)
  $\mathbf{m}_{t+1} = \beta_1 \mathbf{m}_t + (1-\beta_1) G(\mathbf{w}_t)$
  $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_1 \mathbf{m}_{t+1}$
end for

(ii) Functions ψ_i and f_i are bounded by B_ψ and B_f respectively, i.e., $||\psi_i(\mathbf{w}, \lambda)|| \le B_\psi$ and $||f_i(g)|| \le B_f$ for all $\mathbf{w}, \lambda, i, g$.
(iii) Functions L_q are L_L-smooth and μ_L-strongly convex, i.e., $L_L I \succeq \nabla^2_{\lambda\lambda} L_q(\mathbf{w}, \lambda; \mathcal{B}) \succeq \mu_L I$, for all q.
(iv) Unbiased stochastic oracles $g_i, \nabla g_i, \nabla_\lambda L_q, \nabla^2_{\lambda\lambda} L_q, \nabla^2_{\mathbf{w}\lambda} L_q$ have bounded variance σ^2.
(v) $||\nabla^2_{\mathbf{w}\lambda} L_q(\mathbf{w}, \lambda)||^2 \le C^2_{L\mathbf{w}\lambda}$, $\nabla_\lambda L_q(\mathbf{w}, \lambda)$, $\nabla^2_{\mathbf{w}\lambda} L_q(\mathbf{w}, \lambda)$, $\nabla^2_{\lambda\lambda} L_q(\mathbf{w}, \lambda)$ are $L_{L\lambda}$, $L_{L\mathbf{w}\lambda}$, $L_{L\lambda\lambda}$-Lipschitz continuous respectively with respect to $(\mathbf{w}, \lambda)$ for all q.

Remark 1 For *(i)* and *(ii)*, we consider the squared hinge loss $\ell(h_q(\mathbf{x}'; \mathbf{w}), h_q(\mathbf{x}; \mathbf{w})) = \max\{0, h_q(\mathbf{x}'; \mathbf{w}) - h_q(\mathbf{x}; \mathbf{w}) + c\}^2$ where c is a margin parameter. Suppose the score function and its gradients $h_q(\mathbf{x}; \mathbf{w})$, $\nabla_{\mathbf{w}} h_q(\mathbf{x}; \mathbf{w})$, $\nabla^2_{\mathbf{w}} h_q(\mathbf{x}; \mathbf{w})$ are bounded by finite constants $c_h, c_{h'}, c_{h''}$ respectively. As an average of squared hinge loss, function $g_i(\mathbf{w})$ in (25) has bounded gradients $\nabla g_i(\mathbf{w}) \le 8 c_h c_{h'}$ and $\nabla^2 g_i(\mathbf{w}) \le 8 c_{h'}^2 + 8 c_h c_{h''}$ for each $i \in \mathcal{S}$. Hence g_i is Lipschitz continuous and smooth. Moreover, with $m > 2c_h$, there exists $c_\ell > 0$ such that $\ell(h_q(\mathbf{x}_1; \mathbf{w}) - h_q(\mathbf{x}_2; \mathbf{w})) \ge c_\ell$ for all $\mathbf{x}_1, \mathbf{x}_2$. Function $f_i(g) = f_{q,i}(g) = \frac{1}{Z_q} \frac{1 - 2^{y_i^q}}{\log_2(N_q g + 1)}$ is thus bounded, Lipschitz continuous and smooth for each $i = (q, \mathbf{x}_i^q) \in \mathcal{S}$. For function $\psi_i = \psi(h_q(\mathbf{x}_i^q; \mathbf{w}) - \lambda_q(\mathbf{w}))$, we consider the logistic loss, then ψ_i is naturally bounded. Next, we consider the Lipschitz continuity and smoothness of ψ_i. Since the lower-level problem L_q in (25) is smooth and strongly convex, according to Lemma 4.3 proved by Lin et al. (2019), $\lambda_q(\mathbf{w})$ is Lipschitz continuous. Additionally, by leveraging the assumption that $h_q(\mathbf{x}; \mathbf{w})$, $\nabla_{\mathbf{w}} h_q(\mathbf{x}; \mathbf{w})$, $\nabla^2_{\mathbf{w}} h_q(\mathbf{x}; \mathbf{w})$ are bounded, we can verify the smoothness of $\lambda_q(\mathbf{w})$ by calculating the bound of its second-order derivative. Finally, using the above properties, we can compute the bounds for $\|\nabla\psi(h_q(\mathbf{x}_i^q; \mathbf{w}) - \lambda_q(\mathbf{w}))\|$ and $\|\nabla\psi(h_q(\mathbf{x}_i^q; \mathbf{w}_1) - \lambda_q(\mathbf{w}_1)) - \nabla\psi(h_q(\mathbf{x}_i^q; \mathbf{w}_2) - \lambda_q(\mathbf{w}_2))\|$ to verify that ψ_i is Lipschitz continuous and smooth.

Remark 2 Assumption *(iii)* is made in many existing works for SBO (Ghadimi & Wang, 2018; Hong et al., 2023; Chen et al., 2022). We prove the smoothness and strong convexity of L_q in Lemma 4. The strong convexity of L_q implies the lower bound $\gamma = \tau_2$ of $\nabla_{\lambda\lambda} L_q(\mathbf{w}, \lambda; \mathcal{B})$. Assumption *(iv)* is also standard in the literature (Chen et al., 2022; Guo et al., 2021a).

Remark 3 For assumption *(v)*, one can verify the Lipschitz continuity of $\nabla^2_{\mathbf{w}\lambda} L_q(\lambda; \mathbf{w})$ and $\nabla^2_{\lambda\lambda} L_q(\lambda; \mathbf{w})$ by taking the third-order gradients w.r.t. $L_q(\lambda; \mathbf{w})$ and using $\exp\left(\frac{\lambda - h_q(\mathbf{x}_i; \mathbf{w})}{\tau_1}\right) > 0$ and the assumption of the boundedness of $h_q(\mathbf{x}; \mathbf{w})$ and its gradients.

By using the implicit function theorem, the stochastic gradient estimator of $\nabla F(\mathbf{w}_t)$ in Algorithm 8, i.e., $G(\mathbf{w}_t)$, is given by:

$$\frac{1}{|\mathcal{B}^t|}\sum_{i\in\mathcal{B}^t} G_i(\mathbf{w}_t) = \frac{1}{|\mathcal{B}^t|}\sum_{i\in\mathcal{B}^t}\Big\{\Big[\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t, \lambda_q^t) - \nabla^2_{\mathbf{w}\lambda} L_q(\mathbf{w}_t, \lambda_q^t; \mathcal{B}_q^t)[\mathbf{s}_q^t]^{-1}\nabla_\lambda \psi_i(\mathbf{w}_t, \lambda_q^t)\Big] f_i(\mathbf{u}_i^t) + \psi_i(\mathbf{w}_t, \lambda_q^t)\nabla g_i(\mathbf{w}_t; \mathcal{B}_q^t)\nabla f_i(\mathbf{u}_i^t)\Big\}. \tag{26}$$

Now we restate Theorem 2 as follows to include the specifics of the parameters.

Theorem 4 (Restate Theorem 2 with parameter specifics). *Let $F(\mathbf{w}_0) - F(\mathbf{w}^*) \le \Delta_F$ and Assumption 1 hold. Apply K-SONG in Algorithm 8 to solve the problem (25) with the following parameters:*

$$\eta_0 \le \min\left\{\frac{\mu_L}{L_L^2}, \frac{2N}{|\mathcal{B}|\mu_L}, \frac{\mu_L\epsilon^2}{48C_{10}\sigma^2}\right\}, \gamma_0 \le \left\{\frac{1}{2}, \frac{\epsilon^2}{96C_6\sigma^2}\right\}, \beta_1 \ge 1 - \frac{\epsilon^2}{12\left(\frac{C_8}{|\mathcal{B}|} + C_9\sigma^2\right)},$$

$$\gamma_0' \le \left\{1, \frac{\epsilon^2}{96C_7\sigma^2}\right\}, \eta_1^2 \le \min\left\{\frac{\gamma_1^2}{32L_F^2}, \frac{|\mathcal{B}|^2\eta_0^2\mu_L^2}{64N^2C_{10}C_\lambda^2}, \frac{|\mathcal{B}|^2\gamma_0^2}{64|\mathcal{S}|^2C_6C_g^2}, \frac{|\mathcal{B}|^2\gamma_0'^2}{256N^2C_7L_{L\lambda\lambda}^2(1+C_\lambda^2)}\right\},$$

$$T \ge \left\{\frac{30\Delta_F}{\eta_1\epsilon^2}, \frac{15\mathbb{E}[\|\nabla F(\mathbf{w}_0) - \mathbf{m}_1\|^2]}{\gamma_1\epsilon^2}, \frac{30C_{10}\delta_{\lambda,0}}{|\mathcal{B}|\eta_0\mu_L\epsilon^2}, \frac{30C_6\delta_{g,0}}{|\mathcal{B}|\gamma_0\epsilon^2}, \frac{60C_7\delta_{L\lambda\lambda,0}}{|\mathcal{B}|\gamma_0'\epsilon^2}\right\}$$

We have

$$\mathbb{E}[\|\nabla F(\mathbf{w}_\tau)\|^2] \le \epsilon^2$$

where τ is randomly sampled from $\{0, \ldots, T\}$, $C_6, C_7, C_8, C_9, C_{10}$ are constants defined in the proof, and L_F is the Lipschitz continuity constant of $\nabla F(\mathbf{w})$.

To prove Theorem 4, we first present some required Lemmas.

Lemma 5 *Under assumption 1, $F(\mathbf{w})$ is L_F-smooth for some constant $L_F \in \mathbb{R}$.*

Lemma 6 *Consider the update $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_1\mathbf{m}_{t+1}$. Then under assumption 1, with $\eta_1 L_F \le \frac{1}{2}$, we have*

$$F(\mathbf{w}_{t+1}) \le F(\mathbf{w}_t) + \frac{\eta_1}{2}\|\nabla F(\mathbf{w}_t) - \mathbf{m}_{t+1}\|^2 - \frac{\eta_1}{2}\|\nabla F(\mathbf{w}_t)\|^2 - \frac{\eta_1}{4}\|\mathbf{m}_{t+1}\|^2.$$

Lemma 7 (Lemma 4.3 Lin et al. (2019)). *Under assumption 1, $\lambda_q(\mathbf{w})$ is C_λ-Lipschitz continuous with $C_\lambda = L_L/\mu_L$ for all $q = 1, \ldots, N$.*

Lemma 8 *Consider the updates in algorithm 8, under assumption 1, with $\eta_0 \le \min\{\mu_L/L_L^2, \frac{2N}{|\mathcal{B}|\mu_L}\}$ we have*

$$\sum_{t=0}^{T}\mathbb{E}[\delta_{\lambda,t}] \le \frac{2N}{|\mathcal{B}|\eta_0\mu_L}\delta_{\lambda,0} + \frac{4N\eta_0 T\sigma^2}{\mu_L} + \frac{8N^3C_\lambda^2\eta_1^2}{|\mathcal{B}|^2\eta_0^2\mu_L^2}\sum_{t=0}^{T-1}\mathbb{E}[\|\mathbf{m}_{t+1}\|^2] \tag{27}$$

Lemma 9 *Consider algorithm 8, under assumption 1, with $\gamma_0 < 1/2$ we have*

$$\sum_{t=0}^{T}\mathbb{E}[\delta_{g,t}] \leq \frac{2|\mathcal{S}|}{|\mathcal{B}|\gamma_0}\delta_{g,0} + 8|\mathcal{S}|\gamma_0\sigma^2 T + \frac{8|\mathcal{S}|^3 C_g^2\eta_1^2}{|\mathcal{B}|^2\gamma_0^2}\sum_{t=0}^{T-1}\mathbb{E}[\|\mathbf{m}_{t+1}\|^2] \tag{28}$$

Lemma 10 *Consider algorithm 8, under assumption 1, with $\gamma_0' \leq 1$ we have*

$$\sum_{t=0}^{T}\mathbb{E}[\delta_{L\lambda\lambda,t}] \leq \frac{4N}{|\mathcal{B}|\gamma_0'}\delta_{L\lambda\lambda,0} + 32L_{L\lambda\lambda}^2\sum_{t=0}^{T-1}\mathbb{E}[\delta_{\lambda,t}] + 8N\gamma_0' T\sigma^2 + \frac{32N^3L_{L\lambda\lambda}^2(1+C_\lambda^2)\eta_1^2}{|\mathcal{B}|^2\gamma_0'^2}\sum_{t=0}^{T-1}\mathbb{E}[\|\mathbf{m}_{t+1}\|^2]$$

Appendix E.1.1 Proof sketch

Before presenting the formal proof, we first outline the intuition behind it. Here are several key points of the proof:

1. Starting with Lemma 6, it is evident that the quality of the final solution (denoted as $\|\nabla F(\mathbf{w}_t)\|$) is related to the approximation error of the stochastic gradient estimator $\mathbf{m}_t$ during the optimization (denoted as $\|\nabla F(\mathbf{w}_t) - \mathbf{m}_{t+1}\|$).
2. In the proof, we begin by decomposing $\|\nabla F(\mathbf{w}_t) - \mathbf{m}_{t+1}\|$ and demonstrating that it can be bounded by the approximation errors of several crucial inner functions (refer to (31)).
3. Subsequently, by incorporating the approximation errors of these inner functions (as stated in Lemma 8, 9, and 10), we can derive the detailed bound for $\|\nabla F(\mathbf{w}_t)\|$ (refer to (36)).
4. Finally, by setting appropriate parameters, we obtain the expression for the number of iterations required to achieve an ϵ-stationary point (see the end of the following proof). This completes our proof.

Appendix E.2.2 Innovations in proof techniques

First, we analyze the SONG algorithm. Inspired by the average precision maximization algorithm SOAP proposed by Qi et al. (2021), SONG uses a moving average estimator and establishes a convergence rate of $\mathcal{O}(\epsilon^{-4})$, which is better than the $\mathcal{O}(\epsilon^{-5})$ convergence rate established by SOAP. We attribute this improvement to the use of a simple yet effective momentum-style stochastic gradient estimator $\mathbf{m}_t$ and a more refined analysis. Specifically, we can focus on the term $\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2$ in the proofs of both SOAP and SONG. In Lemma 2 of SOAP, $\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2$ is simply bounded by a gradient norm constant. However, in Lemma 6 of our SONG, however, we first use the definition of $\mathbf{m}_t$ and perform an equivalent transformation on the cross term $\nabla F(\mathbf{w}_t)^T(\mathbf{w}_{t+1} - \mathbf{w}_t)$. This allows us to transform $\nabla F(\mathbf{w}_t)^T(\mathbf{w}_{t+1} - \mathbf{w}_t)$ and $\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2$ into $\|\nabla F(\mathbf{w}_t) - \mathbf{m}_{t+1}\|^2$ and a **negative** $\|\mathbf{m}_{t+1}\|^2$ term. This negative term then can cancel out the positive $\|\mathbf{m}_{t+1}\|^2$ term that appears in the bound of $\|\nabla F(\mathbf{w}_t) - \mathbf{m}_{t+1}\|^2$ (refer to (36)), thereby tightening the bound and ultimately improving the convergence rate.

For K-SONG, the key difference from Guo et al. (2021a) is our implementation of parallel speed-up in lines 9–12 of Algorithm 2 and the introduction of new proof techniques to control the estimation error of lower-level solutions. The core steps of our proof are as follows. First, we denote $\tilde{\lambda}_q^t$ as the lower-level solutions updated in the t-th iteration, and use its update rule to derive the estimation error bound $\|\tilde{\lambda}_q^t - \lambda_q(\mathbf{w}_t)\|^2$

$$\mathbb{E}_t[\|\tilde{\lambda}_q^t - \lambda_q(\mathbf{w}_t)\|^2] \leq (1 - \Theta(\eta_0))\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2 + \Theta(\eta_0^2\sigma^2),$$

where we employ $\Theta(\cdot)$ to omit some constants. Then, using the property of conditional expectation, we can establish the estimation error for all lower-level solutions in the t-th iteration

$$\mathbb{E}_t[\|\lambda_q^{t+1} - \lambda_q(\mathbf{w}_t)\|^2] = \frac{|\mathcal{B}|}{N}\mathbb{E}_t[\|\tilde{\lambda}_q^t - \lambda_q(\mathbf{w}_t)\|^2] + \frac{N-|\mathcal{B}|}{N}\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2.$$

At last, using Young's inequality and the above bounds, we can derive the final recursive error bound for all lower-level solutions:

$$\begin{aligned}\mathbb{E}_t[\|\lambda_q^{t+1} - \lambda_q(\mathbf{w}_{t+1})\|^2] &\le (1+\Theta(\eta_0))\mathbb{E}_t[\|\lambda_q^{t+1} - \lambda_q(\mathbf{w}_t)\|^2] + \left(1+\frac{1}{\Theta(\eta_0)}\right)\mathbb{E}_t[\|\lambda_q(\mathbf{w}_{t+1}) - \lambda_q(\mathbf{w}_t)\|^2] \\ &\overset{(*)}{\le} (1-\Theta(\eta_0))\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2 + \Theta(\eta_0^2\sigma^2) + \frac{1}{\Theta(\eta_0)}\mathbb{E}_t[\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2],\end{aligned}$$

where $(*)$ holds under certain parameter settings. Details can be found in the proof of Lemma 8.

Proof of Theorem 4 In proving algorithm convergence, the most critical aspect is establishing the error bound between the stochastic gradient estimator $\mathbf{m}_{t+1} = (1-\gamma_1)\mathbf{m}_t + \gamma_1 G(\mathbf{w}_t)$ and the ground truth gradient $\nabla F(\mathbf{w}_t)$ in the algorithm. We begin the proof by analyzing the error bound for $\|\nabla F(\mathbf{w}_t) - \mathbf{m}_{t+1}\|^2$. Recall that

$$\begin{aligned}\nabla F(\mathbf{w}_t) = \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\Big[&\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t, \lambda_q(\mathbf{w}_t)) \\ &- \nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))]^{-1}\nabla_\lambda\psi_i(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))\Big] f_i(g_i(\mathbf{w}_t)) \\ &+ \psi_i(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))\nabla g_i(\mathbf{w}_t)\nabla f_i(g_i(\mathbf{w}_t)), \\ G(\mathbf{w}_t) = \frac{1}{|\mathcal{B}^t|}\sum_{i\in\mathcal{B}^t}\Big[&\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t, \lambda_q^t) - \nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_t, \lambda_q^t; \mathcal{B}_q^t)[\mathbf{s}_q^t]^{-1}\nabla_\lambda\psi_i(\mathbf{w}_t, \lambda_q^t)\Big] f_i(\mathbf{u}_i^t) \\ &+ \psi_i(\mathbf{w}_t, \lambda_q^t)\nabla g_i(\mathbf{w}_t; \mathcal{B}_q^t)\nabla f_i(\mathbf{u}_i^t).\end{aligned}$$

One can observe that $\nabla F(\mathbf{w}_t)$ and $G(\mathbf{w}_t)$ differ significantly in form. To assist in our proof, we define the following auxiliary function $\nabla F(\mathbf{w}_t, \lambda^t)$:

$$\begin{aligned}\nabla F(\mathbf{w}_t, \lambda^t) &= \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\nabla F_i(\mathbf{w}_t, \lambda^t) \\ &:= \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\Big[\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t, \lambda_q^t) - \nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_t, \lambda_q^t)[\mathbf{s}_q^t]^{-1}\nabla_\lambda\psi_i(\mathbf{w}_t, \lambda_q^t)\Big] f_i(\mathbf{u}_i^t) \\ &\quad + \psi_i(\mathbf{w}_t, \lambda_q^t)\nabla g_i(\mathbf{w}_t)\nabla f_i(\mathbf{u}_i^t).\end{aligned}$$

Note that the difference between $\nabla F(\mathbf{w}_t, \lambda^t)$ and $G(\mathbf{w}_t)$ lies in that $\nabla F(\mathbf{w}_t, \lambda^t)$ is computed over all samples $|\mathcal{S}|$, while $G(\mathbf{w}_t)$ is computed on a mini-batch $\mathcal{B}^t$. The distinction between $\nabla F(\mathbf{w}_t)$ and $\nabla F(\mathbf{w}_t, \lambda^t)$ is that $\nabla F(\mathbf{w}_t, \lambda^t)$ substitutes the estimators $\mathbf{s}_q^t$, $\mathbf{u}_i^t$, and λ_q^t for $\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))$, $g_i(\mathbf{w}_t)$, and $\lambda_q(\mathbf{w}_t)$ used in $\nabla F(\mathbf{w}_t)$, respectively.

We can now employ the update rule $\mathbf{m}_{t+1} = (1-\gamma_1)\mathbf{m}_t + \gamma_1 G(\mathbf{w}_t)$ in Algorithm 8, where $\gamma_1 = 1-\beta_1$, and establish the following error bound:

$$\begin{aligned}&\mathbb{E}_t[\|\nabla F(\mathbf{w}_t) - \mathbf{m}_{t+1}\|^2] = \mathbb{E}_t[\|\nabla F(\mathbf{w}_t) - (1-\gamma_1)\mathbf{m}_t - \gamma_1 G(\mathbf{w}_t)\|^2] \\ &= \mathbb{E}_t[\|(1-\gamma_1)(\nabla F(\mathbf{w}_{t-1}) - \mathbf{m}_t) + (1-\gamma_1)(\nabla F(\mathbf{w}_t) - \nabla F(\mathbf{w}_{t-1})) + \gamma_1(\nabla F(\mathbf{w}_t) - \nabla F(\mathbf{w}_t, \lambda^t))\end{aligned}$$

$$
\begin{aligned}
&+\gamma_1(\nabla F(\mathbf{w}_t,\lambda^t)-G(\mathbf{w}_t))\|^2]\\
&\overset{(a)}{=}\mathbb{E}_t[\|(1-\gamma_1)(\nabla F(\mathbf{w}_{t-1})-\mathbf{m}_t)+(1-\gamma_1)(\nabla F(\mathbf{w}_t)-\nabla F(\mathbf{w}_{t-1}))+\gamma_1(\nabla F(\mathbf{w}_t)-\nabla F(\mathbf{w}_t,\lambda^t))\|^2\\
&+\|\gamma_1(\nabla F(\mathbf{w}_t,\lambda^t)-G(\mathbf{w}_t))\|^2]\\
&\overset{(b)}{\leq}(1+\gamma_1)(1-\gamma_1)^2\|\nabla F(\mathbf{w}_{t-1})-\mathbf{m}_t\|^2+\gamma_1^2\mathbb{E}_t[\|\nabla F(\mathbf{w}_t,\lambda^t)-G(\mathbf{w}_t)\|^2]\\
&+2\left(1+\frac{1}{\gamma_1}\right)\left[\|\nabla F(\mathbf{w}_t)-\nabla F(\mathbf{w}_{t-1})\|^2+\gamma_1^2\|\nabla F(\mathbf{w}_t)-\nabla F(\mathbf{w}_t,\lambda^t)\|^2\right]\\
&\leq(1-\gamma_1)\|\nabla F(\mathbf{w}_{t-1})-\mathbf{m}_t\|^2+2\left(1+\frac{1}{\gamma_1}\right)\Big[L_F^2\|\mathbf{w}_t-\mathbf{w}_{t-1}\|^2+\gamma_1^2\underbrace{\|\nabla F(\mathbf{w}_t)-\nabla F(\mathbf{w}_t,\lambda^t)\|^2}_{ⓐ}\Big]\\
&+\gamma_1^2\underbrace{\mathbb{E}_t[\|\nabla F(\mathbf{w}_t,\lambda^t)-G(\mathbf{w}_t)\|^2]}_{ⓑ},
\end{aligned}
\tag{29}
$$

where $\mathbb{E}_t[\cdot]$ takes expectation over the randomness at iteration t, auxiliary function $\nabla F(\mathbf{w}_t,\lambda^t)$ is introduced in the second equality, inequality (a) follows from $\mathbb{E}_t[G(\mathbf{w}_t)]=\nabla F(\mathbf{w}_t,\lambda^t)$, (b) is due to $\|a+b\|^2\leq(1+\beta)\|a\|^2+(1+\frac{1}{\beta})\|b\|^2$, and the last inequality is due to $(1+\gamma_1)(1-\gamma_1)<1$.

Next, we establish the error bounds for ⓐ and ⓑ in (29). The core idea is to first expand the functions according to their definitions, then decompose the resulting expressions using the inequality $||\mathbf{x}_1+\cdots+\mathbf{x}_n||^2\leq n||\mathbf{x}_1||^2+\cdots+n||\mathbf{x}_n||^2$. For ⓐ, we achieve

$$
\begin{aligned}
ⓐ&=\mathbb{E}_t[\|\nabla F(\mathbf{w}_t)-\nabla F(\mathbf{w}_t,\lambda^t)\|^2]\\
&\leq\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}6\|\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))[f_i(g_i(\mathbf{w}_t))-f_i(\mathbf{u}_i^t)]\|^2\\
&+6\|[\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))-\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t,\lambda_q^t)]f_i(\mathbf{u}_i^t)\|^2\\
&+12\|[\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))-\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_t,\lambda_q^t)\mathbf{I}\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))]^{-1}\\
&\nabla_\lambda\psi_i(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))f_i(g_i(\mathbf{w}_t))\|^2\\
&+12\|\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_t,\lambda_q^t)[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))]^{-1}[\nabla_\lambda\psi_i(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))-\nabla_\lambda\psi_i(\mathbf{w}_t,\lambda_q^t)]f_i(g_i(\mathbf{w}_t))\|^2\\
&+12\|\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_t,\lambda_q^t)[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))]^{-1}\nabla_\lambda\psi_i(\mathbf{w}_t,\lambda_q^t)[f_i(g_i(\mathbf{w}_t))-f_i(\mathbf{u}_i^t)]\|^2\\
&+12\|\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_t,\lambda_q^t)\big[[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))]^{-1}-[\mathbf{s}_q^t]^{-1}\big]\nabla_\lambda\psi_i(\mathbf{w}_t,\lambda_q^t)f_i(\mathbf{u}_i^t)\|^2\\
&+6\|[\psi_i(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))-\psi_i(\mathbf{w}_t,\lambda_q^t)]\nabla g_i(\mathbf{w}_t)\nabla f_i(g_i(\mathbf{w}_i))\|^2\\
&+6\|\psi_i(\mathbf{w}_t,\lambda_q^t)\nabla g_i(\mathbf{w}_t)[\nabla f_i(g_i(\mathbf{w}_i))-\nabla f_i(\mathbf{u}_i^t)]\|^2.
\end{aligned}
$$

To further bound the RHS of the above inequality, we leverage the Lipschitz continuity and smoothness properties of the relevant functions assumed in Assumption 1, yielding the following bound:

$$
\begin{aligned}
ⓐ&\leq\left(\frac{6C_\psi^2C_f^2}{|\mathcal{S}|}+\frac{12C_{L\mathbf{w}\lambda}^2C_\psi^2C_f^2}{\mu_L^2|\mathcal{S}|}+\frac{6B_\psi^2C_g^2L_f^2}{|\mathcal{S}|}\right)\|g(\mathbf{w}_t)-\mathbf{u}^t\|^2\\
&+\frac{12C_{L\mathbf{w}\lambda}^2C_\psi^2B_f^2}{\mu_L^2\gamma^2N}\|\nabla^2_{\lambda\lambda}L(\mathbf{w}_t,\lambda(\mathbf{w}_t))-\mathbf{s}^t\|^2\\
&+\left(\frac{6L_\psi^2B_f^2}{N}+\frac{12L_{L\mathbf{w}\lambda}^2C_\psi^2B_f^2}{\mu_L^2N}+\frac{12C_{L\mathbf{w}\lambda}^2L_\psi^2B_f^2}{\mu_L^2N}+\frac{6C_g^2C_f^2}{N}\right)\|\lambda(\mathbf{w}_t)-\lambda^t\|^2\\
&=:\frac{C_6}{4|\mathcal{S}|}\delta_{g,t}+\frac{C_7}{4N}\delta_{L\lambda\lambda,t}+\frac{C_5}{4N}\delta_{\lambda,t},
\end{aligned}
$$

where C_5, C_6, C_7 are properly chosen constants. Given the previously mentioned difference between $\nabla F(\mathbf{w}_t)$ and $\nabla F(\mathbf{w}_t, \lambda^t)$ regarding the use of estimators, it is reasonable that the bound here is related to the error bounds of these estimators.

Similarly, we can establish the following error bound for part ⓑ:

$$
\begin{aligned}
\text{ⓑ} &= \mathbb{E}_t[\|\nabla F(\mathbf{w}_t, \lambda^t) - G(\mathbf{w}_t)\|^2] \\
&\le \mathbb{E}_t\left[2\left\|\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\nabla F_i(\mathbf{w}_t, \lambda^t) - \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\nabla F_i(\mathbf{w}_t, \lambda^t)\right\|^2\right. \\
&\quad \left. +2\left\|\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\nabla F_i(\mathbf{w}_t, \lambda^t) - \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} G_i(\mathbf{w}_t)\right\|^2\right] \\
&\le \frac{12C_\psi^2 B_f^2 + \frac{12C_{L\mathbf{w}\lambda}^2 C_\psi^2 B_f^2}{\gamma^2} + 12B_\psi^2 C_g^2 C_f^2}{|\mathcal{B}|} + \left\|\psi_i(\mathbf{w}_t, \lambda_q^t)[\nabla g_i(\mathbf{w}_t) - \nabla g_i(\mathbf{w}_t; \mathcal{B}_q)]\nabla f_i(\mathbf{u}_i^t)\right\|^2\Big] \\
&\quad + 2\mathbb{E}_t\left[\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\left\|[\nabla_{\mathbf{w}\lambda}^2 L_q(\mathbf{w}_t, \lambda_q^t) - \nabla_{\mathbf{w}\lambda}^2 L_q(\mathbf{w}_t, \lambda_q^t; \mathcal{B}_q)][\mathbf{s}_q^t]^{-1}\nabla_\lambda \psi_i(\mathbf{w}_t, \lambda_q^t) f_i(\mathbf{u}_i^t)\right\|^2\right. \\
&\le \frac{12C_\psi^2 B_f^2 + \frac{12C_{L\mathbf{w}\lambda}^2 C_\psi^2 B_f^2}{\gamma^2} + 12B_\psi^2 C_g^2 C_f^2}{|\mathcal{B}|} + \frac{C_\psi^2 B_f^2 \sigma^2}{\gamma^2} + B_\psi^2 C_f^2 \sigma^2 =: \frac{C_8}{|\mathcal{B}|} + C_9\sigma^2.
\end{aligned}
$$

Intuitively, the difference between $\nabla F(\mathbf{w}_t, \lambda^t)$ and $G(\mathbf{w}_t)$ lies in the fact that the former uses full data for computation while the latter uses mini-batch data. Therefore, the final bound is actually related to the batch size $|\mathcal{B}|$ and the variance σ.

By substituting the bounds for ⓐ and ⓑ into (29), with the natural assumption $\gamma_1 \le 1$, we can derive the following bound:

$$
\begin{aligned}
\mathbb{E}_t[\|\nabla F(\mathbf{w}_t) - \mathbf{m}_{t+1}\|^2] &\le (1-\gamma_1)\|\nabla F(\mathbf{w}_{t-1}) - \mathbf{m}_t\|^2 \\
&+ \frac{4}{\gamma_1}\left[L_F^2\eta_1^2\|\mathbf{m}_{t-1}\|^2 + \gamma_1^2\frac{C_5}{4N}\delta_{\lambda,t} + \gamma_1^2\frac{C_6}{4|\mathcal{S}|}\delta_{g,t} + \gamma_1^2\frac{C_7}{4N}\delta_{L\lambda\lambda,t}\right] + \gamma_1^2\left(\frac{C_8}{|\mathcal{B}|} + C_9\sigma^2\right).
\end{aligned} \tag{30}
$$

Take expectation over all randomness and summation over $t = 1, \ldots, T$, we obtain:

$$
\begin{aligned}
\sum_{t=0}^{T}\mathbb{E}[\|\nabla F(\mathbf{w}_t) - \mathbf{m}_{t+1}\|^2] &\le \frac{1}{\gamma_1}\mathbb{E}[\|\nabla F(\mathbf{w}_0) - \mathbf{m}_1\|^2] + \frac{4L_F^2\eta_1^2}{\gamma_1^2}\sum_{t=1}^{T}\mathbb{E}[\|\mathbf{m}_t\|^2] + \frac{C_5}{N}\sum_{t=1}^{T}\mathbb{E}[\delta_{\lambda,t}] \\
&+ \frac{C_6}{|\mathcal{S}|}\sum_{t=1}^{T}\mathbb{E}[\delta_{g,t}] + \frac{C_7}{N}\sum_{t=1}^{T}\mathbb{E}[\delta_{L\lambda\lambda,t}] + \gamma_1\left(\frac{C_8}{|\mathcal{B}|} + C_9\sigma^2\right)T.
\end{aligned} \tag{31}
$$

In Lemma 8, Lemma 9, and Lemma 10, we have proven the bounds for $\sum_{t=0}^{T}\mathbb{E}[\delta_{\lambda,t}]$, $\sum_{t=0}^{T}\mathbb{E}[\delta_{g,t}]$ and $\sum_{t=0}^{T}\mathbb{E}[\delta_{L\lambda\lambda,t}]$, respectively. For the convenience of the reader, we provide the specific inequalities here.

$$
\sum_{t=0}^{T}\mathbb{E}[\delta_{\lambda,t}] \le \frac{2N}{|\mathcal{B}|\eta_0\mu_L}\delta_{\lambda,0} + \frac{4N\eta_0 T\sigma^2}{\mu_L} + \frac{8N^3C_\lambda^2\eta_1^2}{|\mathcal{B}|^2\eta_0^2\mu_L^2}\sum_{t=0}^{T-1}\mathbb{E}[\|\mathbf{m}_{t+1}\|^2], \tag{32}
$$

$$
\sum_{t=0}^{T}\mathbb{E}[\delta_{g,t}] \le \frac{2|\mathcal{S}|}{|\mathcal{B}|\gamma_0}\delta_{g,0} + 8|\mathcal{S}|\gamma_0\sigma^2 T + \frac{8|\mathcal{S}|^3C_g^2\eta_1^2}{|\mathcal{B}|^2\gamma_0^2}\sum_{t=0}^{T-1}\mathbb{E}[\|\mathbf{m}_{t+1}\|^2], \tag{33}
$$

$$\sum_{t=0}^{T}\mathbb{E}[\delta_{L\lambda\lambda,t}] \leq \frac{4N}{|\mathcal{B}|\gamma_0'}\delta_{L\lambda\lambda,0}+32L_{L\lambda\lambda}^2\sum_{t=0}^{T-1}\mathbb{E}[\delta_{\lambda,t}]+8N\gamma_0' T\sigma^2 + \frac{32N^3L_{L\lambda\lambda}^2(1+C_\lambda^2)\eta_1^2}{|\mathcal{B}|^2\gamma_0'^2}\sum_{t=0}^{T-1}\mathbb{E}[\|\mathbf{m}_{t+1}\|^2]. \tag{34}$$

By plugging the above three inequalities into (31), we obtain

$$\begin{aligned}
&\sum_{t=0}^{T}\mathbb{E}[\|\nabla F(\mathbf{w}_t)-\mathbf{m}_{t+1}\|^2]\\
&\leq \frac{1}{\gamma_1}\mathbb{E}[\|\nabla F(\mathbf{w}_0)-\mathbf{m}_1\|^2]+\frac{2C_{10}}{|\mathcal{B}|\eta_0\mu_L}\delta_{\lambda,0}+\frac{4C_{10}\eta_0 T\sigma^2}{\mu_L}+\frac{2C_6}{|\mathcal{B}|\gamma_0}\delta_{g,0}+8C_6\gamma_0\sigma^2 T\\
&\quad+\frac{4C_7}{|\mathcal{B}|\gamma_0'}\delta_{L\lambda\lambda,0}+8C_7\gamma_0' T\sigma^2+\gamma_1\left(\frac{C_8}{|\mathcal{B}|}+C_9\sigma^2\right)T\\
&\quad+\left[\frac{4L_F^2\eta_1^2}{\gamma_1^2}+\frac{8N^2C_{10}C_\lambda^2\eta_1^2}{|\mathcal{B}|^2\eta_0^2\mu_L^2}+\frac{8|\mathcal{S}|^2C_6C_g^2\eta_1^2}{|\mathcal{B}|^2\gamma_0^2}+\frac{32N^2C_7L_{L\lambda\lambda}^2(1+C_\lambda^2)\eta_1^2}{|\mathcal{B}|^2\gamma_0'^2}\right]\sum_{t=1}^{T}\mathbb{E}[\|\mathbf{m}_t\|^2],
\end{aligned} \tag{35}$$

where $C_{10}=C_5+32C_7L_{L\lambda\lambda}^2$. Recalling Lemma 6, which primarily relies on the smoothness of function F, we have

$$F(\mathbf{w}_{t+1})\leq F(\mathbf{w}_t)+\frac{\eta_1}{2}\|\nabla F(\mathbf{w}_t)-\mathbf{m}_{t+1}\|^2-\frac{\eta_1}{2}\|\nabla F(\mathbf{w}_t)\|^2-\frac{\eta_1}{4}\|\mathbf{m}_{t+1}\|^2.$$

To ultimately prove the conclusion regarding the stationary point, we move the term $||\nabla F(\mathbf{w}_t)||^2$ to the left side and the remaining terms to the right side, then sum both sides over t. Notably, the $F(\mathbf{w}_t)-F(\mathbf{w}_{t+1})$ terms can cancel out each other. Finally, by substituting (35), we can establish the following bound:

$$\begin{aligned}
&\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}[\|\nabla F(\mathbf{w}_t)\|^2]\\
&\leq\frac{1}{T}\left[\frac{\mathbb{E}[\|\nabla F(\mathbf{w}_0)-\mathbf{m}_1\|^2]}{\gamma_1}+\frac{2C_{10}\delta_{\lambda,0}}{|\mathcal{B}|\eta_0\mu_L}+\frac{2C_6\delta_{g,0}}{|\mathcal{B}|\gamma_0}+\frac{4C_7\delta_{L\lambda\lambda,0}}{|\mathcal{B}|\gamma_0'}\right]\\
&\quad+\frac{2[F(\mathbf{w}_0)-F(\mathbf{w}^*)]}{\eta_1 T}+\frac{4C_{10}\eta_0\sigma^2}{\mu_L}+8C_6\gamma_0\sigma^2+8C_7\gamma_0'\sigma^2+\gamma_1\left(\frac{C_8}{|\mathcal{B}|}+C_9\sigma^2\right)\\
&\quad+\frac{1}{T}\left[\frac{4L_F^2\eta_1^2}{\gamma_1^2}+\frac{8N^2C_{10}C_\lambda^2\eta_1^2}{|\mathcal{B}|^2\eta_0^2\mu_L^2}+\frac{8|\mathcal{S}|^2C_6C_g^2\eta_1^2}{|\mathcal{B}|^2\gamma_0^2}+\frac{32N^2C_7L_{L\lambda\lambda}^2(1+C_\lambda^2)\eta_1^2}{|\mathcal{B}|^2\gamma_0'^2}-\frac{1}{2}\right]\sum_{t=1}^{T}\mathbb{E}[\|\mathbf{m}_t\|^2].
\end{aligned} \tag{36}$$

Next, we can eliminate the $\sum_{t=1}^{T}\mathbb{E}[||\mathbf{m}_t||^2]$ term by appropriately setting η_1 by

$$\eta_1^2\leq\min\left\{\frac{\gamma_1^2}{32L_F^2},\frac{|\mathcal{B}|^2\eta_0^2\mu_L^2}{64N^2C_{10}C_\lambda^2},\frac{|\mathcal{B}|^2\gamma_0^2}{64|\mathcal{S}|^2C_6C_g^2},\frac{|\mathcal{B}|^2\gamma_0'^2}{256N^2C_7L_{L\lambda\lambda}^2(1+C_\lambda^2)}\right\},$$

thus we obtain

$$\frac{4L_F^2\eta_1^2}{\gamma_1^2}+\frac{8N^2C_{10}C_\lambda^2\eta_1^2}{|\mathcal{B}|^2\eta_0^2\mu_L^2}+\frac{8|\mathcal{S}|^2C_6C_g^2\eta_1^2}{|\mathcal{B}|^2\gamma_0^2}+\frac{32N^2C_7L_{L\lambda\lambda}^2(1+C_\lambda^2)\eta_1^2}{|\mathcal{B}|^2\gamma_0'^2}-\frac{1}{2}\leq 0,$$

which implies that the last term of the RHS of inequality (36) are less or equal to zero. Hence

$$\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}[\|\nabla F(\mathbf{w}_t)\|^2] \leq \frac{4C_{10}\eta_0\sigma^2}{\mu_L} + 8C_6\gamma_0\sigma^2 + 8C_7\gamma_0'\sigma^2 + \gamma_1\left(\frac{C_8}{|\mathcal{B}|} + C_9\sigma^2\right) + \frac{2[F(\mathbf{w}_0)-F(\mathbf{w}^*)]}{\eta_1 T} + \frac{1}{T}\left[\frac{\mathbb{E}[\|\nabla F(\mathbf{w}_0)-\mathbf{m}_1\|^2]}{\gamma_1} + \frac{2C_{10}\delta_{\lambda,0}}{|\mathcal{B}|\eta_0\mu_L} + \frac{2C_6\delta_{g,0}}{|\mathcal{B}|\gamma_0} + \frac{4C_7\delta_{L\lambda\lambda,0}}{|\mathcal{B}|\gamma_0'}\right]. \tag{37}$$

To satisfy the criterion for a stationary point in the definition of algorithm convergence, we need to ensure that the right-hand side of the above equation is less than ϵ^2. To achieve this, we can set the values of η_0, γ_0, γ_0', γ_1 and T as follows

$$\eta_0 \leq \frac{\mu_L\epsilon^2}{48C_{10}\sigma^2}, \gamma_0 \leq \frac{\epsilon^2}{96C_6\sigma^2}, \gamma_0' \leq \frac{\epsilon^2}{96C_7\sigma^2}, \gamma_1 \leq \frac{\epsilon^2}{12(\frac{C_8}{|\mathcal{B}|}+C_9\sigma^2)},$$

$$T \geq \left\{\frac{30[F(\mathbf{w}_0)-F(\mathbf{w}^*)]}{\eta_1\epsilon^2}, \frac{15\mathbb{E}[\|\nabla F(\mathbf{w}_0)-\mathbf{m}_1\|^2]}{\gamma_1\epsilon^2}, \frac{30C_{10}\delta_{\lambda,0}}{|\mathcal{B}|\eta_0\mu_L\epsilon^2}, \frac{30C_6\delta_{g,0}}{|\mathcal{B}|\gamma_0\epsilon^2}, \frac{60C_7\delta_{L\lambda\lambda,0}}{|\mathcal{B}|\gamma_0'\epsilon^2}\right\},$$

and finally we have

$$\frac{1}{T+1}\sum_{t=0}^{T}\mathbb{E}[\|\nabla F(\mathbf{w}_t)\|^2] \leq \frac{1}{3}\epsilon^2 + \frac{1}{3}\epsilon^2 < \epsilon^2.$$

From the values of η_0, γ_0, γ_0', γ_1 and T, we can conclude that when $T = \mathcal{O}(1/\epsilon^4)$, the algorithm can find the stationary point. This completes the proof. □

Proof of Lemma 1 Given $\ell(\mathbf{w};\mathbf{x}',\mathbf{x},q) \geq \mathbb{I}(h_q(\mathbf{x}';\mathbf{w})-h_q(\mathbf{x};\mathbf{w}) \geq 0)$, we have $\bar{g}(\mathbf{w};\mathbf{x}_i^q,\mathcal{S}_q) \geq r(\mathbf{w};\mathbf{x}_i^q,\mathcal{S}_q)$ for each $(q,\mathbf{x}_i^q)$, which immediately follows the desired conclusion. □

Proof of Lemma 2 To show the equivalence in the Lemma, it suffices to show that $\lambda_q(\mathbf{w})$ is the $(K+1)$-th largest value in the set $\{h_q(\mathbf{x}';\mathbf{w})|\mathbf{x}' \in \mathcal{S}_q\}$. Let $\{\theta_1,\theta_2,\cdots,\theta_{N_q}\}$ denote a sequence of values defined by sorting $\{h_q(\mathbf{x}';\mathbf{w})|\mathbf{x}' \in \mathcal{S}_q\}$ in descending order, i.e., $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_{N_q}$. θ_k denote the k-th largest value.

Recall the definition of $\lambda_q(\mathbf{w})$

$$\lambda_q(\mathbf{w}) = \arg\min_{\lambda}(K+\varepsilon)\lambda + \sum_{\mathbf{x}'\in\mathcal{S}_q}(h_q(\mathbf{x}';\mathbf{w})-\lambda)_+,$$

where $\varepsilon \in (0,1)$. Define function $\Lambda_q(\lambda) := (K+\varepsilon)\lambda + \sum_{i=1}^{N_q}(\theta_i-\lambda)_+$, then it follows that $\lambda_q(\mathbf{w}) = \arg\min_\lambda \Lambda_q(\lambda)$. Take the derivative of $\Lambda_q(\lambda)$, we have

$$\nabla_\lambda\Lambda_q(\lambda) = K+\varepsilon-\sum_{i=1}^{N_q}d(\theta_i-\lambda), \text{ where } d(\theta_i-\lambda) = \begin{cases}1, & \theta_i > \lambda\\ \varepsilon' \in [0,1], & \theta_i = \lambda\\ 0, & \theta_i < \lambda\end{cases}.$$

First, we assume $\theta_K > \theta_{K+1}$. One may consider this problem in three cases.

- If $\lambda > \theta_{K+1}$, then $\sum_{i=1}^{N_q} d(\theta_i-\lambda) \leq K$, so we have $\nabla_\lambda\Lambda_q(\lambda) \geq K+\varepsilon-K = \varepsilon > 0$.
- If $\lambda < \theta_{K+1}$, then $\sum_{i=1}^{N_q} d(\theta_i-\lambda) \geq K+1$, so we have $\nabla_\lambda\Lambda_q(\lambda) \leq K+\varepsilon-K-1 = \varepsilon-1 < 0$.

– If $\lambda = \theta_{K+1}$, then $\sum_{i=1}^{N_q} d(\theta_i - \lambda) = K + \varepsilon'$, so we have $\nabla_\lambda \Lambda_q(\lambda) = K + \varepsilon - K - \varepsilon' = \varepsilon - \varepsilon'$. Thus we will have $\nabla_\lambda \Lambda_q(\lambda) = 0$ by setting $\varepsilon' = \varepsilon$. Hence $\lambda_q(\mathbf{w}) = \theta_{K+1}$.

Second, if $\theta_K = \theta_{K+1}$. One may consider this problem in three cases.

– If $\lambda > \theta_{K+1}$, then $\sum_{i=1}^{N_q} d(\theta_i - \lambda) \leq K - 1$, so we have $\nabla_\lambda \Lambda_q(\lambda) \geq K + \varepsilon - K + 1 = 1 + \epsilon > 0$.
– If $\lambda < \theta_{K+1}$, then $\sum_{i=1}^{N_q} d(\theta_i - \lambda) \geq K + 1$, so we have $\nabla_\lambda \Lambda_q(\lambda) \leq K + \varepsilon - K - 1 < 0$.
– If $\lambda = \theta_K = \theta_{K+1}$, then $\sum_{i=1}^{N_q} d(\theta_i - \lambda) = K - 1 + 2\epsilon'$, so we have $\nabla_\lambda \Lambda_q(\lambda) = K + \varepsilon - K + 1 - 2\varepsilon' = 1 + \varepsilon - 2\varepsilon'$. Thus we will have $\nabla_\lambda \Lambda_q(\lambda) = 0$ by setting $\varepsilon' = (1 + \varepsilon)/2$. Hence $\lambda_q(\mathbf{w}) = \theta_{K+1}$.

In summary, $\theta_{K+1} = \lambda_q(\mathbf{w}) = \arg\min_\lambda \Lambda_q(\lambda)$. The proof is finished. □

Proof of Lemma 3 Given the condition $\psi(h_q(\mathbf{x}_i^q; \mathbf{w}) - \lambda_q(\mathbf{w})) \leq C\mathbb{I}(h_q(\mathbf{x}_i^q; \mathbf{w}) - \lambda_q(\mathbf{w}) > 0)$ and $\ell(\mathbf{w}; \mathbf{x}', \mathbf{x}, q) \geq \mathbb{I}(h_q(\mathbf{x}'; \mathbf{w}) - h_q(\mathbf{x}; \mathbf{w}) > 0)$, we have

$$\frac{\psi(h_q(\mathbf{x}_i^q; \mathbf{w}) - \lambda_q(\mathbf{w}))(2^{y_i^q} - 1)}{C Z_q^K \log_2(\bar{g}(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q) + 1)} \leq \frac{\mathbb{I}(\mathbf{x}_i^q \in \mathcal{S}_q[K])(2^{y_i^q} - 1)}{Z_q^K \log_2(r(\mathbf{w}; \mathbf{x}_i^q, \mathcal{S}_q) + 1)}$$

for each $(q, \mathbf{x}_i^q)$. The desired result follows. □

Proof of Lemma 4 Recall

$$L_q(\lambda; \mathbf{w}) = \frac{K}{N_q}\lambda + \frac{\tau_2}{2}\lambda^2 + \frac{1}{N_q}\sum_{\mathbf{x}_i \in \mathcal{S}_q} \tau_1 \ln(1 + \exp((h_q(\mathbf{x}_i; \mathbf{w}) - \lambda)/\tau_1)).$$

We first define the following two auxiliary functions:

$$\tilde{L}_q(\lambda; \mathbf{w}) = \frac{K}{N_q}\lambda + \frac{1}{N_q}\sum_{\mathbf{x}_i \in \mathcal{S}_q} (h_q(\mathbf{x}_i; \mathbf{w}) - \lambda)_+, \quad \hat{L}_q(\lambda; \mathbf{w}) = \tilde{L}_q(\lambda; \mathbf{w}) + \frac{\tau_2}{2}\lambda^2.$$

For simplicity, denote $\lambda_* = \arg\min_\lambda L_q(\lambda; \mathbf{w})$, $\tilde{\lambda}_* = \arg\min_\lambda \tilde{L}_q(\lambda; \mathbf{w})$, $\hat{\lambda}_* = \arg\min_\lambda \hat{L}_q(\lambda; \mathbf{w})$. Note that it is obvious to see that when $\lambda \geq 2c_h$, function $\tilde{L}_q(\lambda; \mathbf{w})$ is monotonically increasing, and monotonically decreasing when $\lambda \leq 0$. Thus the optimal point is bounded, i.e. $\tilde{\lambda}_* \in [0, 2c_h]$. Similarly, we have $\nabla_\lambda L_q(\lambda; \mathbf{w}) < 0$ when $\lambda \leq 0$ and $\nabla_\lambda L_q(\lambda; \mathbf{w}) \geq 0$ when $\lambda \geq c_h + \tau_1 \ln N_{max}$ where $N_{max} = \max_q N_q$. This allows us to show that the optimal point λ_* is also bounded, i.e. $\lambda_* \in [0, c_h + \tau_1 \ln N_{max}]$. By applying Lemma 8 in (Yang & Lin, 2018) to $\tilde{L}_q(\lambda; \mathbf{w})$, we know that there exists a constant $c_1 > 0$ such that for all λ we have

$$|\lambda - \lambda_q(\mathbf{w})| \leq c_1(\tilde{L}_q(\lambda; \mathbf{w}) - \tilde{L}_q(\lambda_q(\mathbf{w}); \mathbf{w})). \tag{38}$$

It is trivial to show $\tau_1 \ln(1 + \exp(x/\tau_1)) \geq x_+ \; \forall x \in \mathbb{R}$ and $\tau_1 \ln(1 + \exp(x/\tau_1)) - x_+ \leq (\ln 2)\tau_1$. Then it follows easily that

$$\hat{L}_q(\lambda; \mathbf{w}) \leq L_q(\lambda; \mathbf{w}) \leq \hat{L}_q(\lambda; \mathbf{w}) + c_2\tau_1, \tag{39}$$

where $c_2 = \ln 2$. Then with inequality (39) and the optimality of λ_*, we have

$$\begin{aligned}\tilde{L}_q(\lambda_*; \mathbf{w}) &= \hat{L}_q(\lambda_*; \mathbf{w}) - \frac{\tau_2}{2}\lambda_*^2 \leq L_q(\lambda_*; \mathbf{w}) - \frac{\tau_2}{2}\lambda_*^2 \leq L_q(\tilde{\lambda}_*; \mathbf{w}) - \frac{\tau_2}{2}\lambda_*^2 \\ &\leq \hat{L}_q(\tilde{\lambda}_*; \mathbf{w}) + c_2\tau_1 - \frac{\tau_2}{2}\lambda_*^2 = \tilde{L}_q(\tilde{\lambda}_*; \mathbf{w}) + \frac{\tau_2}{2}\tilde{\lambda}_*^2 + c_2\tau_1 - \frac{\tau_2}{2}\lambda_*^2,\end{aligned}$$

which follows that

$$|\tilde{L}_q(\lambda_*; \mathbf{w}) - \tilde{L}_q(\tilde{\lambda}_*; \mathbf{w})| \leq \frac{\tau_2}{2}\tilde{\lambda}_*^2 + c_2\tau_1 - \frac{\tau_2}{2}\lambda_*^2. \tag{40}$$

Combining inequalities (38), (40) and the boundedness of λ_*, $\tilde{\lambda}_*$, and setting $\tau_1 = \tau_2 = \varepsilon$, we obtain

$$|\lambda_q(\mathbf{w}) - \hat{\lambda}_q(\mathbf{w})| \leq c_1\left(\frac{\tau_2}{2}\tilde{\lambda}_*^2 + c_2\tau_1 - \frac{\tau_2}{2}\lambda_*^2\right) = \mathcal{O}(\varepsilon).$$

To demonstrate the smoothness of $L_q(\lambda; \mathbf{w})$, we first show that

$$\tau_1 \ln(1 + \exp(x/\tau_1)) = \max_{\alpha\in[0,1]} x\alpha - \tau_1[\alpha\ln(\alpha) + (1-\alpha)\ln(1-\alpha)] =: A(\alpha). \tag{41}$$

Note that the solution to $A'(\alpha) = x - \tau_1[\ln(\alpha) - \ln(1-\alpha)] = 0$ is $\alpha^* = 1 - (1+\exp(x/\tau_1))^{-1}$. Then $A(\alpha^*) = \tau_1\ln(1+\exp(x/\tau_1))$, which implies (41). Besides, $A(\alpha)$ is strong concave because

$$(A(\alpha) + \tau_1\alpha^2)'' = -\tau_1\left(\frac{1}{\alpha} + \frac{1}{1-\alpha}\right) + 2\tau_1 < 0.$$

It follows that

$$L_q(\lambda; \mathbf{w}) = \frac{K}{N_q}\lambda + \frac{\tau_2}{2}\lambda^2 + \frac{1}{N_q}\sum_{\mathbf{x}_i\in\mathcal{S}_q}\max_{\alpha\in(0,1)}(h_q(\mathbf{x}_i; \mathbf{w}) - \lambda)\alpha - \tau_1[\alpha\ln(\alpha) + (1-\alpha)\ln(1-\alpha)].$$

Then by Theorem 1 in Nesterov (2005), $L_q(\lambda; \mathbf{w})$ is smooth. The strong convexity of $L_q(\lambda; \mathbf{w})$ follows from the convexity of $L_q(\lambda; \mathbf{w}) - \frac{\tau_2}{2}\lambda^2$, which can be proved by checking the non-negativity of its second derivative

$$\nabla^2\left(L_q(\lambda; \mathbf{w}) - \frac{\tau_2}{2}\lambda^2\right) = \frac{1}{N_q}\sum_{x_i\in\mathcal{S}_q}\frac{\frac{1}{\tau_1}\exp((\lambda - h_q(\mathbf{x}_i; \mathbf{w}))/\tau_1)}{[1 + \exp((\lambda - h_q(\mathbf{x}_i; \mathbf{w}))/\tau_1)]^2} \geq 0.$$

□

Proof of Lemma 5 Here, we aim to prove that the function $F(\mathbf{w})$ is smooth, which essentially means showing that $\|\nabla F(\mathbf{w}_1) - \nabla F(\mathbf{w}_2)\|$ can be bounded by $c\|\mathbf{w}_1 - \mathbf{w}_2\|$, where $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^d$ and c is some constant. Thus, we start with $\|\nabla F(\mathbf{w}_1) - \nabla F(\mathbf{w}_2)\|$ and first expand it according to the definition of $F(\mathbf{w})$ and triangle inequality $\|\mathbf{x}_1 + \cdots + \mathbf{x}_n\| \leq \|\mathbf{x}_1\| + \cdots + \|\mathbf{x}_n\|$:

$$\begin{aligned}
\|\nabla F(\mathbf{w}_1) - \nabla F(\mathbf{w}_2)\| \leq{}& \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\|\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_1, \lambda_q(\mathbf{w}_1))f_i(g_i(\mathbf{w}_1)) - \nabla_{\mathbf{w}}\psi_i(\mathbf{w}_2, \lambda_q(\mathbf{w}_2))f_i(g_i(\mathbf{w}_2))\| \\
&+ \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\|\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_2, \lambda_q(\mathbf{w}_2))[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_2, \lambda_q(\mathbf{w}_2))]^{-1}\nabla_\lambda\psi_i(\mathbf{w}_2, \lambda_q(\mathbf{w}_2))f_i(g_i(\mathbf{w}_2)) \\
&- \nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_1, \lambda_q(\mathbf{w}_1))[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_1, \lambda_q(\mathbf{w}_1))]^{-1}\nabla_\lambda\psi_i(\mathbf{w}_1, \lambda_q(\mathbf{w}_1))f_i(g_i(\mathbf{w}_1))\| \\
&+ \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\|\psi_i(\mathbf{w}_1, \lambda_q(\mathbf{w}_1))\nabla g_i(\mathbf{w}_1)\nabla f_i(g_i(\mathbf{w}_1)) - \psi_i(\mathbf{w}_2, \lambda_q(\mathbf{w}_2))\nabla g_i(\mathbf{w}_2)\nabla f_i(g_i(\mathbf{w}_2))\|.
\end{aligned}$$

Next, we establish the corresponding bounds for the three terms on the RHS of the above equation. The idea is to continuously use inequality $\|\mathbf{x}_1 + \cdots + \mathbf{x}_n\|^2 \leq n\|\mathbf{x}_1\|^2 + \cdots + n\|\mathbf{x}_n\|^2$ and the Lipschitz continuity or smoothness properties of the functions assumed in Assumption 1 to decompose and bound each term. Specifically, for the first term, we have

$$
\begin{aligned}
&\|\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))f_i(g_i(\mathbf{w}_1))-\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_2,\lambda_q(\mathbf{w}_2))f_i(g_i(\mathbf{w}_2))\|^2\\
&\quad\le 2\|\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))[f_i(g_i(\mathbf{w}_1)-f_i(g_i(\mathbf{w}_2))]\|^2\\
&\qquad+2\|[\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))-\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_2,\lambda_q(\mathbf{w}_2))]f_i(g_i(\mathbf{w}_2))\|^2\\
&\quad\le 2C_\psi^2C_f^2\|g_i(\mathbf{w}_1)-g_i(\mathbf{w}_2)\|^2+2L_\psi^2[\|\mathbf{w}_1-\mathbf{w}_2\|^2+2\|\lambda_q(\mathbf{w}_1)-\lambda_q(\mathbf{w}_2)\|^2]B_f^2\\
&\quad\le (2C_\psi^2C_f^2C_g^2+2B_f^2L_\psi^2(1+C_\lambda))\|\mathbf{w}_1-\mathbf{w}_2\|^2=:C_1^2\|\mathbf{w}_1-\mathbf{w}_2\|^2.
\end{aligned}
$$

We can analyze the second term using a similar approach and obtain the following result:

$$
\begin{aligned}
&\|\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_2,\lambda_q(\mathbf{w}_2))[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_2,\lambda_q(\mathbf{w}_2))]^{-1}\nabla_\lambda\psi_i(\mathbf{w}_2,\lambda_q(\mathbf{w}_2))f_i(g_i(\mathbf{w}_2))\\
&\quad-\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))]^{-1}\nabla_\lambda\psi_i(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))f_i(g_i(\mathbf{w}_1))\|^2\\
&\le 4\|[\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_2,\lambda_q(\mathbf{w}_2))-\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))]\\
&\quad[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_2,\lambda_q(\mathbf{w}_2))]^{-1}\nabla_\lambda\psi_i(\mathbf{w}_2,\lambda_q(\mathbf{w}_2))f_i(g_i(\mathbf{w}_2))\|^2\\
&\quad+4\|\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))\Big[[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_2,\lambda_q(\mathbf{w}_2))]^{-1}\\
&\quad-[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))]^{-1}\Big]\nabla_\lambda\psi_i(\mathbf{w}_2,\lambda_q(\mathbf{w}_2))f_i(g_i(\mathbf{w}_2))\|^2\\
&\quad+4\|\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))]^{-1}[\nabla_\lambda\psi_i(\mathbf{w}_2,\lambda_q(\mathbf{w}_2))\\
&\quad-\nabla_\lambda\psi_i(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))]f_i(g_i(\mathbf{w}_2))\|^2\\
&\quad+4\|\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))]^{-1}\nabla_\lambda\psi_i(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))[f_i(g_i(\mathbf{w}_2))-f_i(g_i(\mathbf{w}_1))]\|^2\\
&\le\left[\left(\frac{4L^2_{L\mathbf{w}\lambda}C^2_\psi B^2_f}{\mu^2_L}+\frac{4C^2_{L\mathbf{w}\lambda}L^2_{L\lambda}C^2_\psi B^2_f}{\mu^4_L}+\frac{4C^2_{L\mathbf{w}\lambda}L^2_\psi B^2_f}{\mu^2_L}\right)(1+C^2_\lambda)+\frac{4C^2_{L\mathbf{w}\lambda}C^2_\psi C^2_f C^2_g}{\mu^2_L}\right]\|\mathbf{w}_1-\mathbf{w}_2\|^2\\
&=:C_2^2\|\mathbf{w}_1-\mathbf{w}_2\|^2,
\end{aligned}
$$

Similarly, for the third term, we can derive the following bound:

$$
\begin{aligned}
&\|\psi_i(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))\nabla g_i(\mathbf{w}_1)\nabla f_i(g_i(\mathbf{w}_1))-\psi_i(\mathbf{w}_2,\lambda_q(\mathbf{w}_2))\nabla g_i(\mathbf{w}_2)\nabla f_i(g_i(\mathbf{w}_2))\|^2\\
&\quad\le 3\|[\psi_i(\mathbf{w}_1,\lambda_q(\mathbf{w}_1))-\psi_i(\mathbf{w}_2,\lambda_q(\mathbf{w}_2))]\nabla g_i(\mathbf{w}_1)\nabla f_i(g_i(\mathbf{w}_1))\|^2\\
&\qquad+3\|\psi_i(\mathbf{w}_2,\lambda_q(\mathbf{w}_2))[\nabla g_i(\mathbf{w}_1)-\nabla g_i(\mathbf{w}_2)]\nabla f_i(g_i(\mathbf{w}_1))\|^2\\
&\qquad+3\|\psi_i(\mathbf{w}_2,\lambda_q(\mathbf{w}_2))\nabla g_i(\mathbf{w}_2)[\nabla f_i(g_i(\mathbf{w}_1))-\nabla f_i(g_i(\mathbf{w}_2))]\|^2\\
&\quad\le\left[3C^2_\psi C^2_g C^2_f(1+C^2_\lambda)+3B^2_\psi L^2_g C^2_f+3B^2_\ell C^2_g L^2_f C^2_g\right]\|\mathbf{w}_1-\mathbf{w}_2\|^2=:C_3^2\|\mathbf{w}_1-\mathbf{w}_2\|^2.
\end{aligned}
$$

By collecting the results obtained above, we can finally derive the following desired result

$$
\|\nabla F(\mathbf{w}_1)-\nabla F(\mathbf{w}_2)\|\le\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}(C_1+C_2+C_3)\|\mathbf{w}_1-\mathbf{w}_2\|=L_F\|\mathbf{w}_1-\mathbf{w}_2\|,
$$

where $L_F:=C_1+C_2+C_3$. In this way, we prove the smoothness property of the objective function $F(\mathbf{w})$. □

Proof of Lemma 6 We employ the fact that the function $F(\mathbf{w})$ is L_F-smooth to establish the relationship between the stochastic gradient estimation error $\|\nabla F(\mathbf{w}_t)-\mathbf{m}_{t+1}\|^2$ and other terms:

$$
\begin{aligned}
F(\mathbf{w}_{t+1})&\le F(\mathbf{w}_t)+\nabla F(\mathbf{w}_t)^T(\mathbf{w}_{t+1}-\mathbf{w}_t)+\frac{L_F}{2}\|\mathbf{w}_{t+1}-\mathbf{w}_t\|^2\\
&=F(\mathbf{w}_t)-\eta_1\nabla F(\mathbf{w}_t)^T\mathbf{m}_{t+1}+\frac{L_F}{2}\eta_1^2\|\mathbf{m}_{t+1}\|^2
\end{aligned}
$$

$$= F(\mathbf{w}_t) + \frac{\eta_1}{2}\|\nabla F(\mathbf{w}_t) - \mathbf{m}_{t+1}\|^2 - \frac{\eta_1}{2}\|\nabla F(\mathbf{w}_t)\|^2 + \left(\frac{L_F}{2}\eta_1^2 - \frac{\eta_1}{2}\right)\|\mathbf{m}_{t+1}\|^2$$
$$\leq F(\mathbf{w}_t) + \frac{\eta_1}{2}\|\nabla F(\mathbf{w}_t) - \mathbf{m}_{t+1}\|^2 - \frac{\eta_1}{2}\|\nabla F(\mathbf{w}_t)\|^2 - \frac{\eta_1}{4}\|\mathbf{m}_{t+1}\|^2,$$

□

where the first inequality directly uses the smoothness property of $F(\mathbf{w})$, the second inequality utilizes $\eta_1 \leq \frac{1}{2L_F}$, the first equation employs the update rule of $\mathbf{m}_{t+1}$, and the second equation uses the fact that $-2a^\top b = \|a-b\|^2 - \|a\|^2 - \|b\|^2$.

Proof of Lemma 8 Here, we establish the error bound for the solutions to the lower-level problems. First, recall and define the following notations

$$\lambda_q^{t+1} = \begin{cases} \lambda_q^t - \eta_0 \nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t; \mathcal{B}_q) & \text{if } q \in \mathcal{B} \\ \lambda_q^t & \text{o.w.} \end{cases}, \quad \widetilde{\lambda}_q^t := \lambda_q^t - \eta_0 \nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t; \mathcal{B}_q).$$

In this lemma, our strategy is to first obtain the estimation error bound for the solutions updated in the t-th iteration, followed by establishing the estimation error bound for all solutions. The error bound between $\widetilde{\lambda}_q^t$ (the portion updated by the current mini-batch) and $\lambda_q(\mathbf{w}_t)$ can be derived as follows:

$$\begin{aligned}
&\mathbb{E}_t[\|\widetilde{\lambda}_q^t - \lambda_q(\mathbf{w}_t)\|^2] = \mathbb{E}_t[\|\lambda_q^t - \eta_0\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t; \mathcal{B}_q) - \lambda_q(\mathbf{w}_t)\|^2] \\
&= \mathbb{E}_t[\|\lambda_q^t - \eta_0\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t; \mathcal{B}_q) - \lambda_q(\mathbf{w}_t) + \eta_0\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t)) + \eta_0\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t) - \eta_0\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t)\|^2] \\
&= \|\lambda_q^t - \lambda_q(\mathbf{w}_t) + \eta_0\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t)) - \eta_0\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t)\|^2 \\
&\quad + \mathbb{E}_t[\|\eta_0\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t) - \eta_0\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t; \mathcal{B}_q)\|^2] \\
&\leq \|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2 + \eta_0^2\|\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t)) - \nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t)\|^2 \\
&\quad + 2\eta_0\langle\lambda_q^t - \lambda_q(\mathbf{w}_t), \nabla_\lambda L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t)) - \nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t)\rangle + \eta_0^2\sigma^2 \\
&\overset{(a)}{\leq} \|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2 + \eta_0^2 L_L^2\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2 - 2\eta_0\mu_L\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2 + \eta_0^2\sigma^2 \\
&\overset{(b)}{\leq} (1 - \eta_0\mu_L)\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2 + \eta_0^2\sigma^2,
\end{aligned} \tag{42}$$

where $\mathbb{E}_t[\cdot]$ takes expectation over the randomness at iteration t, the first equality holds because $\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t)) = 0$, (a) uses the strong monotonicity of $L_q(\mathbf{w}_t, \cdot)$ as it is assumed to be μ_L-strongly convex, and (b) uses the assumption $\eta_0 \leq \mu_L / L_L^2$.

Moreover, consider the randomness on the query sampling $\mathcal{B}$, we have

$$\mathbb{E}_t[\|\lambda_q^{t+1} - \lambda_q(\mathbf{w}_t)\|^2] = \frac{|\mathcal{B}|}{N}\mathbb{E}_t[\|\widetilde{\lambda}_q^t - \lambda_q(\mathbf{w}_t)\|^2] + \frac{N - |\mathcal{B}|}{N}\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2,$$

which follows

$$\mathbb{E}_t[\|\widetilde{\lambda}_q^t - \lambda_q(\mathbf{w}_t)\|^2] = \frac{N}{|\mathcal{B}|}\mathbb{E}_t[\|\lambda_q^{t+1} - \lambda_q(\mathbf{w}_t)\|^2] - \frac{N - |\mathcal{B}|}{|\mathcal{B}|}\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2. \tag{43}$$

Combining inequalities (42) and (43), we obtain

$$\begin{aligned}
\mathbb{E}_t[\|\lambda_q^{t+1} - \lambda_q(\mathbf{w}_t)\|^2] &\leq \frac{N - |\mathcal{B}|}{N}\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2 + \frac{|\mathcal{B}|}{N}(1 - \eta_0\mu_L)\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2 + \frac{|\mathcal{B}|}{N}\eta_0^2\sigma^2 \\
&\leq \left(1 - \frac{|\mathcal{B}|\eta_0\mu_L}{N}\right)\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2 + \frac{|\mathcal{B}|\eta_0^2\sigma^2}{N}.
\end{aligned} \tag{44}$$

To further derive the recursive relationship for the error bound concerning $\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2$, we proceed as follows

$$\begin{aligned}
&\mathbb{E}_t[\|\lambda_q^{t+1} - \lambda_q(\mathbf{w}_{t+1})\|^2] \\
&\leq \left(1 + \frac{|\mathcal{B}|\eta_0\mu_L}{2N}\right)\mathbb{E}_t[\|\lambda_q^{t+1} - \lambda_q(\mathbf{w}_t)\|^2] + \left(1 + \frac{2N}{|\mathcal{B}|\eta_0\mu_L}\right)\mathbb{E}_t[\|\lambda_q(\mathbf{w}_{t+1}) - \lambda_q(\mathbf{w}_t)\|^2] \\
&\leq \left(1 - \frac{|\mathcal{B}|\eta_0\mu_L}{2N}\right)\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2 + \frac{2|\mathcal{B}|\eta_0^2\sigma^2}{N} + \frac{4NC_\lambda^2}{|\mathcal{B}|\eta_0\mu_L}\mathbb{E}_t[\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2], \qquad (45)
\end{aligned}$$

where we employ $\|a+b\|^2 \leq (1+\frac{1}{\gamma})\|a\|^2 + (1+\gamma)\|b\|^2, \gamma > 0$ in the first inequality, and use (44) and the assumption $\eta_0 \leq \frac{2N}{|\mathcal{B}|\mu_L}$, i.e., $\frac{|\mathcal{B}|\eta_0\mu_L}{2N} \leq 1$ in the second inequality.

Taking summation over all queries and expectation over all randomness, we have

$$\mathbb{E}[\|\lambda^{t+1} - \lambda(\mathbf{w}_{t+1})\|^2] \leq (1 - \frac{|\mathcal{B}|\eta_0\mu_L}{2N})\mathbb{E}[\|\lambda^t - \lambda(\mathbf{w}_t)\|^2] + 2|\mathcal{B}|\eta_0^2\sigma^2 + \frac{4N^2C_\lambda^2}{|\mathcal{B}|\eta_0\mu_L}\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2]. \qquad (46)$$

Finally, we take summation over $t = 0, \ldots, T-1$, and derive the following error bound

$$\sum_{t=0}^{T}\mathbb{E}[\|\lambda^t - \lambda(\mathbf{w}_t)\|^2] \leq \frac{2N}{|\mathcal{B}|\eta_0\mu_L}\|\lambda^0 - \lambda(\mathbf{w}_0)\|^2 + \frac{4N\eta_0 T\sigma^2}{\mu_L} + \frac{8N^3C_\lambda^2}{|\mathcal{B}|^2\eta_0^2\mu_L^2}\sum_{t=0}^{T-1}\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2]. \qquad (47)$$

□

Proof of Lemma 9 We aim to establish the tracking error bound for the moving average estimator $\mathbf{u}^t$ of the function $g(\mathbf{w}_t)$. To achieve this, we first establish the recursive relationship for this tracking error, starting with the analysis of $\|\mathbf{u}^{t+1} - g(\mathbf{w}_t)\|^2$:

$$\begin{aligned}
&\|\mathbf{u}^{t+1} - g(\mathbf{w}_t)\|^2 \\
&= \|\mathbf{u}^{t+1} - \mathbf{u}^t + \mathbf{u}^t - g(\mathbf{w}_t)\|^2 = \|\mathbf{u}^{t+1} - \mathbf{u}^t\|^2 + \|\mathbf{u}^t - g(\mathbf{w}_t)\|^2 + 2\langle \mathbf{u}^{t+1} - \mathbf{u}^t, \mathbf{u}^t - g(\mathbf{w}_t)\rangle \\
&= \|\mathbf{u}^{t+1} - \mathbf{u}^t\|^2 + \|\mathbf{u}^t - g(\mathbf{w}_t)\|^2 + 2\sum_{i\in\mathcal{B}}\langle \mathbf{u}_i^{t+1} - \mathbf{u}_i^t, \mathbf{u}_i^t - g_i(\mathbf{w}_t)\rangle \\
&= \|\mathbf{u}^{t+1} - \mathbf{u}^t\|^2 + \|\mathbf{u}^t - g(\mathbf{w}_t)\|^2 \\
&\quad + \underbrace{2\sum_{i\in\mathcal{B}}\langle \mathbf{u}_i^{t+1} - \mathbf{u}_i^t, \mathbf{u}_i^t - g_i(\mathbf{w}_t; \mathcal{B}_q)\rangle}_{\clubsuit} + \underbrace{2\sum_{i\in\mathcal{B}}\langle \mathbf{u}_i^{t+1} - \mathbf{u}_i^t, g_i(\mathbf{w}_t; \mathcal{B}_q) - g_i(\mathbf{w}_t)\rangle}_{\spadesuit}. \qquad (48)
\end{aligned}$$

In the above derivation, we perform a fine-grained decomposition of the original terms to facilitate establishing bounds for each of them individually. Next, we analyze ♣ and ♠ separately. For ♣, based on the update rule $\mathbf{u}_i^t - \mathbf{u}_i^{t+1} = \gamma_0(\mathbf{u}_i^t - g_i(\mathbf{w}_t; B_q))\ \forall i \in \mathcal{B}$ and the inequality $2\langle b-a, a-c\rangle = \|b-c\|^2 - \|a-b\|^2 - \|a-c\|^2$, we have

$$\begin{aligned}
\clubsuit &= 2\sum_{i\in\mathcal{B}}\langle \mathbf{u}_i^{t+1} - g_i(\mathbf{w}_t), \mathbf{u}_i^t - g_i(\mathbf{w}_t, \mathcal{B}_q)\rangle + 2\sum_{i\in\mathcal{B}}\langle g_i(\mathbf{w}_t) - \mathbf{u}_i^t, \mathbf{u}_i^t - g_i(\mathbf{w}_t, \mathcal{B}_q)\rangle \\
&= \frac{2}{\gamma_0}\sum_{i\in\mathcal{B}}\langle \mathbf{u}_i^{t+1} - g_i(\mathbf{w}_t), \mathbf{u}_i^t - \mathbf{u}_i^{t+1}\rangle + 2\sum_{i\in\mathcal{B}}\langle g_i(\mathbf{w}_t) - \mathbf{u}_i^t, \mathbf{u}_i^t - g_i(\mathbf{w}_t, \mathcal{B}_q)\rangle \\
&= \frac{1}{\gamma_0}\sum_{i\in\mathcal{B}}[\|\mathbf{u}_i^t - g_i(\mathbf{w}_t)\|^2 - \|\mathbf{u}_i^{t+1} - g_i(\mathbf{w}_t)\|^2 - \|\mathbf{u}_i^{t+1} - \mathbf{u}_i^t\|^2]
\end{aligned}$$

$$+2\sum_{i\in\mathcal{B}}\langle g_i(\mathbf{w}_t)-\mathbf{u}_i^t,\mathbf{u}_i^t-g_i(\mathbf{w}_t,\mathcal{B}_q)\rangle$$
$$=\frac{1}{\gamma_0}\|\mathbf{u}^t-g(\mathbf{w}_t)\|^2-\frac{1}{\gamma_0}\|\mathbf{u}^{t+1}-g(\mathbf{w}_t)\|^2-\frac{1}{\gamma_0}\|\mathbf{u}^{t+1}-\mathbf{u}^t\|^2+2\sum_{i\in\mathcal{B}}\langle g_i(\mathbf{w}_t)-\mathbf{u}_i^t,\mathbf{u}_i^t-g_i(\mathbf{w}_t,\mathcal{B}_q)\rangle, \tag{49}$$

where the last equality is due to the fact $\|\mathbf{u}_i^t-g_i(\mathbf{w}_t)\|^2=\|\mathbf{u}_i^{t+1}-g_i(\mathbf{w}_t)\|^2$ and $\|\mathbf{u}_i^{t+1}-\mathbf{u}_i^t\|^2=0$ for all $i\notin\mathcal{B}$. Taking expectation over the randomness at iteration t, we have

$$\begin{aligned}\mathbb{E}_t[\clubsuit]&\le\frac{1}{\gamma_0}\|\mathbf{u}^t-g(\mathbf{w}_t)\|^2-\frac{1}{\gamma_0}\mathbb{E}_t[\|\mathbf{u}^{t+1}-g(\mathbf{w}_t)\|^2]-\frac{1}{\gamma_0}\mathbb{E}_t[\|\mathbf{u}^{t+1}-\mathbf{u}^t\|^2]\\&\quad-2\mathbb{E}_t\left[\sum_{i\in\mathcal{B}}\|\mathbf{u}_i^t-g_i(\mathbf{w}_t)\|^2\right]\\&=\frac{1}{\gamma_0}\|\mathbf{u}^t-g(\mathbf{w}_t)\|^2-\frac{1}{\gamma_0}\mathbb{E}_t[\|\mathbf{u}^{t+1}-g(\mathbf{w}_t)\|^2]-\frac{1}{\gamma_0}\mathbb{E}_t[\|\mathbf{u}^{t+1}-\mathbf{u}^t\|^2]\\&\quad ss-2\frac{|\mathcal{B}|}{|\mathcal{S}|}\|\mathbf{u}^t-g(\mathbf{w}_t)\|^2.\end{aligned} \tag{50}$$

On the other hand, for $\spadesuit$, we can establish the following bound

$$\begin{aligned}\spadesuit&\le\left(\frac{1}{\gamma_0}-1\right)\sum_{i\in\mathcal{B}}\|\mathbf{u}_i^{t+1}-\mathbf{u}_i^t\|^2+\frac{1}{\frac{1}{\gamma_0}-1}\sum_{i\in\mathcal{B}}\|g_i(\mathbf{w}_t;\mathcal{B}_q)-g_i(\mathbf{w}_t)\|^2\\&\le\left(\frac{1}{\gamma_0}-1\right)\sum_{i\in\mathcal{B}}\|\mathbf{u}_i^{t+1}-\mathbf{u}_i^t\|^2+2\gamma_0\sum_{i\in\mathcal{B}}\|g_i(\mathbf{w}_t;\mathcal{B}_q)-g_i(\mathbf{w}_t)\|^2\\&\le\left(\frac{1}{\gamma_0}-1\right)\|\mathbf{u}^{t+1}-\mathbf{u}^t\|^2+2\gamma_0|\mathcal{B}|\sigma^2,\end{aligned} \tag{51}$$

where we employ the inequality $2a^\top b\le\gamma\|a\|^2+\frac{1}{\gamma}\|b\|^2$, the assumption $\gamma_0<1/2$, and the variance for $g_i(\mathbf{w}_t;B_q)$ in our derivation. Then by plugging (49), (50), (51) back into (48), we achieve the following relationship:

$$\begin{aligned}&\mathbb{E}[\|\mathbf{u}^{t+1}-g(\mathbf{w}_t)\|^2]\\&\le\mathbb{E}[\|\mathbf{u}^{t+1}-\mathbf{u}^t\|^2]+\mathbb{E}[\|\mathbf{u}^t-g(\mathbf{w}_t)\|^2]+\frac{1}{\gamma_0}\mathbb{E}[\|\mathbf{u}^t-g(\mathbf{w}_t)\|^2]-\frac{1}{\gamma_0}\mathbb{E}[\|\mathbf{u}^{t+1}-g(\mathbf{w}_t)\|^2]\\&\quad-\frac{1}{\gamma_0}\mathbb{E}[\|\mathbf{u}^{t+1}-\mathbf{u}^t\|^2]-2\frac{|\mathcal{B}|}{|\mathcal{S}|}\mathbb{E}[\|\mathbf{u}^t-g(\mathbf{w}_t)\|^2]+\left(\frac{1}{\gamma_0}-1\right)\mathbb{E}[\|\mathbf{u}^{t+1}-\mathbf{u}^t\|^2]+2\gamma_0|\mathcal{B}|\sigma^2\\&=\left(1+\frac{1}{\gamma_0}-2\frac{|\mathcal{B}|}{|\mathcal{S}|}\right)\mathbb{E}[\|\mathbf{u}^t-g(\mathbf{w}_t)\|^2]-\frac{1}{\gamma_0}\mathbb{E}[\|\mathbf{u}^{t+1}-g(\mathbf{w}_t)\|^2]+2\gamma_0|\mathcal{B}|\sigma^2.\end{aligned}$$

Note that $\frac{\left(1+\frac{1}{\gamma_0}-2\frac{|\mathcal{B}|}{|\mathcal{S}|}\right)}{1+\frac{1}{\gamma_0}}=1-\frac{2|\mathcal{B}|\gamma_0}{(1+\gamma_0)|\mathcal{S}|}\le1-\frac{|\mathcal{B}|\gamma_0}{|\mathcal{S}|}$ and $(1+\frac{a}{2})(1-a)\le1-\frac{a}{2}$, thus we obtain

$$\mathbb{E}[\|\mathbf{u}^{t+1}-g(\mathbf{w}_t)\|^2]\le\left(1-\frac{|\mathcal{B}|\gamma_0}{|\mathcal{S}|}\right)\mathbb{E}[\|\mathbf{u}^t-g(\mathbf{w}_t)\|^2]+2\gamma_0^2|\mathcal{B}|\sigma^2.$$

Now, we can employ the above relationship and inequality $\|a+b\|^2\le(1+\gamma)\|a\|^2+(1+\frac{1}{\gamma})\|b\|^2$, $\gamma>0$, and establish the recursive relationship for the tracking error $\|\mathbf{u}^t-g(\mathbf{w}_t)\|^2$:

$$\mathbb{E}[\|\mathbf{u}^{t+1}-g(\mathbf{w}_{t+1})\|^2]\le\left(1+\frac{|\mathcal{B}|\gamma_0}{2|\mathcal{S}|}\right)\mathbb{E}[\|\mathbf{u}^{t+1}-g(\mathbf{w}_t)\|^2]+\left(1+\frac{2|\mathcal{S}|}{|\mathcal{B}|\gamma_0}\right)\mathbb{E}[\|g(\mathbf{w}_t)-g(\mathbf{w}_{t+1})\|^2]$$

$$\leq \left(1+\frac{|\mathcal{B}|\gamma_0}{2|\mathcal{S}|}\right)\left[(1-\frac{|\mathcal{B}|\gamma_0}{|\mathcal{S}|})\mathbb{E}[\|\mathbf{u}^t-g(\mathbf{w}_t)\|^2]+2\gamma_0^2|\mathcal{B}|\sigma^2\right]+\left(1+\frac{2|\mathcal{S}|}{|\mathcal{B}|\gamma_0}\right)C_g^2|\mathcal{S}|\mathbb{E}[\|\mathbf{w}_t-\mathbf{w}_{t+1}\|^2]$$
$$\leq \left(1-\frac{|\mathcal{B}|\gamma_0}{2|\mathcal{S}|}\right)\mathbb{E}[\|\mathbf{u}^t-g(\mathbf{w}_t)\|^2]+4\gamma_0^2|\mathcal{B}|\sigma^2+\frac{4|\mathcal{S}|^2C_g^2\eta_1^2}{|\mathcal{B}|\gamma_0}\mathbb{E}[\|\mathbf{m}_{t+1}\|^2].$$

Finally, based on the above recursive relationship, summing over $t=0,\ldots,T-1$ and using the offset cancellation of tracking errors at different times, we can derive the desired bound:

$$\sum_{t=0}^{T}\mathbb{E}[\|\mathbf{u}^t-g(\mathbf{w}_t)\|^2]\leq\frac{2|\mathcal{S}|}{|\mathcal{B}|\gamma_0}\mathbb{E}[\|\mathbf{u}^0-g(\mathbf{w}_0)\|^2]+8|\mathcal{S}|\gamma_0\sigma^2T+\frac{8|\mathcal{S}|^3C_g^2\eta_1^2}{|\mathcal{B}|^2\gamma_0^2}\sum_{t=0}^{T-1}\mathbb{E}[\|\mathbf{m}_{t+1}\|^2].$$

□

Proof of Lemma 10 We try to derive the tracking error bound for the moving average estimator $\mathbf{s}_q^t$ of the function $\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda(\mathbf{w}_t))$. Recall and define the following notations

$$\mathbf{s}_q^{t+1}=\begin{cases}(1-\gamma_0')\mathbf{s}_q^t+\gamma_0'\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q^t;\mathcal{B}_q) & \text{if } q\in\mathcal{B}\\ \mathbf{s}_q^t & \text{o.w.}\end{cases},\quad \widetilde{\mathbf{s}}_q^t=(1-\gamma_0')\mathbf{s}_q^t+\gamma_0'\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q^t;\mathcal{B}_q),$$

where $\widetilde{\mathbf{s}}_q^t$ represents the components of $\mathbf{s}$ updates in the t-th iteration. Our strategy here remains the same: first, we achieve the estimation error bound for the components updated in the t-th iteration, and then we establish the estimation error bound for all components. We first analyze the tracking error for $\widetilde{\mathbf{s}}_q^t$:

$$\underbrace{\mathbb{E}_t[\|\widetilde{\mathbf{s}}_q^t-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))\|^2]}_{(*)}=\mathbb{E}_t[\|(1-\gamma_0')\mathbf{s}_q^t+\gamma_0'\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q^t;\mathcal{B}_q)-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))\|^2]$$
$$=\mathbb{E}_t[\|(1-\gamma_0')[\mathbf{s}_q^t-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))]+\gamma_0'[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q^t;\mathcal{B}_q)-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q^t)]$$
$$+\gamma_0'[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q^t)-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))]\|^2]$$
$$=\|(1-\gamma_0')[\mathbf{s}_q^t-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))]+\gamma_0'[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q^t)-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))]\|^2$$
$$+\mathbb{E}_t[\|\gamma_0'[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q^t;\mathcal{B}_q)-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q^t)]\|^2]$$
$$\leq\left(1+\frac{\gamma_0'}{2}\right)(1-\gamma_0')^2\|\mathbf{s}_q^t-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))\|^2$$
$$+\left(1+\frac{2}{\gamma_0'}\right)\gamma_0'^2\|\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q^t)-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))\|^2+\gamma_0'^2\sigma^2$$
$$\leq\left(1-\frac{\gamma_0'}{2}\right)\|\mathbf{s}_q^t-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))\|^2+4\gamma_0'L^2_{L\lambda\lambda}\|\lambda_q^t-\lambda_q(\mathbf{w}_t)\|^2+\gamma_0'^2\sigma^2,\tag{52}$$

where we use the fact that $\mathbb{E}_t[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q^t;\mathcal{B}_q)]=\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q^t)$ in the third equality, and Young's inequality in the first inequality. Note that for the randomness of query sampling, we further have

$$\mathbb{E}_t[\|\mathbf{s}_q^{t+1}-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))\|^2]$$
$$=\frac{|\mathcal{B}|}{N}\mathbb{E}_t[\|\widetilde{\mathbf{s}}_q^t-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))\|^2]+\frac{N-|\mathcal{B}|}{N}\|\mathbf{s}_q^t-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))\|^2,$$

which follows that

$$(*)=\frac{N}{|\mathcal{B}|}\mathbb{E}_t[\|\mathbf{s}_q^{t+1}-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))\|^2]-\frac{N-|\mathcal{B}|}{|\mathcal{B}|}\|\mathbf{s}_q^t-\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))\|^2.$$

Then by plugging the above equality into inequality (52), we obtain

$$\frac{N}{|\mathcal{B}|}\mathbb{E}_t[\|\mathbf{s}_q^{t+1} - \nabla^2_{\lambda\lambda} L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))\|^2] - \frac{N-|\mathcal{B}|}{|\mathcal{B}|}\|\mathbf{s}_q^t - \nabla^2_{\lambda\lambda} L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))\|^2$$
$$\leq \left(1 - \frac{\gamma_0'}{2}\right)\|\mathbf{s}_q^t - \nabla^2_{\lambda\lambda} L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))\|^2 + 4\gamma_0' L^2_{L\lambda\lambda}\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2 + \gamma_0'^2\sigma^2.$$

It follows

$$\mathbb{E}_t[\|\mathbf{s}_q^{t+1} - \nabla^2_{\lambda\lambda} L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))\|^2] \leq \left(1 - \frac{|\mathcal{B}|\gamma_0'}{2N}\right)\|\mathbf{s}_q^t - \nabla^2_{\lambda\lambda} L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))\|^2$$
$$+ \frac{4|\mathcal{B}|\gamma_0' L^2_{L\lambda\lambda}}{N}\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2 + \frac{|\mathcal{B}|\gamma_0'^2\sigma^2}{N}.$$

Now, we can establish the recursive relationship for the tracking error $\|\mathbf{s}_q^t - \nabla^2_{\lambda\lambda} L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))\|^2$:

$$\mathbb{E}_t[\|\mathbf{s}_q^{t+1} - \nabla^2_{\lambda\lambda} L_q(\mathbf{w}_{t+1}, \lambda_q(\mathbf{w}_{t+1}))\|^2] \leq \left(1 - \frac{|\mathcal{B}|\gamma_0'}{4N}\right)\|\mathbf{s}_q^t - \nabla^2_{\lambda\lambda} L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))\|^2$$
$$+ \frac{8|\mathcal{B}|\gamma_0' L^2_{L\lambda\lambda}}{N}\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2 + \frac{2|\mathcal{B}|\gamma_0'^2\sigma^2}{N} + \frac{8NL^2_{L\lambda\lambda}(1+C_\lambda^2)}{|\mathcal{B}|\gamma_0'}\mathbb{E}_t[\|\mathbf{w}_t - \mathbf{w}_{t+1}\|^2],$$

where we use the assumption $\gamma_0' \leq 1 \leq \frac{4N}{|\mathcal{B}|}$ i.e. $\frac{4N}{|\mathcal{B}|\gamma_0'} \geq 1$. Taking expectation over all randomness and taking summation over all queries and $t = 0, \ldots, T-1$, we have

$$\mathbb{E}[\|\mathbf{s}^{t+1} - \nabla^2_{\lambda\lambda} L(\mathbf{w}_{t+1}, \lambda(\mathbf{w}_{t+1}))\|^2] \leq \left(1 - \frac{|\mathcal{B}|\gamma_0'}{4N}\right)\mathbb{E}[\|\mathbf{s}^t - \nabla^2_{\lambda\lambda} L(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))\|^2]$$
$$+ \frac{8|\mathcal{B}|\gamma_0' L^2_{L\lambda\lambda}}{N}\mathbb{E}[\|\lambda^t - \lambda(\mathbf{w}_t)\|^2] + 2|\mathcal{B}|\gamma_0'^2\sigma^2 + \frac{8N^2L^2_{L\lambda\lambda}(1+C_\lambda^2)}{|\mathcal{B}|\gamma_0'}\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}_{t+1}\|^2].$$

At last, based on the above recursive relationship, we take summation over $t = 0, \ldots, T-1$ and obtain the following desired bound

$$\sum_{t=0}^{T}\mathbb{E}[\|\mathbf{s}^t - \nabla^2_{\lambda\lambda} L(\mathbf{w}_t, \lambda(\mathbf{w}_t))\|^2] \leq \frac{4N}{|\mathcal{B}|\gamma_0'}\|\mathbf{s}^0 - \nabla^2_{\lambda\lambda} L(\mathbf{w}_0, \lambda_q(\mathbf{w}_0))\|^2$$
$$+ 32L^2_{L\lambda\lambda}\sum_{t=0}^{T-1}\mathbb{E}[\|\lambda^t - \lambda(\mathbf{w}_t)\|^2] + 8N\gamma_0' T\sigma^2 + \frac{32N^3L^2_{L\lambda\lambda}(1+C_\lambda^2)}{|\mathcal{B}|^2\gamma_0'^2}\sum_{t=0}^{T-1}\mathbb{E}[\|\mathbf{w}_t - \mathbf{w}_{t+1}\|^2].$$

□

Appendix F SONG and K-SONG with faster convergence

Similar to our analysis of SONG and K-SONG, we address the compositional bilevel problem described in (25) under the same assumptions as in Assumption 1. We reorganize the set $\mathcal{S}$ so each pair $(q, \mathbf{x}_i^q)$ is indexed by a new, singular index i, with $\mathcal{S}$ now denoting this updated indexing set. Subscript q indicates the variable or function block corresponding to query q.

We outline Faster K-SONG$^{v1/v2}$ again in Algorithm 9. Note that the terms τ_t and η_t in the updates for λ^{t+1} and $\mathbf{w}_{t+1}$ are primarily for theoretical discourse. Practically, $\tau\tau_t$ and $\alpha\eta_t$ are considered as two distinct learning rate parameters. Let $\mathbb{E}[\cdot]$ denote the expectation over the

Algorithm 9 Restate Faster K-SONG$^{\text{v1/v2}}$ with new indexing

Require: $\mathbf{w}_0, \mathbf{w}_1$, initialize $\mathbf{m}_0, \lambda^0, \lambda^1, \mathbf{z}^0, \mathbf{u}^0, \mathbf{s}^0, \mathbf{u}^1, \mathbf{s}^1, \mathbf{v}^0, \mathbf{r}^0$ to 0, update type: v1 or v2
Ensure: $\mathbf{w}_{T+1}$
for $t = 1, 2, \ldots, T$ **do**
  Draw some relevant Q-I pairs $\mathcal{B} = \{(q, \mathbf{x}_i^q)\} \subset \mathcal{S}$
  For each $q \in \mathcal{B}$ draw a batch of items $\mathcal{B}_q \subset \mathcal{S}_q$
  if using v1 type update **then**

$$\mathbf{u}_i^{t+1} = \begin{cases} (1-\gamma_{u,t})\mathbf{u}_i^t + \gamma_{u,t} g_i(\mathbf{w}_t; \mathcal{B}_q) + \beta_{u,t}(g_i(\mathbf{w}_t; \mathcal{B}_q) - g_i(\mathbf{w}_{t-1}; \mathcal{B}_q)) & \text{if } i \in \mathcal{B} \\ \mathbf{u}_i^t & \text{o.w.} \end{cases}$$

$$\mathbf{s}_q^{t+1} = \begin{cases} (1-\gamma_{s,t})\mathbf{s}_q^t + \gamma_{s,t}\nabla^2_{\lambda\lambda} L_q(\mathbf{w}_t, \lambda_q^t; \mathcal{B}_q) \\ \qquad +\beta_{s,t}(\nabla^2_{\lambda\lambda} L_q(\mathbf{w}_t, \lambda_q^t; \mathcal{B}_q) - \nabla^2_{\lambda\lambda} L_q(\mathbf{w}_{t-1}, \lambda_q^{t-1}; \mathcal{B}_q)) & \text{if } q \in \mathcal{B} \\ \mathbf{s}_q^t & \text{o.w.} \end{cases}$$

  else // using v2 type update
    Compute $\nabla_u \tilde{g}_i(\mathbf{u}, \mathbf{w}; \mathcal{B}_q) = \mathbf{u}_i - g(\mathbf{w}; \mathbf{x}_i^q, \mathcal{B}_q)$

$$\mathbf{v}_i^t = \begin{cases} (1-\gamma_{v,t})\mathbf{v}_i^{t-1} + \gamma_{v,t}\nabla_u \tilde{g}_i(\mathbf{u}^t, \mathbf{w}_t; \mathcal{B}_q) \\ \qquad +\beta_{v,t}(\nabla_u \tilde{g}_i(\mathbf{u}^t, \mathbf{w}_t; \mathcal{B}_q) - \nabla_u \tilde{g}_i(\mathbf{u}^{t-1}, \mathbf{w}_{t-1}; \mathcal{B}_q)) & \text{if } q \in \mathcal{B} \\ \mathbf{v}_i^{t-1} & \text{o.w.} \end{cases}$$

    Compute $\mathbf{r}_q^t$ according to (18)
    Update $\mathbf{u}_i^{t+1} = \mathbf{u}_i^t - \tau\tau_t \mathbf{v}_i^t$, $\mathbf{s}_q^{t+1} = \mathbf{s}_q^t - \tau\tau_t \mathbf{r}_q^t$
  end if

$$\mathbf{z}_q^t = \begin{cases} (1-\gamma_{z,t})\mathbf{z}_q^{t-1} + \gamma_{z,t}\nabla L_q(\mathbf{w}_t, \lambda_q^t; \mathcal{B}_q)) \\ \qquad +\beta_{z,t}(\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t; \mathcal{B}_q) - \nabla_\lambda L_q(\mathbf{w}_{t-1}, \lambda_q^{t-1}; \mathcal{B}_q))) & \text{if } q \in \mathcal{B} \\ \mathbf{z}_q^{t-1} & \text{o.w.} \end{cases}$$

$$\lambda_q^{t+1} = \begin{cases} \lambda_q^t - \tau\tau_t \mathbf{z}_q^t & \text{if } q \in \mathcal{B} \\ \lambda_q^t & \text{o.w.} \end{cases}$$

  Compute stochastic gradient estimator $G(\mathbf{w}_{t-1})$ and $G(\mathbf{w}_t)$ according to (26)
  $\mathbf{m}_t = (1-\gamma_{m,t})(\mathbf{m}_{t-1} - G(\mathbf{w}_{t-1})) + G(\mathbf{w}_t)$
  $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha\eta_t \mathbf{m}_t$
end for

randomness of the algorithm until the current iteration, and $\mathbb{E}_t[\cdot]$ represent the expectation over the randomness at iteration t.

For the two new quadratic functions introduced in the v2 type update rules, which are both smooth and strongly convex, thus we make the following assumptions:

Assumption 2 (i) Functions $\tilde{g}_{q,i}(\mathbf{u}, \mathbf{w})$ defined in (14) are $L_{\tilde{g}}$-smooth and $\mu_{\tilde{g}}$-strongly convex for all (q, i).
(ii) Functions $\phi_q(\mathbf{s}, \hat{\lambda}_q(\mathbf{w}), \mathbf{w})$ in (17) are L_ϕ-smooth and μ_ϕ-strongly convex for all q.

Theorem 5 (Restate of Theorem 3 with parameter specifics). *Let Assumption 1 and 2 hold and apply Faster K-SONG in Algorithm 9 to solve the problem in (25) with the following parameters:*

$$\tau \le \min\left\{\frac{1}{2L_L}, \frac{8N}{\mu_L|\mathcal{B}|}\right\}, \alpha \le \min\left\{\frac{|\mathcal{B}|}{12C_7N}, \frac{|\mathcal{B}|}{12C_8|\mathcal{S}|}, \frac{1}{C_\lambda}\sqrt{\frac{\tau\mu_L|\mathcal{B}|}{96CN}}\right\}, \eta_t = \tau_t = \frac{c}{(c_0+t)^{1/3}},$$

$$\gamma_{z,t+1} = \frac{N\eta_t^2}{|\mathcal{B}|}\left(\frac{1}{7L_F c^3} + \frac{8C\alpha\tau_t\tau|\mathcal{B}|}{\mu_L N\eta_t}\right), \quad \gamma_{u,t+1} = \left(\frac{2|\mathcal{S}|}{7|\mathcal{B}|L_F c^3} + \frac{4C_1\alpha|\mathcal{S}|}{|\mathcal{B}|}\right)\eta_t^2,$$

$$\gamma_{s,t+1} = \left(\frac{2N}{7|\mathcal{B}|L_F c^3} + \frac{4C_2\alpha N}{|\mathcal{B}|}\right)\eta_t^2, \quad \gamma_{m,t+1} = \left(\frac{1}{7L_F c^3} + \alpha\right)\eta_t^2,$$

$$\beta_{u,t} = 1 - \gamma_{u,t} + \frac{|\mathcal{S}| - |\mathcal{B}|}{|\mathcal{B}|(1-\gamma_{u,t})}, \beta_{z,t} = 1 - \gamma_{z,t} + \frac{N - |\mathcal{B}|}{|\mathcal{B}|(1-\gamma_{z,t})}, \beta_{s,t} = 1 - \gamma_{s,t} + \frac{N - |\mathcal{B}|}{|\mathcal{B}|(1-\gamma_{s,t})},$$

$$C \geq \max\left\{\frac{8C_0N}{\tau\mu_L|\mathcal{B}|}, \frac{2(C_9+2C_5)N^2}{3\alpha|\mathcal{B}|^2}\right\}, c_0 \geq \max\left\{2, (4L_Fc)^3, \left(\frac{8N}{7|\mathcal{B}|L_Fc}\right)^{3/2}, \left(\frac{8|\mathcal{S}|}{7|\mathcal{B}|L_Fc}\right)^{3/2}, \left(\frac{32C\alpha\tau c^2}{\mu_L}\right)^{3/2}, \left(\frac{16C_1\alpha|\mathcal{S}|c^2}{|\mathcal{B}|}\right)^{3/2}, \left(\frac{64|\mathcal{S}|C_3}{7L_F|\mathcal{B}|c}\right)^{3/2}, \left(\frac{128C_1|\mathcal{S}|C_3\alpha c^2}{|\mathcal{B}|}\right)^{3/2}, \left(\frac{16C_2\alpha Nc^2}{|\mathcal{B}|}\right)^{3/2}, \left(\frac{64NC_6}{7L_F|\mathcal{B}|c}\right)^{3/2}, \left(\frac{128C_2NC_6\alpha c^2}{|\mathcal{B}|}\right)^{3/2}, \left(\frac{4}{7L_Fc}\right)^{3/2}, \left(4\alpha c^2\right)^{3/2}\right\},$$

where $C_0, C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9$ are constants specified in the proof. Algorithm 9 ensures that after $T = \mathcal{O}(\frac{1}{\epsilon^3})$ iterations, we can find an ϵ-stationary solution of $F(\mathbf{w}_t)$, i.e., $\mathbb{E}\left[\sum_{t=1}^{T}\frac{1}{T}\|\nabla F(\mathbf{w}_t)\|^2\right] \leq \mathcal{O}\left(\frac{1}{T^{2/3}}\right)$.

Now we present the convergence analysis of Theorem 5. First of all, by using the smoothness of $F(\mathbf{w})$ (proved in Lemma 5), we have the following lemma, which is similar to lemma 6.

Lemma 11 *Consider the update $\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha\eta_t\mathbf{m}_t$. Then under Assumption 1, with $\alpha\eta_t L_F \leq \frac{1}{2}$, we have*

$$F(\mathbf{w}_{t+1}) \leq F(\mathbf{w}_t) + \frac{\alpha\eta_t}{2}||\nabla F(\mathbf{w}_t) - \mathbf{m}_t||^2 - \frac{\alpha\eta_t}{2}||\nabla F(\mathbf{w}_t)||^2 - \frac{\alpha\eta_t}{4}||\mathbf{m}_t||^2.$$

Then, we aim to derive an upper bound of $||\nabla F(\mathbf{w}_t) - \mathbf{m}_t||^2$ in lemma 11. Note the update rule of $\mathbf{m}_t$ in Algorithm 9 is more complicated than that of SONG/K-SONG, and we introduce the following lemma to decompose this error into several terms that can be bounded separately:

Lemma 12 *Let $\sum_{q\in\mathcal{Q}}\left\|\lambda_q(\mathbf{w}_t) - \lambda_q^t\right\|^2 = \left\|\lambda(\mathbf{w}_t) - \lambda^t\right\|^2$, $\sum_{i\in\mathcal{S}}\left\|\mathbf{u}_i^t - g_i(\mathbf{w}_t)\right\|^2 = \left\|\mathbf{u}^t - g(\mathbf{w}_t)\right\|^2$ and $\sum_{q\in\mathcal{Q}}\left\|\mathbf{s}_q^t - \nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t, \lambda_q^t)\right\|^2 = \left\|\mathbf{s}^t - \nabla^2_{\lambda\lambda}L(\mathbf{w}_t, \lambda^t)\right\|^2$. Consider the updates in Algorithm 9, under Assumption 1, for all $t > 0$, we have*

$$\begin{aligned}\|\nabla F(\mathbf{w}_t) - \mathbf{m}_t\|^2 \leq & 2\left\|\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_t) - \mathbf{m}_t\right\|^2 + \frac{4C_0}{N}\left\|\lambda(\mathbf{w}_t) - \lambda^t\right\|^2 \\ & + \frac{4C_1}{|\mathcal{S}|}\left\|\mathbf{u}^t - g(\mathbf{w}_t)\right\|^2 + \frac{4C_2}{N}\left\|\mathbf{s}^t - \nabla^2_{\lambda\lambda}L(\mathbf{w}_t, \lambda^t)\right\|^2,\end{aligned}$$

where C_0, C_1, C_2 are constants given in the proof.

Next, we will bound each term on the RHS of the above lemma separately. We first bound the first term $\left\|\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_t) - \mathbf{m}_t\right\|^2$ according to the following lemma

Lemma 13 *Let $\sum_{i\in\mathcal{S}}\left\|\mathbf{u}_i^t - \mathbf{u}_i^{t-1}\right\|^2 = \left\|\mathbf{u}^t - \mathbf{u}^{t-1}\right\|^2$, $\sum_{q\in\mathcal{Q}}\left\|\lambda_q^t - \lambda_q^{t-1}\right\|^2 = \left\|\lambda^t - \lambda^{t-1}\right\|^2$ and $\sum_{q\in\mathcal{Q}}\left\|\mathbf{s}_q^t - \mathbf{s}_q^{t-1}\right\|^2 = \left\|\mathbf{s}^t - \mathbf{s}^{t-1}\right\|^2$. Assume $\mathbb{E}\left[\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}G_i(\mathbf{w}_t) - \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_t)\right] \leq \sigma^2$, then*

$$\mathbb{E}\left[\left\|\mathbf{m}_t - \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_t)\right\|^2\right] \leq (1-\gamma_{m,t})\mathbb{E}\left[\left\|\mathbf{m}_{t-1} - \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_{t-1})\right\|^2\right] + 2\gamma_{m,t}^2\sigma^2$$

$$+\frac{2C_3}{|\mathcal{S}|}\left\|\mathbf{u}^t-\mathbf{u}^{t-1}\right\|^2+2C_4\left\|\mathbf{w}_t-\mathbf{w}_{t-1}\right\|^2+\frac{2C_5}{N}\left\|\lambda^t-\lambda^{t-1}\right\|^2+\frac{2C_6}{N}\left\|\mathbf{s}^t-\mathbf{s}^{t-1}\right\|^2,$$

where C_3, C_4, C_5, C_6 are constants given in the proof.

To bound the $\left\|\lambda_q(\mathbf{w}_t)-\lambda_q^t\right\|^2$ term in Lemma 12, we can use the following lemma.

Lemma 14 *Consider the update in Algorithm 9. Then under Assumption 1, with $\tau_t \le \frac{1}{2}$ and $\tau_t\tau \le \frac{4N}{\mu_L|\mathcal{B}|}$, we have*

$$\mathbb{E}\left[\left\|\lambda^{t+1}-\lambda(\mathbf{w}_{t+1})\right\|^2\right]\le\left(1-\frac{\tau\tau_t\mu_L|\mathcal{B}|}{4N}\right)\mathbb{E}\left[\left\|\lambda(\mathbf{w}_t)-\lambda^t\right\|^2\right]+\frac{8\tau_t\tau|\mathcal{B}|}{\mu_L N}\left\|\nabla_\lambda L(\mathbf{w}_t,\lambda^t)-\mathbf{z}^t\right\|^2$$
$$-\frac{3\tau|\mathcal{B}|}{\tau_t N}\left(\frac{1}{\tau}-L_L\right)\left\|\lambda^{t+1}-\lambda^t\right\|^2+\frac{8N^2C_\lambda^2}{\tau\tau_t\mu_L|\mathcal{B}|}\mathbb{E}\left[\left\|\mathbf{w}_{t+1}-\mathbf{w}_t\right\|^2\right].$$

When we adopt v2 type update rules to estimate $g(\mathbf{w};\mathbf{x}_i^q,\mathcal{S}_q)$ and $\nabla^2_{\lambda\lambda}L_q(\hat{\lambda}(\mathbf{w});\mathbf{w})$, it is notable that their corresponding quadratic problems are smooth and strongly convex, and their update rules are fully consistent with that for λ. Therefore, we can follow the derivation process of the above lemma to establish the approximation error bounds for these functions, as demonstrated below.

Lemma 15 (The error bound for $g(\mathbf{w};\mathbf{x}_i^q,\mathcal{S}_q)$ when using v2 type update). *Consider the update in Algorithm 9. Then under Assumption 1 and 2, with $\tau_t \le \frac{1}{2}$ and $\tau_t\tau \le \frac{4|\mathcal{S}|}{\mu_{\tilde{g}}|\mathcal{B}|}$, we have*

$$\mathbb{E}\left[\left\|\mathbf{u}^{t+1}-g(\mathbf{w}_{t+1})\right\|^2\right]\le\left(1-\frac{\tau\tau_t\mu_{\tilde{g}}|\mathcal{B}|}{4|\mathcal{S}|}\right)\mathbb{E}\left[\left\|g(\mathbf{w}_t)-\mathbf{u}^t\right\|^2\right]+\frac{8\tau_t\tau|\mathcal{B}|}{\mu_{\tilde{g}}|\mathcal{S}|}\left\|\nabla_u\tilde{g}(\mathbf{w}_t)-\mathbf{v}^t\right\|^2$$
$$-\frac{3\tau|\mathcal{B}|}{\tau_t|\mathcal{S}|}\left(\frac{1}{\tau}-L_{\tilde{g}}\right)\left\|\mathbf{u}^{t+1}-\mathbf{u}^t\right\|^2+\frac{8|\mathcal{S}|^2C_g^2}{\tau\tau_t\mu_{\tilde{g}}|\mathcal{B}|}\mathbb{E}\left[\|\mathbf{w}_{t+1}-\mathbf{w}_t\|^2\right].$$

Lemma 16 (The error bound for $\nabla^2_{\lambda\lambda}L_q(\hat{\lambda}(\mathbf{w});\mathbf{w})$ when using v2 type update). *Consider the update in Algorithm 9. Then under Assumption 1 and 2, with $\tau_t \le \frac{1}{2}$ and $\tau_t\tau \le \frac{4N}{\mu_\phi|\mathcal{B}|}$, we have*

$$\mathbb{E}\left[\left\|\mathbf{s}^{t+1}-\nabla^2_{\lambda\lambda}L_q(\hat{\lambda}(\mathbf{w}_{t+1});\mathbf{w}_{t+1})\right\|^2\right]\le\left(1-\frac{\tau\tau_t\mu_\phi|\mathcal{B}|}{4N}\right)\mathbb{E}\left[\left\|\mathbf{s}^t-\nabla^2_{\lambda\lambda}L_q(\hat{\lambda}(\mathbf{w}_t);\mathbf{w}_t)\right\|^2\right]$$
$$+\frac{8\tau_t\tau|\mathcal{B}|}{\mu_\phi N}\left\|\nabla_s\phi(\mathbf{w}_t)-\mathbf{r}^t\right\|^2-\frac{3\tau|\mathcal{B}|}{\tau_t N}\left(\frac{1}{\tau}-L_\phi\right)\left\|\mathbf{s}^{t+1}-\mathbf{s}^t\right\|^2+\frac{8N^2L_{L\lambda\lambda}^2}{\tau\tau_t\mu_L|\mathcal{B}|}\mathbb{E}\left[\left\|\mathbf{w}_{t+1}-\mathbf{w}_t\right\|^2\right].$$

The following lemma bound the terms involving $\mathbf{u}^t, \mathbf{z}^t$ and $\mathbf{s}^t$ on the RHS in the inequalities of the above two lemmas.

Lemma 17 (Lemma 1 Jiang et al. (2022)). *Suppose $f_i, i = 1, 2, \cdots, n$ is a mapping, $\mathbb{E}[f_i(\mathbf{x};\xi)] = f_i(\mathbf{x})$ and $\mathbb{E}[f_i(\mathbf{x};\xi)-f_i(\mathbf{x})] \le \sigma^2$. In each iteration, we sample a mini-batch $\mathcal{M}$ with the size of m. let*

$$\mathbf{d}_i^t=\begin{cases}(1-\gamma_t)\mathbf{d}_i^{t-1}+\gamma_t f_i(\mathbf{x}_t;\xi_t)+\beta_t(f_i(\mathbf{x}_t;\xi_t)-f_i(\mathbf{x}_{t-1};\xi_t)) & \text{if } i\in\mathcal{M}\\ \mathbf{d}_i^{t-1} & \text{o.w.}\end{cases}.$$

By setting $\gamma_t \le \frac{1}{2}$ and $\beta_t = 1-\gamma_t+\frac{n-m}{m(1-\gamma_t)}$, for $t \ge 1$, we have

$$\mathbb{E}\left[\left\|\mathbf{d}^t - f(\mathbf{x}_t)\right\|^2\right] = \sum_{i=1}^{n} \mathbb{E}\left[\left\|\mathbf{d}_i^t - f_i(\mathbf{x}_t)\right\|^2\right]$$
$$\leq \left(1 - \frac{\gamma_t m}{n}\right) \mathbb{E}\left[\left\|\mathbf{d}^{t-1} - f(\mathbf{x}_{t-1})\right\|^2\right] + \frac{8n}{m} \sum_{i=1}^{n} \mathbb{E}\left[\left\|f_i(\mathbf{x}_t; \xi_t) - f_i(\mathbf{x}_{t-1}; \xi_t)\right\|^2\right] + 2m\gamma_t^2\sigma^2,$$
$$\mathbb{E}\left[\left\|\mathbf{d}^t - \mathbf{d}^{t-1}\right\|^2\right] = \sum_{i=1}^{n} \mathbb{E}\left[\left\|\mathbf{d}_i^t - \mathbf{d}_i^{t-1}\right\|^2\right]$$
$$\leq 2m\gamma_t^2\sigma^2 + \frac{4m\gamma_t^2}{n} \mathbb{E}\left[\left\|f(\mathbf{x}_{t-1}) - \mathbf{d}^{t-1}\right\|^2\right] + \frac{9n}{m} \sum_{i=1}^{n} \mathbb{E}\left[\left\|f_i(\mathbf{x}_t; \xi_t) - f_i(\mathbf{x}_{t-1}; \xi_t)\right\|^2\right].$$

Appendix F.1.1 Proof sketch

In terms of the overall approach for proving Theorem 5, we integrate some proof techniques of STORM and K-SONG. Here, we provide several key points to help the readers better understand the formal proof that follows.

1. First, similar to our previous analysis of SONG/K-SONG, Lemma 11 establishes the connection between the quality of the solution (denoted as $\|\nabla F(\mathbf{w}_t)\|$) and the gradient approximation error (denoted as $\|\nabla F(\mathbf{w}_t) - \mathbf{m}_t\|$).
2. Then, through Lemmas 12 and 13, one can observe that $\|\nabla F(\mathbf{w}_t) - \mathbf{m}_t\|$ can be further bounded by the approximation errors of several crucial inner functions (such as $\|\lambda(\mathbf{w}_t) - \lambda^t\|$, $\|\mathbf{u}^t - g(\mathbf{w}_t)\|$, $\|\mathbf{s}^t - \nabla^2_{\lambda\lambda} L(\mathbf{w}_t, \lambda^t)\|$) and by the differences of these functions at $\mathbf{w}_t$ and $\mathbf{w}_{t-1}$ (such as $\|\mathbf{u}^t - \mathbf{u}^{t-1}\|$, $\|\lambda^t - \lambda^{t-1}\|$, $\|\mathbf{s}^t - \mathbf{s}^{t-1}\|$).
3. Subsequently, we can establish the recursions for the above error terms. For the variables using the v1 type update, we can directly apply the results from MSVR, specifically Lemma 17. For the solutions of the lower-level problems λ and the variables using the v2 type update, we provide their recursions in Lemma 14, 15, and 16, respectively.
4. We employ the technique used in STORM to perform staggered summation on the above recursions. Specifically, for the terms $\|\mathbf{z}^t - \nabla_\lambda L(\mathbf{w}_t, \lambda^t)\|$, $\|\mathbf{u}^t - g(\mathbf{w}_t)\|$, and $\|\mathbf{s}^t - \nabla^2_{\lambda\lambda} L(\mathbf{w}_t, \lambda^t)\|$, we show their staggered subtraction results in (53), (57), and (58), respectively. Utilizing the techniques introduced in STORM, we set η_t and τ_t according to (55), which then allows us to simplify the RHS of (53), (57), and (58).
5. Finally, by summing Lemmas 11, 12, 13, and all the recursions obtained above, we derive (60). Through careful parameter settings, we can eliminate most of the irrelevant terms, ultimately establishing the desired bound.

Appendix F.2.2 Innovations in proof techniques

The problem we address involves SBO with multiple lower-level problems. Cutkosky and Orabona (2019) and Jiang et al. (2022) introduce variance-reduced estimators that effectively control function estimation errors, but their methods are not applicable to SBO problems. Guo et al. (2021a) consider a problem similar to ours, but do not implement parallel solving of lower-level problems. Our algorithms are the first to achieve both optimal convergence rate and parallel speed-up.

To achieve this, we not only employ advanced variance-reduced estimators but also propose new algorithm designs and proof techniques. From an algorithmic perspective, to better control the estimation error of lower-level solutions, we design a new variance-reduced stochastic gradient estimator $\mathbf{z}_q$ for updates, as opposed to directly use stochastic gradients

as in K-SONG. In proving the estimation error bound in Lemma 14, we fully utilized the definition of $\mathbf{z}_q$ along with the smoothness and strong convexity properties of lower-level functions L_q. Here, we present the key steps and results, omitting some intermediate details. The full details are provided in the proof of Lemma 14.

First, in the t-th iteration, the update rule for the sampled components is $\lambda_q^{t+1} = \lambda_q^t - \tau\tau_t\mathbf{z}_q^t$. We introduce an intermediate variable $\tilde{\lambda}_q^t = \lambda_q^t - \tau\mathbf{z}_q^t$ and set $\lambda_q^{t+1} = \bar{\lambda}_q^t = \lambda_q^t + \tau_t(\tilde{\lambda}_q^t - \lambda_q^t)$. Based on these definitions, we have the following estimation error relationship for the sampled components:

$$\|\bar{\lambda}_q^t - \lambda_q(\mathbf{w}_t)\|^2 = \|\lambda_q^t - \lambda_q(\mathbf{w}_t)\|^2 + \tau_t^2\|\tilde{\lambda}_q^t - \lambda_q^t\|^2 + 2\tau_t(\lambda_q^t - \lambda_q(\mathbf{w}_t))(\tilde{\lambda}_q^t - \lambda_q^t).$$

Next, by leveraging the strongly convexity of L_q and performing a fine-grained decomposition, we obtain

$$\begin{aligned} L_q(\mathbf{w}_t, \lambda) \geq\ & L_q(\mathbf{w}_t, \lambda_q^t) + \mathbf{z}_q^t(\lambda - \tilde{\lambda}_q^t) \\ & + (\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t) - \mathbf{z}_q^t)(\lambda - \tilde{\lambda}_q^t) + \nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t)(\tilde{\lambda}_q^t - \lambda_q^t) + \frac{\mu_L}{2}\|\lambda - \lambda_q^t\|^2. \end{aligned}$$

Note that we only present the final bound here, omitting the detailed decomposition process from the proof. The $(\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t) - \mathbf{z}_q^t)(\lambda - \tilde{\lambda}_q^t)$ term on the RHS is actually related to the estimation error of $\mathbf{z}_q$, whose bound can be established by Lemma 17. For the other two terms on the RHS, $\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t)(\tilde{\lambda}_q^t - \lambda_q^t)$ and $\mathbf{z}_q^t(\lambda - \tilde{\lambda}_q^t)$, we can obtain the following relationships using the smoothness of L_q and the update rule of $\mathbf{z}_q$, respectively,

$$\begin{aligned} & L_q(\mathbf{w}_t, \tilde{\lambda}_q^t) - \nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t)(\tilde{\lambda}_q^t - \lambda_q^t) - \frac{L_L}{2}\left\|\tilde{\lambda}_q^t - \lambda_q^t\right\|^2 \leq L_q(\mathbf{w}_t, \lambda_q^t), \\ & \mathbf{z}_q^t(\lambda - \tilde{\lambda}_q^t) = \frac{1}{\tau}(\lambda_q^t - \tilde{\lambda}_q^t)(\lambda - \lambda_q^t) + \frac{1}{\tau}\left\|\lambda_q^t - \tilde{\lambda}_q^t\right\|^2. \end{aligned}$$

Finally, by combining these relationships, we have

$$\begin{aligned} & \|\bar{\lambda}_q^t - \lambda_q(\mathbf{w}_t)\|^2 \\ & \leq \left(1 - \frac{\tau\tau_t\mu_L}{2}\right)\left\|\lambda_q(\mathbf{w}_t) - \lambda_q^t\right\|^2 + \frac{4\tau_t\tau}{\mu_L}\left\|\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t) - \mathbf{z}_q^t\right\|^2 - 2\tau\tau_t\left(\frac{3}{4\tau} - \frac{3}{4}L_L\right)\left\|\tilde{\lambda}_q^t - \lambda_q^t\right\|^2, \end{aligned}$$

which establishes the estimation error bound for the component updated in the t-th iteration.

The subsequent proof follows a similar approach to K-SONG, using properties of conditional expectation and inequality scaling to derive the final recursion. Note that the $\left\|\tilde{\lambda}_q^t - \lambda_q^t\right\|^2$ term on the RHS is defined to be equal to $\frac{1}{\tau_t}(\bar{\lambda}_q^t - \lambda_q^t) = \frac{1}{\tau_t}(\lambda_q^{t+1} - \lambda_q^t)$. This negative $\|\lambda_q^{t+1} - \lambda_q^t\|^2$ is crucial for canceling out the positive $\|\lambda_q^{t+1} - \lambda_q^t\|^2$ terms appearing elsewhere in the proof, thereby ensuring the convergence guarantee.

Proof of Theorem 5 Now, we can combine all lemmas presented above and prove the theoretical guarantee. First, we apply Lemma 17 to $\delta_{L\lambda,t} = \left\|\mathbf{z}^t - \nabla_\lambda L(\mathbf{w}_t, \lambda^t)\right\|^2$ and establish the following recursion for the tracking error of $\mathbf{z}_t$

$$\mathbb{E}\left[\delta_{L\lambda,t+1}\right] \leq \left(1 - \frac{\gamma_{z,t+1}|\mathcal{B}|}{N}\right)\mathbb{E}\left[\delta_{L\lambda,t}\right] + \frac{16N^2L_L^2}{|\mathcal{B}|}\left(\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2 + \frac{1}{N}\left\|\lambda^{t+1} - \lambda^t\right\|^2\right) + 2|\mathcal{B}|\gamma_{z,t+1}^2\sigma^2.$$

In the proof of this theorem, we will reorganize the obtained tracking error recursions into the following form

$$\mathbb{E}\left[\frac{\delta_{L\lambda,t+1}}{N\eta_t} - \frac{\delta_{L\lambda,t}}{N\eta_{t-1}}\right] \leq \frac{2|\mathcal{B}|\gamma_{z,t+1}^2\sigma^2}{N\eta_t} + \frac{1}{N}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \frac{\gamma_{z,t+1}|\mathcal{B}|}{N\eta_t}\right)\mathbb{E}\left[\delta_{L\lambda,t}\right]$$

$$+\frac{16NL_L^2}{|\mathcal{B}|\eta_t}\left(\|\mathbf{w}_{t+1}-\mathbf{w}_t\|^2+\frac{1}{N}\left\|\lambda^{t+1}-\lambda^t\right\|^2\right). \tag{53}$$

The advantage of this form is that the left side of the equation can be simplified through misalignment cancellation, while the right side can be simplified by carefully setting η_t, as we will demonstrate below.

As for the estimation error regarding $\lambda(\mathbf{w}_t)$, i.e., $\|\lambda^t-\lambda(\mathbf{w}_t)\|^2$, Lemma 14 provides its specific form. We reorganize it into the following form

$$\begin{aligned}\mathbb{E}\left[\frac{C\alpha}{N}(\delta_{\lambda,t+1}-\delta_{\lambda,t})\right]\le&-\frac{C\alpha\tau\tau_t\mu_L|\mathcal{B}|}{4N^2}\mathbb{E}\left[\delta_{\lambda,t}\right]+\frac{8C\alpha\tau_t\tau|\mathcal{B}|}{\mu_L N^2}\mathbb{E}\left[\delta_{L\lambda,t}\right]\\&-\frac{3C\alpha\tau|\mathcal{B}|}{\tau_t N^2}\left(\frac{1}{\tau}-L_L\right)\mathbb{E}\left[\left\|\lambda^{t+1}-\lambda^t\right\|^2\right]+\frac{8C\alpha NC_\lambda^2}{\tau\tau_t\mu_L|\mathcal{B}|}\mathbb{E}\left[\|\mathbf{w}_{t+1}-\mathbf{w}_t\|^2\right],\end{aligned} \tag{54}$$

where C will be given below.

We follow the STORM approach (Cutkosky & Orabona, 2019) to set η_t and τ_t in the algorithm, as this allows us to further simplify $\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}$. Specifically, we set $\eta_t=\tau_t=\frac{c}{(c_0+t)^{1/3}}$, and make $c_0\ge(4L_Fc)^3$ to ensure $\eta_t\le\frac{1}{4L_F}$. Thus, we have

$$\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}=\frac{(c_0+t)^{1/3}}{c}-\frac{(c_0+t-1)^{1/3}}{c}\le\frac{1}{3c(c_0+t-1)^{2/3}}\le\frac{2^{2/3}}{3c(c_0+t)^{2/3}}\le\frac{2^{2/3}}{3c^3}\eta_t^2\le\frac{1}{7L_Fc^3}\eta_t, \tag{55}$$

where the first inequality holds by the concavity of the function $f(x)=x^{1/3}$, i.e., $(x+y)^{1/3}\le x^{1/3}+\frac{y}{3x^{2/3}}$, the second inequality is because $c_0\ge2$. Then with $\gamma_{z,t+1}=\frac{N\eta_t^2}{|\mathcal{B}|}\left(\frac{1}{7L_Fc^3}+\frac{8C\alpha\tau_t\tau|\mathcal{B}|}{\mu_LN\eta_t}\right)$, where $\gamma_{z,t+1}<\frac{1}{2}$ for $c_0\ge\max\left\{(\frac{4N}{7|\mathcal{B}|L_Fc})^{3/2},(\frac{32C\alpha\tau c^2}{\mu_L})^{3/2}\right\}$, we can establish the following inequality by combining (53) and (54):

$$\begin{aligned}&\mathbb{E}\left[\frac{\delta_{L\lambda,t+1}}{N\eta_t}-\frac{\delta_{L\lambda,t}}{N\eta_{t-1}}\right]+\mathbb{E}\left[\frac{C\alpha}{N}(\delta_{\lambda,t+1}-\delta_{\lambda,t})\right]\\&\le\frac{2|\mathcal{B}|\gamma_{z,t+1}^2\sigma^2}{N\eta_t}+\frac{16NL_L^2}{|\mathcal{B}|\eta_t}\left(\|\mathbf{w}_{t+1}-\mathbf{w}_t\|^2+\frac{1}{N}\left\|\lambda^{t+1}-\lambda^t\right\|^2\right)\\&\quad-\frac{C\alpha\tau\tau_t\mu_L|\mathcal{B}|}{4N^2}\mathbb{E}\left[\delta_{\lambda,t}\right]-\frac{3C\alpha\tau|\mathcal{B}|}{\tau_tN^2}\left(\frac{1}{\tau}-L_L\right)\mathbb{E}\left[\left\|\lambda^{t+1}-\lambda^t\right\|^2\right]+\frac{8C\alpha NC_\lambda^2}{\tau\tau_t\mu_L|\mathcal{B}|}\mathbb{E}\left[\|\mathbf{w}_{t+1}-\mathbf{w}_t\|^2\right].\end{aligned} \tag{56}$$

In Lemma 13 and Lemma 17, we also established the tracking error recusions for $\delta_{g,t}=\left\|\mathbf{u}^t-g(\mathbf{w}_t)\right\|^2$, $\delta_{L\lambda\lambda,t}=\left\|\mathbf{s}^t-\nabla^2_{\lambda\lambda}L(\mathbf{w}_t,\lambda^t)\right\|^2$, and $\delta_{m,t}=\left\|\mathbf{m}_t-\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_t)\right\|^2$. Next, we perform transformations on these recursions similar to the ones above, and obtain

$$\begin{aligned}\mathbb{E}\left[\frac{\delta_{g,t+1}}{|\mathcal{S}|\eta_t}-\frac{\delta_{g,t}}{|\mathcal{S}|\eta_{t-1}}\right]\le&\frac{1}{|\mathcal{S}|}\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}-\frac{\gamma_{u,t+1}|\mathcal{B}|}{|\mathcal{S}|\eta_t}\right)\mathbb{E}\left[\delta_{g,t}\right]+\frac{8|\mathcal{S}|C_g^2}{|\mathcal{B}|\eta_t}\mathbb{E}\left[\left\|\mathbf{w}_{t+1}-\mathbf{w}_t\right\|^2\right]\\&+\frac{2|\mathcal{B}|\gamma_{u,t+1}^2\sigma^2}{|\mathcal{S}|\eta_t}\end{aligned} \tag{57}$$

$$\begin{aligned}\mathbb{E}\left[\frac{\delta_{L\lambda\lambda,t+1}}{N\eta_t}-\frac{\delta_{L\lambda\lambda,t}}{N\eta_{t-1}}\right]\le&\frac{1}{N}\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}-\frac{\gamma_{s,t+1}|\mathcal{B}|}{N\eta_t}\right)\mathbb{E}\left[\delta_{L\lambda\lambda,t}\right]\\&+\frac{16NL_{L\lambda\lambda}^2}{|\mathcal{B}|\eta_t}\mathbb{E}\left[\|\mathbf{w}_{t+1}-\mathbf{w}_t\|^2+\frac{1}{N}\left\|\lambda^{t+1}-\lambda^t\right\|^2\right]+\frac{2|\mathcal{B}|\gamma_{s,t+1}^2\sigma^2}{N\eta_t}\end{aligned} \tag{58}$$

$$\mathbb{E}\left[\frac{\delta_{m,t+1}}{\eta_t}-\frac{\delta_{m,t}}{\eta_{t-1}}\right]\leq\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}-\frac{\gamma_{m,t+1}}{\eta_t}\right)\mathbb{E}\left[\delta_{m,t}\right]+\frac{2\gamma_{m,t+1}^2\sigma^2}{\eta_t}+\frac{2C_3}{|S|\eta_t}\mathbb{E}\left[\left\|\mathbf{u}^{t+1}-\mathbf{u}^t\right\|^2\right]$$
$$+\frac{2C_4}{\eta_t}\mathbb{E}\left[\left\|\mathbf{w}_{t+1}-\mathbf{w}_t\right\|^2\right]+\frac{2C_5}{N\eta_t}\mathbb{E}\left[\left\|\lambda^{t+1}-\lambda^t\right\|^2\right]+\frac{2C_6}{N\eta_t}\mathbb{E}\left[\left\|\mathbf{s}^{t+1}-\mathbf{s}^t\right\|^2\right]. \tag{59}$$

To control the $\left\|\mathbf{u}^{t+1}-\mathbf{u}^t\right\|^2$ and $\left\|\mathbf{s}^{t+1}-\mathbf{s}^t\right\|^2$ terms in (59), i.e., the differences in the MSVR estimators $\mathbf{u}$ and $\mathbf{s}$ across iterations, we apply Lemma 17 and obtain the following

$$\frac{2C_3}{|\mathcal{S}|\eta_t}\mathbb{E}\left[\left\|\mathbf{u}^{t+1}-\mathbf{u}^t\right\|^2\right]+\frac{2C_6}{N\eta_t}\mathbb{E}\left[\left\|\mathbf{s}^{t+1}-\mathbf{s}^t\right\|^2\right]$$
$$\leq\frac{8|\mathcal{B}|C_3\gamma_{u,t+1}^2}{|\mathcal{S}|^2\eta_t}\mathbb{E}\left[\delta_{g,t}\right]+\frac{8|\mathcal{B}|C_6\gamma_{s,t+1}^2}{N^2\eta_t}\mathbb{E}\left[\delta_{L\lambda\lambda,t}\right]++\frac{2(2\gamma_{u,t+1}^2C_3+2\gamma_{s,t+1}^2C_6)|\mathcal{B}|}{N\eta_t}\sigma^2$$
$$+\left(\frac{18C_3|\mathcal{S}|C_g^2}{\eta_t|\mathcal{B}|}+\frac{36C_6NL_{L\lambda\lambda}^2}{\eta_t|\mathcal{B}|}\right)\mathbb{E}\left[\|\mathbf{w}_{t+1}-\mathbf{w}_t\|^2\right]+\frac{36C_6L_{L\lambda\lambda}^2}{\eta_t|\mathcal{B}|}\mathbb{E}\left[\left\|\lambda^{t+1}-\lambda^t\right\|^2\right].$$

Now, by summing all the recursions obtained above with (56), we can derive the following result. Note that the left sides of these recursions can be canceled after summing over t, while we combine like terms on the right side for convenience in subsequent proofs.

$$\mathbb{E}\left[\frac{\delta_{L\lambda,t+1}}{N\eta_t}-\frac{\delta_{L\lambda,t}}{N\eta_{t-1}}\right]+\mathbb{E}\left[\frac{C\alpha}{N}(\delta_{\lambda,t+1}-\delta_{\lambda,t})\right]+\mathbb{E}\left[\frac{\delta_{g,t+1}}{|\mathcal{S}|\eta_t}-\frac{\delta_{g,t}}{|\mathcal{S}|\eta_{t-1}}\right]+\mathbb{E}\left[\frac{\delta_{L\lambda\lambda,t+1}}{N\eta_t}-\frac{\delta_{L\lambda\lambda,t}}{N\eta_{t-1}}\right]$$
$$+\mathbb{E}\left[\frac{\delta_{m,t+1}}{\eta_t}-\frac{\delta_{m,t}}{\eta_{t-1}}\right]\leq\frac{2(\gamma_{z,t+1}^2+\gamma_{u,t+1}^2+\gamma_{s,t+1}^2+2\gamma_{u,t+1}^2C_3+2\gamma_{s,t+1}^2C_6)|\mathcal{B}|+2N\gamma_{m,t+1}^2}{N\eta_t}\sigma^2$$
$$+\frac{1}{|\mathcal{S}|}\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}-\frac{\gamma_{u,t+1}|\mathcal{B}|}{|\mathcal{S}|\eta_t}+\frac{8|\mathcal{B}|C_3\gamma_{u,t+1}^2}{|\mathcal{S}|\eta_t}\right)\mathbb{E}\left[\delta_{g,t}\right]+\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}-\frac{\gamma_{m,t+1}}{\eta_t}\right)\mathbb{E}\left[\delta_{m,t}\right]$$
$$+\frac{1}{N}\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}-\frac{\gamma_{s,t+1}|\mathcal{B}|}{N\eta_t}+\frac{8|\mathcal{B}|C_6\gamma_{s,t+1}^2}{N\eta_t}\right)\mathbb{E}\left[\delta_{L\lambda\lambda,t}\right]-\frac{C\alpha\tau\tau_t\mu_L|\mathcal{B}|}{4N^2}\mathbb{E}\left[\delta_{\lambda,t}\right]$$
$$+\left(\frac{16L_L^2+16L_{L\lambda\lambda}^2+36C_6L_{L\lambda\lambda}^2}{|\mathcal{B}|\eta_t}+\frac{2C_5}{N\eta_t}--\frac{3C\alpha\tau|\mathcal{B}|}{\tau_tN^2}\left(\frac{1}{\tau}-L_L\right)\right)\mathbb{E}\left[\left\|\lambda^{t+1}-\lambda^t\right\|^2\right]$$
$$+\left(\frac{8C\alpha NC_\lambda^2}{\tau\tau_t\mu_L|\mathcal{B}|}+\frac{(16L_L^2+16L_{L\lambda\lambda}^2+2C_4+36C_6L_{L\lambda\lambda}^2)N}{|\mathcal{B}|\eta_t}+\frac{(8C_g^2+18C_3C_g^2)|\mathcal{S}|}{|\mathcal{B}|\eta_t}\right)$$
$$\times\alpha^2\eta_t^2\mathbb{E}\left[\|\mathbf{m}_t\|^2\right],$$

where we use $\mathbf{w}_{t+1}=\mathbf{w}_t-\alpha\eta_t\mathbf{m}_t$. To simplify the inequality above, we denote $C_7=16L_L^2+16L_{L\lambda\lambda}^2+2C_4+36C_6L_{L\lambda\lambda}^2, C_8=8C_g^2+18C_3C_g^2, C_9=16L_L^2+16L_{L\lambda\lambda}^2+36C_6L_{L\lambda\lambda}^2$.

We then incorporate the results from Lemma 11 and 12 into the above inequality. Since our final goal is to prove the conclusion regarding the stationary point (which involves the $\nabla F(\mathbf{w}_t)$ term), Lemma 11 is needed. Additionally, Lemma 12 provides the stochastic gradient estimation error bound $\|\nabla F(\mathbf{w}_t)-\mathbf{m}_t\|^2$, so it also needs to be included. The final result is as follows:

$$\mathbb{E}\left[\frac{\delta_{L\lambda,t+1}}{N\eta_t}-\frac{\delta_{L\lambda,t}}{N\eta_{t-1}}\right]+\mathbb{E}\left[\frac{C\alpha}{N}(\delta_{\lambda,t+1}-\delta_{\lambda,t})\right]+\mathbb{E}\left[\frac{\delta_{g,t+1}}{|\mathcal{S}|\eta_t}-\frac{\delta_{g,t}}{|\mathcal{S}|\eta_{t-1}}\right]+\mathbb{E}\left[\frac{\delta_{L\lambda\lambda,t+1}}{N\eta_t}-\frac{\delta_{L\lambda\lambda,t}}{N\eta_{t-1}}\right]$$
$$+\mathbb{E}\left[\frac{\delta_{m,t+1}}{\eta_t}-\frac{\delta_{m,t}}{\eta_{t-1}}\right]+\frac{\alpha\eta_t}{2}\|\nabla F(\mathbf{w}_t)\|^2\leq F(\mathbf{w}_t)-F(\mathbf{w}_{t+1})+\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}-\frac{\gamma_{m,t+1}}{\eta_t}\right)\mathbb{E}\left[\delta_{m,t}\right]$$
$$+\frac{2(\gamma_{z,t+1}^2+\gamma_{u,t+1}^2+\gamma_{s,t+1}^2+2\gamma_{u,t+1}^2C_3+2\gamma_{s,t+1}^2C_6)|\mathcal{B}|+2N\gamma_{m,t+1}^2}{N\eta_t}\sigma^2+\frac{2C_0\alpha\eta_t}{N}\mathbb{E}\left[\delta_{\lambda,t}\right]$$

$$+\frac{1}{|\mathcal{S}|}\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}-\frac{\gamma_{u,t+1}|\mathcal{B}|}{|\mathcal{S}|\eta_t}+\frac{8|\mathcal{B}|C_3\gamma_{u,t+1}^2}{|\mathcal{S}|\eta_t}\right)\mathbb{E}\left[\delta_{g,t}\right]+\frac{2C_1\alpha\eta_t}{|\mathcal{S}|}\mathbb{E}\left[\delta_{g,t}\right]+\alpha\eta_t\mathbb{E}\left[\delta_{m,t}\right]$$
$$+\frac{1}{N}\left(\frac{1}{\eta_t}-\frac{1}{\eta_{t-1}}-\frac{\gamma_{s,t+1}|\mathcal{B}|}{N\eta_t}+\frac{8|\mathcal{B}|C_6\gamma_{s,t+1}^2}{N\eta_t}\right)\mathbb{E}\left[\delta_{L\lambda\lambda,t}\right]+\frac{2C_2\alpha\eta_t}{N}\mathbb{E}\left[\delta_{L\lambda\lambda,t}\right]$$
$$+\left(\frac{C_9}{|\mathcal{B}|\eta_t}+\frac{2C_5}{N\eta_t}-\frac{3C\alpha\tau|\mathcal{B}|}{\tau_t N^2}\left(\frac{1}{\tau}-L_L\right)\right)\mathbb{E}\left[\left\|\lambda^{t+1}-\lambda^t\right\|^2\right]-\frac{C\alpha\tau\tau_t\mu_L|\mathcal{B}|}{4N^2}\mathbb{E}\left[\delta_{\lambda,t}\right]$$
$$+\left(\frac{8C\alpha NC_\lambda^2}{\tau\tau_t\mu_L|\mathcal{B}|}+\frac{C_7N}{|\mathcal{B}|\eta_t}+\frac{C_8|\mathcal{S}|}{|B|\eta_t}\right)\alpha^2\eta_t^2\mathbb{E}\left[\|\mathbf{m}_t\|^2\right]-\frac{\alpha\eta_t}{4}\mathbb{E}\left[\|\mathbf{m}_t\|^2\right]. \tag{60}$$

Although the RHS of (60) contains many tracking error terms, we can actually eliminate these terms by appropriately setting the parameters to make their coefficients negative. Specifically,

- For $\mathbb{E}[\delta_{\lambda,t}]$, we can set $C \geq \frac{8C_0N}{\tau\mu_L|\mathcal{B}|}$ to make $\frac{2C_0\alpha\eta_t}{N}-\frac{C\alpha\tau\tau_t\mu_L|\mathcal{B}|}{4N^2}\leq 0$.
- For $\mathbb{E}[\delta_{g,t}]$, with $\gamma_{u,t+1}\leq\frac{1}{16C_3}$, we have $\frac{8|\mathcal{B}|C_3\gamma_{u,t+1}^2}{|\mathcal{S}|\eta_t}\leq\frac{\gamma_{u,t+1}|\mathcal{B}|}{2|\mathcal{S}|\eta_t}$. Thus, we can set $\gamma_{u,t+1}=\left(\frac{2|\mathcal{S}|}{7|\mathcal{B}|L_Fc^3}+\frac{4C_1\alpha|\mathcal{S}|}{|\mathcal{B}|}\right)\eta_t^2$, and $\gamma_{u,t+1}\leq\min\{\frac{1}{2},\frac{1}{16C_3}\}$ can be achieved by setting $c_0\geq\max\left\{\left(\frac{8|\mathcal{S}|}{7L_F|\mathcal{B}|c}\right)^{3/2},\left(\frac{16C_1\alpha|\mathcal{S}|c^2}{|\mathcal{B}|}\right)^{3/2},\left(\frac{64|\mathcal{S}|C_3}{7L_F|\mathcal{B}|c}\right)^{3/2},\left(\frac{128C_1|\mathcal{S}|C_3\alpha c^2}{|\mathcal{B}|}\right)^{3/2}\right\}$.
- For $\mathbb{E}[\delta_{L\lambda\lambda,t}]$, with $\gamma_{s,t+1}\leq\frac{1}{16C_6}$, we have $\frac{8|\mathcal{B}|C_6\gamma_{s,t+1}^2}{N\eta_t}\leq\frac{\gamma_{s,t+1}|\mathcal{B}|}{2N\eta_t}$. Thus, we can set $\gamma_{s,t+1}=\left(\frac{2N}{7|\mathcal{B}|L_Fc^3}+\frac{4C_2\alpha N}{|\mathcal{B}|}\right)\eta_t^2$, and $\gamma_{s,t+1}\leq\min\{\frac{1}{2},\frac{1}{16C_6}\}$ can be achieved by setting $c_0\geq\max\left\{\left(\frac{8N}{7L_F|\mathcal{B}|c}\right)^{3/2},\left(\frac{16C_2\alpha Nc^2}{|\mathcal{B}|}\right)^{3/2},\left(\frac{64NC_6}{7L_F|\mathcal{B}|c}\right)^{3/2},\left(\frac{128C_2NC_6\alpha c^2}{|\mathcal{B}|}\right)^{3/2}\right\}$.
- For $\mathbb{E}[\delta_{m,t}]$, we can set $\gamma_{m,t+1}=\left(\frac{1}{7L_Fc^3}+\alpha\right)\eta_t^2$ and $\gamma_{m,t+1}\leq\frac{1}{2}$ can be achieved by setting $c_0\geq\max\left\{\left(\frac{4}{7L_Fc}\right)^{3/2},\left(4\alpha c^2\right)^{3/2}\right\}$.
- For $\mathbb{E}[\|\lambda^{t+1}-\lambda^t\|^2]$, with $L_L\leq\frac{1}{2\tau}$, we have $\frac{C_9}{|\mathcal{B}|\eta_t}+\frac{2C_5}{N\eta_t}-\frac{3C\alpha\tau|\mathcal{B}|}{\tau_tN^2}\left(\frac{1}{\tau}-L_L\right)\leq\frac{C_9}{|\mathcal{B}|\eta_t}+\frac{2C_5}{N\eta_t}-\frac{3C\alpha\tau|\mathcal{B}|}{2\tau_tN^2}$. By setting $C\geq\frac{2(C_9+2C_5)N^2}{3\alpha|\mathcal{B}|^2}$, we have $\frac{C_9}{|\mathcal{B}|\eta_t}+\frac{2C_5}{N\eta_t}-\frac{3C\alpha\tau|\mathcal{B}|}{2\tau_tN^2}\leq 0$.
- For $\mathbb{E}[\|\mathbf{m}_t\|^2]$, with $\alpha\leq\min\left\{\frac{|\mathcal{B}|}{12C_7N},\frac{|\mathcal{B}|}{12C_8|\mathcal{S}|},\frac{1}{C_\lambda}\sqrt{\frac{\tau\mu_L|\mathcal{B}|}{96CN}}\right\}$, we further have $\left(\frac{8C\alpha NC_\lambda^2}{\tau\tau_t\mu_L|\mathcal{B}|}+\frac{C_7N}{|\mathcal{B}|\eta_t}+\frac{C_8|\mathcal{S}|}{|\mathcal{B}|\eta_t}\right)\alpha^2\eta_t^2-\frac{\alpha\eta_t}{4}\leq 0$.

With all these considerations in hand, we obtain

$$\mathbb{E}\left[\frac{\delta_{L\lambda,t+1}}{n\eta_t}-\frac{\delta_{L\lambda,t}}{n\eta_{t-1}}\right]+\mathbb{E}\left[\frac{C\alpha}{n}(\delta_{\lambda,t+1}-\delta_{\lambda,t})\right]+\mathbb{E}\left[\frac{\delta_{g,t+1}}{n\eta_t}-\frac{\delta_{g,t}}{n\eta_{t-1}}\right]$$
$$+\mathbb{E}\left[\frac{\delta_{L\lambda\lambda,t+1}}{n\eta_t}-\frac{\delta_{L\lambda\lambda,t}}{n\eta_{t-1}}\right]$$
$$+\mathbb{E}\left[\frac{\delta_{m,t+1}}{\eta_t}-\frac{\delta_{m,t}}{\eta_{t-1}}\right]+\frac{\alpha\eta_t}{2}\|\nabla F(\mathbf{w}_t)\|^2\leq F(\mathbf{w}_t)-F(\mathbf{w}_{t+1})+\mathcal{O}(1)\eta_t^3.$$

As we discussed earlier, by summing the above equation over $t=1,2,\cdots,T$, the left-hand side terms can be offset and canceled. Thus, we obtain:

$$\mathbb{E}\left[\sum_{t=1}^{T}\frac{\alpha\eta_t}{2}\|\nabla F(\mathbf{w}_t)\|^2\right]\leq F(\mathbf{w}_1)-F(\mathbf{w}_{T+1})+\frac{\mathbb{E}\left[\delta_{L\lambda,1}+\delta_{g,1}+\delta_{L\lambda\lambda,1}+n\delta_{m,1}\right]}{n\eta_1}$$

$$+\frac{C\alpha}{n}\mathbb{E}\left[\delta_{\lambda,1}\right]+\mathcal{O}(\log(T+1)).$$

Denote $M = F(\mathbf{w}_1) - F(\mathbf{w}_{T+1}) + \frac{1}{n\eta_1}\mathbb{E}\left[\delta_{L\lambda,1}+\delta_{g,1}+\delta_{L\lambda\lambda,1}+n\delta_{m,1}\right] + \frac{C\alpha}{n}\mathbb{E}\left[\delta_{\lambda,1}\right] + \mathcal{O}(\log(T+1))$, then we have

$$\mathbb{E}\left[\sum_{t=1}^{T}\frac{\alpha}{2T}\left\|\nabla F(\mathbf{w}_t)\right\|^2\right] \leq \frac{M}{\eta_T T}.$$

Note that $\eta_T = \frac{c}{(c_0+T)^{1/3}}$, so $\frac{M}{\eta_T T} = \frac{M}{T}\frac{(c_0+T)^{1/3}}{c} \leq \frac{Mc_0^{1/3}}{Tc} + \frac{MT^{1/3}}{Tc} \sim \mathcal{O}\left(\frac{1}{T^{2/3}}\right)$, where the inequality is due to $(a+b)^{1/3} \leq a^{1/3} + b^{1/3}$, thus we have

$$\mathbb{E}\left[\sum_{t=1}^{T}\frac{1}{T}\left\|\nabla F(\mathbf{w}_t)\right\|^2\right] \leq \mathcal{O}\left(\frac{1}{T^{2/3}}\right).$$

□

Proof of Lemma 12 In this lemma, we will derive the gradient estimation error bound $\|\nabla F(\mathbf{w}_t) - \mathbf{m}_t\|^2$ for the objective function $F(\mathbf{w})$. First, we introduce an intermediate term $\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} G_i(\mathbf{w}_t)$ to facilitate the analysis

$$\begin{aligned}
\left\|\nabla F(\mathbf{w}_t) - \mathbf{m}_t\right\|^2 &= \left\|\nabla F(\mathbf{w}_t) - \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} G_i(\mathbf{w}_t) + \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} G_i(\mathbf{w}_t) - \mathbf{m}_t\right\|^2 \\
&\leq 2\left\|\nabla F(\mathbf{w}_t) - \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} G_i(\mathbf{w}_t)\right\|^2 + 2\left\|\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} G_i(\mathbf{w}_t) - \mathbf{m}_t\right\|^2 \\
&= 2\left\|\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\nabla F_i(\mathbf{w}_t) - \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} G_i(\mathbf{w}_t)\right\|^2 + 2\left\|\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} G_i(\mathbf{w}_t) - \mathbf{m}_t\right\|^2 \\
&\leq 2\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\left\|\nabla F_i(\mathbf{w}_t) - G_i(\mathbf{w}_t)\right\|^2 + 2\left\|\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}} G_i(\mathbf{w}_t) - \mathbf{m}_t\right\|^2. \qquad (61)
\end{aligned}$$

In order to bound $\|\nabla F_i(\mathbf{w}_t) - G_i(\mathbf{w}_t)\|^2$, we introduce $\nabla F_i(\mathbf{w}_t, \lambda_q^t)$, which uses the estimated lower-level solution λ_q^t instead of $\lambda_q(\mathbf{w}_t)$ in $\nabla F_i(\mathbf{w}_t)$. We can first establish the bound for $\left\|\nabla F_i(\mathbf{w}_t) - \nabla F_i\left(\mathbf{w}_t, \lambda_q^t\right)\right\|^2$ as follows

$$\begin{aligned}
&\left\|\nabla F_i(\mathbf{w}_t) - \nabla F_i\left(\mathbf{w}_t, \lambda_q^t\right)\right\|^2 \\
&\leq \left\|\left[\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t, \lambda_q(\mathbf{w}_t)) - \nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}, \lambda_q(\mathbf{w}_t))[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))]^{-1}\nabla_\lambda\psi_i(\mathbf{w}, \lambda_q(\mathbf{w}_t))\right] f_i(g_i(\mathbf{w}_t))\right. \\
&\quad +\psi_i(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))\nabla g_i(\mathbf{w}_t)\nabla f_i(g_i(\mathbf{w}_t)) - \psi_i(\mathbf{w}_t, \lambda_q^t)\nabla g_i(\mathbf{w}_t)\nabla f_i(g_i(\mathbf{w}_t)) \\
&\quad \left.-\left[\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t, \lambda_q^t) - \nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}, \lambda_q^t)[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t, \lambda_q^t)]^{-1}\nabla_\lambda\psi_i(\mathbf{w}, \lambda_q^t)\right] f_i(g_i(\mathbf{w}_t))\right\|^2 \\
&\leq 3\left\|\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t, \lambda_q(\mathbf{w}_t)) f_i(g_i(\mathbf{w}_t)) - \nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t, \lambda_q^t) f_i(g_i(\mathbf{w}_t))\right\|^2 \\
&\quad + 3\left\|\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}, \lambda_q(\mathbf{w}_t))[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t))]^{-1}\nabla_\lambda\psi_i(\mathbf{w}, \lambda_q(\mathbf{w}_t)) f_i(g_i(\mathbf{w}_t))\right. \\
&\quad \left.-\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}, \lambda_q^t)[\nabla^2_{\lambda\lambda}L_q(\mathbf{w}_t, \lambda_q^t)]^{-1}\nabla_\lambda\psi_i(\mathbf{w}, \lambda_q^t) f_i(g_i(\mathbf{w}_t))\right\|^2
\end{aligned}$$

$$
\begin{aligned}
&+3\left\|\psi_i(\mathbf{w}_t,\lambda_q(\mathbf{w}_t))\nabla g_i(\mathbf{w}_t)\nabla f_i(g_i(\mathbf{w}_t))-\psi_i(\mathbf{w}_t,\lambda_q^t)\nabla g_i(\mathbf{w}_t)\nabla f_i(g_i(\mathbf{w}_t))\right\|^2\\
&\le 3L_\psi^2B_f^2\left\|\lambda_q(\mathbf{w}_t)-\lambda_q^t\right\|^2+9\frac{L_{\mathbf{w}\lambda}^2C_\psi^2B_f^2}{\gamma^2}\left\|\lambda_q(\mathbf{w}_t)-\lambda_q^t\right\|^2+9\frac{C_{L\mathbf{w}\lambda}^2L_{L\lambda\lambda}^2C_\psi^2B_f^2}{\gamma^4}\left\|\lambda_q(\mathbf{w}_t)-\lambda_q^t\right\|^2\\
&+9\frac{C_{L\mathbf{w}\lambda}^2L_\psi^2B_f^2}{\gamma^2}\left\|\lambda_q(\mathbf{w}_t)-\lambda_q^t\right\|^2+3C_g^2C_f^2C_\psi^2\left\|\lambda_q(\mathbf{w}_t)-\lambda_q^t\right\|^2\\
&=\underbrace{\left(3L_\psi^2B_f^2+9\frac{L_{\mathbf{w}\lambda}^2C_\psi^2B_f^2}{\gamma^2}+9\frac{C_{L\mathbf{w}\lambda}^2L_{L\lambda\lambda}^2C_\psi^2B_f^2}{\gamma^4}+9\frac{C_{L\mathbf{w}\lambda}^2L_\psi^2B_f^2}{\gamma^2}+3C_g^2C_f^2C_\psi^2\right)}_{C_0}\left\|\lambda_q(\mathbf{w}_t)-\lambda_q^t\right\|^2,
\end{aligned}
\tag{62}
$$

where we first expand $\left\|\nabla F_i\left(\mathbf{w}_t\right)-\nabla F_i\left(\mathbf{w}_t,\lambda_q^t\right)\right\|^2$ by the definitions, then decompose it into several terms according to inequality $\|\mathbf{x}_1+\cdots+\mathbf{x}_n\|^2\le n\|\mathbf{x}_1\|^2+\cdots+n\|\mathbf{x}_n\|^2$, and finally use the Lipschitz continuity or smoothness properties of these functions assumed in Assumption 1 to derive their bounds.

Similarly, we can establish the bound for $\left\|\nabla F_i\left(\mathbf{w}_t,\lambda_q^t\right)-G_i\left(\mathbf{w}_t\right)\right\|^2$

$$
\begin{aligned}
&\left\|\nabla F_i\left(\mathbf{w}_t,\lambda_q^t\right)-G_i\left(\mathbf{w}_t\right)\right\|^2\\
&=\left\|\left[\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t,\lambda_q^t)-\nabla_{\mathbf{w}\lambda}^2L_q(\mathbf{w},\lambda_q^t)[\nabla_{\lambda\lambda}^2L_q(\mathbf{w}_t,\lambda_q^t)]^{-1}\nabla_\lambda\psi_i(\mathbf{w},\lambda_q^t)\right]f_i(g_i(\mathbf{w}_t))\right.\\
&\quad+\psi_i(\mathbf{w}_t,\lambda_q^t)\nabla g_i(\mathbf{w}_t)\nabla f_i(g_i(\mathbf{w}_t))-\psi_i(\mathbf{w}_t,\lambda_q^t)\nabla g_i(\mathbf{w}_t)\nabla f_i(\mathbf{u}_i^t))\\
&\quad\left.-\left[\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t,\lambda_q^t)-\nabla_{\mathbf{w}\lambda}^2L_q(\mathbf{w},\lambda_q^t)[\mathbf{s}_q^t]^{-1}\nabla_\lambda\psi_i(\mathbf{w},\lambda_q^t)\right]f_i(\mathbf{u}_i^t)\right\|^2\\
&\le 3\left\|\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t,\lambda_q^t)f_i(g_i(\mathbf{w}_t))-\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t,\lambda_q^t)f_i(\mathbf{u}_i^t))\right\|^2\\
&\quad+3\left\|\psi_i(\mathbf{w}_t,\lambda_q^t)\nabla g_i(\mathbf{w}_t)\nabla f_i(g_i(\mathbf{w}_t))-\psi_i(\mathbf{w}_t,\lambda_q^t)\nabla g_i(\mathbf{w}_t)\nabla f_i(\mathbf{u}_i^t))\right\|^2\\
&\quad+3\left\|\nabla_{\mathbf{w}\lambda}^2L_q(\mathbf{w},\lambda_q^t)[\nabla_{\lambda\lambda}^2L_q(\mathbf{w}_t,\lambda_q^t)]^{-1}\nabla_\lambda\psi_i(\mathbf{w},\lambda_q^t)f_i(g_i(\mathbf{w}_t))\right.\\
&\quad\left.-\nabla_{\mathbf{w}\lambda}^2L_q(\mathbf{w},\lambda_q^t)[\mathbf{s}_q^t]^{-1}\nabla_\lambda\psi_i(\mathbf{w},\lambda_q^t)f_i(\mathbf{u}_i^t)\right\|^2\\
&\le\underbrace{\left(3C_\psi^2C_f^2+3B_\psi^2C_g^2L_f^2+6\frac{C_{L\mathbf{w}\lambda}^2C_\psi^2C_f^2}{\gamma^2}\right)}_{C_1}\left\|\mathbf{u}_i^t-g_i(\mathbf{w}_t)\right\|^2+\underbrace{6\frac{C_{L\mathbf{w}\lambda}^2C_\psi^2B_f^2}{\gamma^4}}_{C_2}\left\|\mathbf{s}_q^t-\nabla_{\lambda\lambda}^2L_q(\mathbf{w}_t,\lambda_q^t)\right\|^2.
\end{aligned}
\tag{63}
$$

Thus, by combining (62) and (63), we have the bound for $\|\nabla F_i\left(\mathbf{w}_t\right)-G_i\left(\mathbf{w}_t\right)\|^2$

$$
\begin{aligned}
\|\nabla F_i\left(\mathbf{w}_t\right)-G_i\left(\mathbf{w}_t\right)\|^2&\le 2\left\|\nabla F_i\left(\mathbf{w}_t\right)-\nabla F_i\left(\mathbf{w}_t,\lambda_q^t\right)\right\|^2+2\left\|\nabla F_i\left(\mathbf{w}_t,\lambda_q^t\right)-G_i\left(\mathbf{w}_t\right)\right\|^2\\
&\le 2C_0\left\|\lambda_q(\mathbf{w}_t)-\lambda_q^t\right\|^2+2C_1\left\|\mathbf{u}_i^t-g_i(\mathbf{w}_t)\right\|^2+2C_2\left\|\mathbf{s}_q^t-\nabla_{\lambda\lambda}^2L_q(\mathbf{w}_t,\lambda_q^t)\right\|^2.
\end{aligned}
\tag{64}
$$

As a result, combining (61) and (64), we derive the following inequality.

$$
\begin{aligned}
&\|\nabla F\left(\mathbf{w}_t\right)-\mathbf{m}_t\|^2\\
&\le 2\left\|\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i\left(\mathbf{w}_t\right)-\mathbf{m}_t\right\|^2+\frac{4C_0}{N}\left\|\lambda(\mathbf{w}_t)-\lambda^t\right\|^2+\frac{4C_1}{|\mathcal{S}|}\left\|\mathbf{u}^t-g(\mathbf{w}_t)\right\|^2
\end{aligned}
$$

$$+\frac{4C_2}{N}\left\|\mathbf{s}^t-\nabla^2_{\lambda\lambda}L(\mathbf{w}_t,\lambda^t)\right\|^2.$$

One can observe that the stochastic gradient estimation error $\|\nabla F(\mathbf{w}_t)-\mathbf{m}_t\|^2$ is actually composed of the STORM estimation error $\left\|\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_t)\right\|^2$, the lower-level solutions estimation error $\left\|\lambda(\mathbf{w}_t)-\lambda^t\right\|^2$, and two tracking errors ($\left\|\mathbf{u}^t-g(\mathbf{w}_t)\right\|^2$ and $\left\|\mathbf{s}^t-\nabla^2_{\lambda\lambda}L(\mathbf{w}_t,\lambda^t)\right\|^2$). □

Proof of Lemma 13 We now begin to analyze the error bound between the stochastic gradient estimator $\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_t)$ and its STORM estimator $\mathbf{m}_t$. According to the update rule of $\mathbf{m}_t=(1-\gamma_{m,t})\left(\mathbf{m}_{t-1}-\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}G_i(\mathbf{w}_{t-1})\right)+\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}G_i(\mathbf{w}_t)$, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_t)-\mathbf{m}_t\right\|^2\right]=\mathbb{E}\left[\left\|\mathbf{m}_t-\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_t)\right\|^2\right]\\
&=\mathbb{E}\left[\left\|(1-\gamma_{m,t})\left(\mathbf{m}_{t-1}-\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}G_i(\mathbf{w}_{t-1})\right)+\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}G_i(\mathbf{w}_t)-\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_t)\right\|^2\right]\\
&=\mathbb{E}\left[\left\|(1-\gamma_{m,t})\left(\mathbf{m}_{t-1}-\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_{t-1})\right)+\gamma_{m,t}\left(\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}G_i(\mathbf{w}_t)-\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_t)\right)\right.\right.\\
&\quad\left.\left.+(1-\gamma_{m,t})\left(\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}G_i(\mathbf{w}_t)-\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}G_i(\mathbf{w}_{t-1})-\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_t)+\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_{t-1})\right)\right\|^2\right].
\end{aligned}\tag{65}$$

Due to the fact that $\mathbb{E}\left[\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}G_i(\mathbf{w}_t)-\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_t)\right]\le\sigma^2$ and the expectation over the last two terms equals to zero, we obtain

$$(65)\le\mathbb{E}\left[(1-\gamma_{m,t})^2\left\|\mathbf{m}_{t-1}-\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_{t-1})\right\|^2+2\gamma_{mt}^2\sigma^2+2(1-\gamma_{m,t})^2\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\|G_i(\mathbf{w}_t)-G_i(\mathbf{w}_{t-1})\|^2\right].\tag{66}$$

Next, we aim to bound $\|G_i(\mathbf{w}_t)-G_i(\mathbf{w}_{t-1})\|^2$. We first expand the error term according to the definition of $G_i(\mathbf{w}_t)$, then proceed with decomposition by inequality $\|\mathbf{x}_1+\cdots+\mathbf{x}_n\|^2\le n\|\mathbf{x}_1\|^2+\cdots+n\|\mathbf{x}_n\|^2$:

$$\begin{aligned}
\left\|G_i(\mathbf{w}_t)-G_i(\mathbf{w}_{t-1})\right\|^2&=\left\|\left[\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t,\lambda_q^t)-\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_t,\lambda_q^t;\mathcal{B}_q)[\mathbf{s}_q^t]^{-1}\nabla_\lambda\psi_i(\mathbf{w}_t,\lambda_q^t)\right]f_i(\mathbf{u}_i^t)\right.\\
&\quad-\left[\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_{t-1},\lambda_q^{t-1})-\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_{t-1},\lambda_q^{t-1};\mathcal{B}_q)[\mathbf{s}_q^{t-1}]^{-1}\nabla_\lambda\psi_i(\mathbf{w}_{t-1},\lambda_q^{t-1})\right]f_i(\mathbf{u}_i^{t-1})\\
&\quad\left.+\psi_i(\mathbf{w}_t,\lambda_q^t)\nabla g_i(\mathbf{w}_t;\mathcal{B}_q)\nabla f_i(\mathbf{u}_i^t)-\psi_i(\mathbf{w}_{t-1},\lambda_q^{t-1})\nabla g_i(\mathbf{w}_{t-1};\mathcal{B}_q)\nabla f_i(\mathbf{u}_i^{t-1})\right\|^2\\
&\le 3\left\|\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_t,\lambda_q^t)f_i(\mathbf{u}_i^t)-\nabla_{\mathbf{w}}\psi_i(\mathbf{w}_{t-1},\lambda_q^{t-1})f_i(\mathbf{u}_i^{t-1})\right\|^2\\
&\quad+3\left\|\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_t,\lambda_q^t;\mathcal{B}_q)[\mathbf{s}_q^t]^{-1}\nabla_\lambda\psi_i(\mathbf{w}_t,\lambda_q^t)f_i(\mathbf{u}_i^t)\right.\\
&\quad\left.-\nabla^2_{\mathbf{w}\lambda}L_q(\mathbf{w}_{t-1},\lambda_q^{t-1};\mathcal{B}_q)[\mathbf{s}_q^{t-1}]^{-1}\nabla_\lambda\psi_i(\mathbf{w}_{t-1},\lambda_q^{t-1})f_i(\mathbf{u}_i^{t-1})\right\|^2\\
&\quad+3\left\|\psi_i(\mathbf{w}_t,\lambda_q^t)\nabla g_i(\mathbf{w}_t;\mathcal{B}_q)\nabla f_i(\mathbf{u}_i^t)-\psi_i(\mathbf{w}_{t-1},\lambda_q^{t-1})\nabla g_i(\mathbf{w}_{t-1};\mathcal{B}_q)\nabla f_i(\mathbf{u}_i^{t-1})\right\|^2
\end{aligned}\tag{67}$$

Afterwards, we continue to use a similar method for decomposition. For the resulting terms, we use the Lipschitz continuity or smoothness properties of the relevant functions assumed in Assumption 1 to establish bounds. Finally, we combine like terms and achieve

$$
\begin{aligned}
&\left\|G_i(\mathbf{w}_t)-G_i(\mathbf{w}_{t-1})\right\|^2 \le 6C_\psi^2C_f^2\left\|\mathbf{u}_i^t-\mathbf{u}_i^{t-1}\right\|^2+12B_f^2L_\psi^2\left\|\mathbf{w}_t-\mathbf{w}_{t-1}\right\|^2\\
&\quad+12B_f^2L_\psi^2\left\|\lambda_q^t-\lambda_q^{t-1}\right\|^2\\
&\quad+12\frac{C_\psi^2B_f^2}{\gamma^2}L_{L\mathbf{w}\lambda}^2(2\left\|\mathbf{w}_t-\mathbf{w}_{t-1}\right\|^2+2\left\|\lambda_q^t-\lambda_q^{t-1}\right\|^2)+12\frac{C_{L\mathbf{w}\lambda}^2C_\psi^2B_f^2}{\gamma^4}\left\|\mathbf{s}_q^t-\mathbf{s}_q^{t-1}\right\|^2\\
&\quad+12\frac{L_\psi^2B_f^2}{\gamma^2}C_{L\mathbf{w}\lambda}^2(2\left\|\mathbf{w}_t-\mathbf{w}_{t-1}\right\|^2+2\left\|\lambda_q^t-\lambda_q^{t-1}\right\|^2)+12\frac{C_{L\mathbf{w}\lambda}^2C_\psi^2C_f^2}{\gamma^2}\left\|\mathbf{u}_i^t-\mathbf{u}_i^{t-1}\right\|^2\\
&\quad+9C_g^2C_f^2C_\psi^2(2\left\|\mathbf{w}_t-\mathbf{w}_{t-1}\right\|^2+2\left\|\lambda_q^t-\lambda_q^{t-1}\right\|^2)\\
&\quad+9B_\psi^2C_f^2L_g^2\left\|\mathbf{w}_t-\mathbf{w}_{t-1}\right\|^2+9B_\psi^2C_g^2L_f^2\left\|\mathbf{u}_i^t-\mathbf{u}_i^{t-1}\right\|^2\\
&\le\underbrace{\left(6C_\psi^2C_f^2+12\frac{C_{L\mathbf{w}\lambda}^2C_\psi^2C_f^2}{\gamma^2}+9B_\psi^2C_g^2L_f^2\right)}_{C_3}\left\|\mathbf{u}_i^t-\mathbf{u}_i^{t-1}\right\|^2\\
&\quad+\underbrace{\left(12B_f^2L_\psi^2+24\frac{C_\psi^2B_f^2}{\gamma^2}L_{L\mathbf{w}\lambda}^2+24\frac{L_\psi^2B_f^2}{\gamma^2}C_{L\mathbf{w}\lambda}^2+18C_g^2C_f^2C_\psi^2+9B_\psi^2C_f^2L_g^2\right)}_{C_4}\left\|\mathbf{w}_t-\mathbf{w}_{t-1}\right\|^2\\
&\quad+\underbrace{\left(12B_f^2L_\psi^2+24\frac{C_\psi^2B_f^2}{\gamma^2}L_{L\mathbf{w}\lambda}^2+24\frac{L_\psi^2B_f^2}{\gamma^2}C_{L\mathbf{w}\lambda}^2+18C_g^2C_f^2C_\psi^2\right)}_{C_5}\left\|\lambda_q^t-\lambda_q^{t-1}\right\|^2\\
&\quad+\underbrace{12\frac{C_{L\mathbf{w}\lambda}^2C_\psi^2B_f^2}{\gamma^4}}_{C_6}\left\|\mathbf{s}_q^t-\mathbf{s}_q^{t-1}\right\|^2. \qquad (68)
\end{aligned}
$$

By combining (66) and (68), we obtain the error bound for the STORM estimator $\mathbf{m}_t$, which has the following recursive form:

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\mathbf{m}_t-\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_t)\right\|^2\right]\\
&\le\mathbb{E}\left[(1-\gamma_{m,t})^2\left\|\mathbf{m}_{t-1}-\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_{t-1})\right\|^2+2\gamma_{m,t}^2\sigma^2\right.\\
&\quad\left.+2(1-\gamma_{m,t})^2\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\left\|G_i(\mathbf{w}_t)-G_i(\mathbf{w}_{t-1})\right\|^2\right]\\
&\le(1-\gamma_{m,t})\mathbb{E}\left[\left\|\mathbf{m}_{t-1}-\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_{t-1})\right\|^2\right]+2\gamma_{m,t}^2\sigma^2+\frac{2(1-\gamma_{m,t})^2C_3}{|\mathcal{S}|}\left\|\mathbf{u}^t-\mathbf{u}^{t-1}\right\|^2\\
&\quad+2(1-\gamma_{m,t})^2C_4\left\|\mathbf{w}_t-\mathbf{w}_{t-1}\right\|^2+\frac{2(1-\gamma_{m,t})^2C_5}{N}\left\|\lambda^t-\lambda^{t-1}\right\|^2+\frac{2(1-\gamma_{m,t})^2C_6}{N}\left\|\mathbf{s}^t-\mathbf{s}^{t-1}\right\|^2
\end{aligned}
$$

$$\leq (1-\gamma_{m,t})\mathbb{E}\left[\left\|\mathbf{m}_{t-1}-\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}G_i(\mathbf{w}_{t-1})\right\|^2\right]+2\gamma_{m,t}^2\sigma^2$$
$$+\frac{2C_3}{|\mathcal{S}|}\left\|\mathbf{u}^t-\mathbf{u}^{t-1}\right\|^2+2C_4\left\|\mathbf{w}_t-\mathbf{w}_{t-1}\right\|^2+\frac{2C_5}{N}\left\|\lambda^t-\lambda^{t-1}\right\|^2+\frac{2C_6}{N}\left\|\mathbf{s}^t-\mathbf{s}^{t-1}\right\|^2.$$

It can be observed that the RHS of this error bound mainly involves the differences between other estimators (including $\mathbf{u}$, $\mathbf{w}$, λ, and $\mathbf{s}$) at different iterations. □

Proof of Lemma 14 This lemma will establish the estimation error bound for the solutions λ for the lower-level problems, which is crucial for achieving parallel speed-up and the optimal convergence rate. First, we consider the stochastic gradient estimators $\mathbf{z}$ for the lower-level problems and the update rule for λ:

$$\mathbf{z}_q^t=\begin{cases}(1-\gamma_{z,t})\mathbf{z}_q^{t-1}+\gamma_{z,t}\nabla L_q(\lambda_q^t;\mathbf{w}_t;\mathcal{B}_q)\\ \qquad\quad+\beta_{z,t}(\nabla_\lambda L_q(\lambda_q^t;\mathbf{w}_t;\mathcal{B}_q)-\nabla_\lambda L_q(\lambda_q^{t-1};\mathbf{w}_{t-1};\mathcal{B}_q)), & \text{if } q\in\mathcal{B},\\ \mathbf{z}_q^{t-1}, & \text{o.w.}\end{cases}$$
$$\lambda_q^{t+1}=\begin{cases}\lambda_q^t-\tau\tau_t\mathbf{z}_q^t & \text{if } q\in\mathcal{B}\\ \lambda_q^t & \text{o.w.}\end{cases}.$$

Our proof strategy is to first establish the estimation error bound for the components updated in the t-th iteration, and then extend this to establish the estimation error bound for all components. For the convenience of the subsequent proof, we define these two intermediate variables: $\tilde{\lambda}_q^t=\lambda_q^t-\tau z_q^t$, $\bar{\lambda}_q^t=\lambda_q^t+\tau_t(\tilde{\lambda}_q^t-\lambda_q^t)$ for $q\in\mathcal{B}$. Note that

$$\left\|\bar{\lambda}_q^t-\lambda_q(\mathbf{w}_t)\right\|^2=\left\|\lambda_q^t+\tau_t(\tilde{\lambda}_q^t-\lambda_q^t)-\lambda_q(\mathbf{w}_t)\right\|^2$$
$$=\left\|\lambda_q^t-\lambda_q(\mathbf{w}_t)\right\|^2+\tau_t^2\left\|\tilde{\lambda}_q^t-\lambda_q^t\right\|^2+2\tau_t(\lambda_q^t-\lambda_q(\mathbf{w}_t))(\tilde{\lambda}_q^t-\lambda_q^t).$$

By rearranging the above equation, we obtain:

$$(\lambda_q^t-\lambda_q(\mathbf{w}_t))(\tilde{\lambda}_q^t-\lambda_q^t)=\frac{1}{2\tau_t}\left(\left\|\bar{\lambda}_q^t-\lambda_q(\mathbf{w}_t)\right\|^2-\left\|\lambda_q^t-\lambda_q(\mathbf{w}_t)\right\|^2-\tau_t^2\left\|\tilde{\lambda}_q^t-\lambda_q^t\right\|^2\right).\tag{69}$$

Next, we derive a tight bound by leveraging the strongly convexity and smoothness of L_q. By utilizing the strongly convexity property of $L_q(\mathbf{w}_t,\lambda)$, we can establish the following lower bound

$$L_q(\mathbf{w}_t,\lambda)\geq L_q(\mathbf{w}_t,\lambda_q^t)+\nabla_\lambda L_q(\mathbf{w}_t,\lambda_q^t)(\lambda-\lambda_q^t)+\frac{\mu_L}{2}\left\|\lambda-\lambda_q^t\right\|^2$$
$$=L_q(\mathbf{w}_t,\lambda_q^t)+\nabla_\lambda L_q(\mathbf{w}_t,\lambda_q^t)(\lambda-\tilde{\lambda}_q^t)+\nabla_\lambda L_q(\mathbf{w}_t,\lambda_q^t)(\tilde{\lambda}_q^t-\lambda_q^t)+\frac{\mu_L}{2}\left\|\lambda-\lambda_q^t\right\|^2$$
$$=L_q(\mathbf{w}_t,\lambda_q^t)+\mathbf{z}_q^t(\lambda-\tilde{\lambda}_q^t)+(\nabla_\lambda L_q(\mathbf{w}_t,\lambda_q^t)-\mathbf{z}_q^t)(\lambda-\tilde{\lambda}_q^t)+\nabla_\lambda L_q(\mathbf{w}_t,\lambda_q^t)(\tilde{\lambda}_q^t-\lambda_q^t)$$
$$+\frac{\mu_L}{2}\left\|\lambda-\lambda_q^t\right\|^2.$$

It is notable that we perform a fine-grained decomposition of the RHS terms. In fact, we will see that $(\nabla_\lambda L_q(\mathbf{w}_t,\lambda_q^t)-\mathbf{z}_q^t)(\lambda-\tilde{\lambda}_q^t)$ is closely related to the estimation error of $\mathbf{z}_q^t$. For $\nabla_\lambda L_q(\mathbf{w}_t,\lambda_q^t)(\tilde{\lambda}_q^t-\lambda_q^t)$ and $\mathbf{z}_q^t(\lambda-\tilde{\lambda}_q^t)$, they can actually be bounded by the smoothness

property of L_q and the update rule of $\mathbf{z}_q$. Specifically, since the function $L_q(\mathbf{w}_t, \lambda)$ is also smooth, we have

$$L_q(\mathbf{w}_t, \tilde{\lambda}_q^t) - \nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t)(\tilde{\lambda}_q^t - \lambda_q^t) - \frac{L_L}{2}\left\|\tilde{\lambda}_q^t - \lambda_q^t\right\|^2 \leq L_q(\mathbf{w}_t, \lambda_q^t).$$

Combining the above inequalities, we have

$$\begin{aligned} L_q(\mathbf{w}_t, \lambda) \geq\ & L_q(\mathbf{w}_t, \tilde{\lambda}_q^t) + \mathbf{z}_q^t(\lambda - \tilde{\lambda}_q^t) + (\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t) - \mathbf{z}_q^t)(\lambda - \tilde{\lambda}_q^t) \\ & + \frac{\mu_L}{2}\left\|\lambda - \lambda_q^t\right\|^2 - \frac{L_L}{2}\left\|\tilde{\lambda}_q^t - \lambda_q^t\right\|^2. \end{aligned}$$

Besides, by using the update rule of $\mathbf{z}_q$, we obtain

$$\begin{aligned} \mathbf{z}_q^t(\lambda - \tilde{\lambda}_q^t) &= \frac{1}{\tau}(\lambda_q^t - \tilde{\lambda}_q^t)(\lambda - \tilde{\lambda}_q^t) = \frac{1}{\tau}(\lambda_q^t - \tilde{\lambda}_q^t)(\lambda - \lambda_q^t) + \frac{1}{\tau}(\lambda_q^t - \tilde{\lambda}_q^t)(\lambda_q^t - \tilde{\lambda}_q^t) \\ &= \frac{1}{\tau}(\lambda_q^t - \tilde{\lambda}_q^t)(\lambda - \lambda_q^t) + \frac{1}{\tau}\left\|\lambda_q^t - \tilde{\lambda}_q^t\right\|^2. \end{aligned}$$

By combining the above two formulations, we obtain the following:

$$\begin{aligned} L_q(\mathbf{w}_t, \lambda) \geq\ & L_q(\mathbf{w}_t, \tilde{\lambda}_q^t) + \frac{1}{\tau}(\lambda_q^t - \tilde{\lambda}_q^t)(\lambda - \lambda_q^t) + \frac{1}{\tau}\left\|\lambda_q^t - \tilde{\lambda}_q^t\right\|^2 \\ & + (\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t) - \mathbf{z}_q^t)(\lambda - \tilde{\lambda}_q^t) + \frac{\mu_L}{2}\left\|\lambda - \lambda_q^t\right\|^2 - \frac{L_L}{2}\left\|\tilde{\lambda}_q^t - \lambda_q^t\right\|^2. \end{aligned}$$

Note that the second term on the RHS of the above equation can be bounded using (69). Thus, we have

$$\begin{aligned} & L_q(\mathbf{w}_t, \tilde{\lambda}_q^t) \geq L_q(\mathbf{w}_t, \lambda_q(\mathbf{w}_t)) \geq L_q(\mathbf{w}_t, \tilde{\lambda}_q^t) + \frac{1}{\tau}(\lambda_q^t - \tilde{\lambda}_q^t)(\lambda_q(\mathbf{w}_t) - \lambda_q^t) + \frac{1}{\tau}\left\|\lambda_q^t - \tilde{\lambda}_q^t\right\|^2 \\ & + (\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t) - \mathbf{z}_q^t)(\lambda_q(\mathbf{w}_t) - \tilde{\lambda}_q^t) + \frac{\mu_L}{2}\left\|\lambda_q(\mathbf{w}_t) - \lambda_q^t\right\|^2 - \frac{L_L}{2}\left\|\tilde{\lambda}_q^t - \lambda_q^t\right\|^2 \\ & \geq L_q(\mathbf{w}_t, \tilde{\lambda}_q^t) + \frac{1}{\tau}(\lambda_q^t - \tilde{\lambda}_q^t)(\lambda_q(\mathbf{w}_t) - \lambda_q^t) + \frac{1}{\tau}\left\|\lambda_q^t - \tilde{\lambda}_q^t\right\|^2 - \frac{2}{\mu_L}\left\|\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t) - \mathbf{z}_q^t\right\|^2 \\ & - \frac{\mu_L}{4}\left\|\lambda_q(\mathbf{w}_t) - \lambda_q^t\right\|^2 - \frac{\mu_L}{4}\left\|\lambda_q^t - \tilde{\lambda}_q^t\right\|^2 + \frac{\mu_L}{2}\left\|\lambda_q(\mathbf{w}_t) - \lambda_q^t\right\|^2 - \frac{L_L}{2}\left\|\tilde{\lambda}_q^t - \lambda_q^t\right\|^2 \\ & \geq L_q(\mathbf{w}_t, \tilde{\lambda}_q^t) + \frac{1}{2\tau_t\tau}\left(\left\|\tilde{\lambda}_q^t - \lambda_q(\mathbf{w}_t)\right\|^2 - \left\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\right\|^2 - \tau_t^2\left\|\tilde{\lambda}_q^t - \lambda_q^t\right\|^2\right) \\ & - \frac{2}{\mu_L}\left\|\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t) - \mathbf{z}_q^t\right\|^2 + \frac{\mu_L}{4}\left\|\lambda_q(\mathbf{w}_t) - \lambda_q^t\right\|^2 + \left(\frac{1}{\tau} - \frac{\mu_L}{4} - \frac{L_L}{2}\right)\left\|\tilde{\lambda}_q^t - \lambda_q^t\right\|^2. \end{aligned}$$

Since $L_q(\mathbf{w}_t, \tilde{\lambda}_q^t)$ cancels out on both sides of the equation, we finally obtain the following result regarding the estimation error bound for the components updated in the t-th iteration

$$\begin{aligned} & \left\|\tilde{\lambda}_q^t - \lambda_q(\mathbf{w}_t)\right\|^2 \\ & \leq \frac{4\tau_t\tau}{\mu_L}\left\|\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t) - \mathbf{z}_q^t\right\|^2 + \left(1 - \frac{\tau\tau_t\mu_L}{2}\right)\left\|\lambda_q(\mathbf{w}_t) - \lambda_q^t\right\|^2 \\ & \quad - 2\tau\tau_t\left(\frac{1}{\tau} - \frac{\mu_L}{4} - \frac{L_L}{2} - \frac{\tau_t}{2\tau}\right)\left\|\tilde{\lambda}_q^t - \lambda_q^t\right\|^2 \\ & \leq \left(1 - \frac{\tau\tau_t\mu_L}{2}\right)\left\|\lambda_q(\mathbf{w}_t) - \lambda_q^t\right\|^2 + \frac{4\tau_t\tau}{\mu_L}\left\|\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t) - \mathbf{z}_q^t\right\|^2 \\ & \quad - 2\tau\tau_t\left(\frac{3}{4\tau} - \frac{3}{4}L_L\right)\left\|\tilde{\lambda}_q^t - \lambda_q^t\right\|^2, \end{aligned}$$

where we use $\tau_t \leq \frac{1}{2}$ and $\mu_L \leq L_L$ in the second inequality, and use $\bar{\lambda}_q^t = \lambda_q^t + \tau_t(\tilde{\lambda}_q^t - \lambda_q^t)$ in the last inequality.

Notice that for $q \in \mathcal{B}$, we have $\bar{\lambda}_q^t = \lambda_q^{t+1}$, and according to the properties of conditional expectation, we obtain

$$\begin{aligned}
&\mathbb{E}\left[\left\|\lambda_q^{t+1} - \lambda_q(\mathbf{w}_t)\right\|^2\right] = \frac{|\mathcal{B}|}{N}\mathbb{E}\left[\left\|\bar{\lambda}_q^t - \lambda_q(\mathbf{w}_t)\right\|^2\right] + \frac{N-|\mathcal{B}|}{N}\mathbb{E}\left[\left\|\lambda_q^t - \lambda_q(\mathbf{w}_t)\right\|^2\right] \\
&\leq \left(1 - \frac{\tau\tau_t\mu_L|\mathcal{B}|}{2N}\right)\left\|\lambda_q(\mathbf{w}_t) - \lambda_q^t\right\|^2 + \frac{4\tau_t\tau|\mathcal{B}|}{\mu_L N}\left\|\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t) - \mathbf{z}_q^t\right\|^2 \\
&\quad - \frac{3\tau|\mathcal{B}|}{2\tau_t N}\left(\frac{1}{\tau} - L_L\right)\left\|\lambda_q^{t+1} - \lambda_q^t\right\|^2.
\end{aligned}$$

Further, we can derive the recursion for the estimation error bound as follows

$$\begin{aligned}
&\mathbb{E}\left[\left\|\lambda_q^{t+1} - \lambda_q(\mathbf{w}_{t+1})\right\|^2\right] \\
&\leq \left(1 + \frac{\tau\tau_t\mu_L|\mathcal{B}|}{4N}\right)\mathbb{E}\left[\left\|\lambda_q^{t+1} - \lambda_q(\mathbf{w}_t)\right\|^2\right] + \left(1 + \frac{4N}{\tau\tau_t\mu_L|\mathcal{B}|}\right)\mathbb{E}\left[\left\|\lambda_q(\mathbf{w}_{t+1}) - \lambda_q(\mathbf{w}_t)\right\|^2\right] \\
&\leq \left(1 - \frac{\tau\tau_t\mu_L|\mathcal{B}|}{4N}\right)\mathbb{E}\left[\left\|\lambda_q(\mathbf{w}_t) - \lambda_q^t\right\|^2\right] + \frac{8\tau_t\tau|\mathcal{B}|}{\mu_L N}\left\|\nabla_\lambda L_q(\mathbf{w}_t, \lambda_q^t) - \mathbf{z}_q^t\right\|^2 \\
&\quad - \frac{3\tau|\mathcal{B}|}{\tau_t N}\left(\frac{1}{\tau} - L_L\right)\left\|\lambda_q^{t+1} - \lambda_q^t\right\|^2 + \frac{8NC_\lambda^2}{\tau\tau_t\mu_L|\mathcal{B}|}\mathbb{E}\left[\left\|\mathbf{w}_{t+1} - \mathbf{w}_t\right\|^2\right],
\end{aligned}$$

where we use inequality $\|a+b\|^2 \leq (1+\gamma)\|a\|^2 + (1+\frac{1}{\gamma})\|b\|^2\ \gamma > 0, (1-\epsilon)(1+\frac{\epsilon}{2}) \leq 1-\frac{\epsilon}{2}$, and the assumption $\tau_t\tau \leq \frac{4N}{\mu_L|\mathcal{B}|}$ i.e., $\frac{\tau_t\tau\mu_L|\mathcal{B}|}{4N} \leq 1$ in the last inequality.

Taking summation over all queries and expectation over all randomness, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\lambda^{t+1} - \lambda(\mathbf{w}_{t+1})\right\|^2\right] \leq \left(1 - \frac{\tau\tau_t\mu_L|\mathcal{B}|}{4N}\right)\mathbb{E}\left[\left\|\lambda(\mathbf{w}_t) - \lambda^t\right\|^2\right] + \frac{8\tau_t\tau|\mathcal{B}|}{\mu_L N}\left\|\nabla_\lambda L(\mathbf{w}_t, \lambda^t) - \mathbf{z}^t\right\|^2 \\
&\quad - \frac{3\tau|\mathcal{B}|}{\tau_t N}\left(\frac{1}{\tau} - L_L\right)\left\|\lambda^{t+1} - \lambda^t\right\|^2 + \frac{8N^2C_\lambda^2}{\tau\tau_t\mu_L|\mathcal{B}|}\mathbb{E}\left[\left\|\mathbf{w}_{t+1} - \mathbf{w}_t\right\|^2\right].
\end{aligned}$$

We can see that due to the unique design of $\mathbf{z}$ and the meticulous analysis, the RHS of the above equation includes a negative $\left\|\lambda^{t+1} - \lambda^t\right\|^2$ term, This is crucial for offsetting the positive $\left\|\lambda^{t+1} - \lambda^t\right\|^2$ terms that appear elsewhere in the proof, thereby establishing convergence. □

Author Contributions All authors contributed to the algorithm design and analysis. Formal analysis was performed by Zi-Hao Qiu, Quanqi Hu, and Tianbao Yang. The experiments were performed by Zi-Hao Qiu and Yongjian Zhong. The first draft of the manuscript was written by Zi-Hao Qiu, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript. Lijun Zhang and Tianbao Yang acquired the funding, and were responsible for the research activity planning and execution.

Funding Q. Hu, Y. Zhong and T. Yang were partially supported by NSF Grant 2110545 and NSF Career Award 1844403. Z. Qiu and L. Zhang were partially supported by NSFC (62122037, 61921006). Part work of Z. Qiu was done when he was visiting T. Yang's lab virtually.

Data Availability Our proposed methods are implemented in the LibAUC library at https://www.libauc.org. The data and code to reproduce the results in this paper is available at https://github.com/zhqiu/NDCG-Optimization.

Declarations

Conflict of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

Ai, Q., Wang, X., Bruch, S., Golbandi, N., Bendersky, M., & Najork, M. (2019). Learning groupwise multivariate scoring functions using deep neural networks. In: *Proceedings of the 2019 ACM SIGIR international conference on theory of information retrieval*, pp 85–92.

Arjevani, Y., Carmon, Y., Duchi, J. C., Foster, D. J., Srebro, N., & Woodworth, B. (2022). Lower bounds for non-convex stochastic optimization. *Mathematical Programming, 199*(1), 165–214.

Balasubramanian, K., Ghadimi, S., & Nguyen, A. (2022). Stochastic multi-level composition optimization algorithms with level-independent convergence rates. *SIAM Journal on Optimization, 32*(2), 519–544.

Bennett, J., Lanning, S., *et al.* (2007). The Netflix prize. In: *Proceedings of KDD Cup and Workshop*, vol 2007, p 35.

Bhatia, K., Jain, H., Kar, P., Varma, M., & Jain, P. (2015). Sparse local embeddings for extreme multi-label classification. *Advances in Neural Information Processing Systems, 29*, 730–738.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005a). Learning to rank using gradient descent. In: *Proceedings of the 22nd international conference on machine learning*, pp 89–96.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005b). Learning to rank using gradient descent. In: *Proceedings of the 22nd international conference on machine learning*, pp 89–96.

Burges, C. J. (2010). From ranknet to lambdarank to lambdamart: An overview. *Learning, 11*(23–581), 81.

Cao, Z., Qin, T., Liu, T., Tsai, M., & Li, H. (2007). Learning to rank: from pairwise approach to listwise approach. In: *Proceedings of the 24th international conference on machine learning*, pp 129–136.

Chakrabarti, S., Khanna, R., Sawant, U., & Bhattacharyya, C. (2008). Structured learning for non-smooth ranking losses. In: *Proceeding of the 14th ACM SIGKDD conference on knowledge discovery and data mining*, pp 88–96.

Chapelle, O., & Chang, Y. (2011). Yahoo! learning to rank challenge overview. In: *Proceedings of the learning to rank challenge, PMLR*, pp 1–24.

Chen, T., Sun, Y., & Yin, W. (2021). Solving stochastic compositional optimization is nearly as easy as solving stochastic optimization. *IEEE Transactions on Signal Processing, 69*, 4937–4948.

Chen, T., Sun, Y., & Yin, W. (2022). A single-timescale stochastic bilevel optimization method. In: *Proceedings of the 25th international conference on artificial intelligence and statistics*, vol 151, pp 2466–2488.

Colson, B., Marcotte, P., & Savard, G. (2007). An overview of bilevel optimization. *Annals of Operations Research, 153*(1), 235–256.

Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In: *Proceedings of the 4th ACM conference on recommender systems*, pp 39–46.

Cutkosky, A., & Orabona, F. (2019). Momentum-based variance reduction in non-convex sgd. In: *Advances in neural information processing systems*, vol 32.

Dagréou, M., Ablin, P., Vaiter, S., & Moreau, T. (2022). A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. *Advances in Neural Information Processing Systems, 35*, 26698–26710.

Fang, C., Li, C. J., Lin, Z., & Zhang, T. (2018). Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In: *Advances in neural information processing systems*, vol 31.

Gao, H., Wang, Z., & Ji, S. (2018). Large-scale learnable graph convolutional networks. In: *Proceedings of the 24th ACM SIGKDD conference on knowledge discovery and data mining*, pp 1416–1424.

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

Ghadimi, S., & Wang, M. (2018). Approximation methods for bilevel programming. arXiv preprint arXiv:1802.02246.

Grover, A., Wang, E., Zweig, A., & Ermon, S. (2019). Stochastic optimization of sorting networks via continuous relaxations. In: *the 7th international conference on learning representations*.

Guo, Z., Hu, Q., Zhang, L., & Yang, T. (2021a). Randomized stochastic variance-reduced methods for multi-task stochastic bilevel optimization. arXiv preprint arXiv:2105.02266.

Guo, Z., Xu, Y., Yin, W., Jin, R., & Yang, T. (2021b). On stochastic moving-average estimators for non-convex optimization. arXiv preprint arXiv:2104.14840.
Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems, 5*(4), 1–19.
He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering. In: *Proceedings of the 26th international conference on world wide web*, pp 173–182.
He, X., He, Z., Song, J., Liu, Z., Jiang, Y. G., & Chua, T. S. (2018). Nais: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering, 30*(12), 2354–2366.
He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (2020). Lightgcn: Simplifying and powering graph convolution network for recommendation. In: *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pp 639–648.
Hong, M., Wai, H. T., Wang, Z., & Yang, Z. (2023). A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *SIAM Journal on Optimization, 33*(1), 147–180.
Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., & Leskovec, J. (2020a). Open graph benchmark: Datasets for machine learning on graphs. arXiv preprint arXiv:2005.00687.
Hu, Y., Zhang, S., Chen, X., & He, N. (2020b). Biased stochastic first-order methods for conditional stochastic optimization and applications in meta learning. *Advances in Neural Information Processing Systems, 33*, 2759–2770.
Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems, 20*(4), 422–446.
Jiang, W., Li, G., Wang, Y., Zhang, L., & Yang, T. (2022). Multi-block-single-probe variance reduced estimator for coupled compositional optimization. *Advances in Neural Information Processing Systems, 35*, 32499–32511.
Kishida, K. (2005). Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments. National Institute of Informatics Tokyo, Japan.
Kunisch, K., & Pock, T. (2013). A bilevel optimization approach for parameter learning in variational models. *SIAM Journal on Imaging Sciences, 6*(2), 938–983.
Li, J., Gu, B., & Huang, H. (2022). A fully single loop algorithm for bilevel optimization without hessian inverse. *Proceedings of the AAAI Conference on Artificial Intelligence, 36*, 7426–7434.
Lin, T., Jin, C., & Jordan, M. I. (2019). On gradient descent ascent for nonconvex-concave minimax problems. arXiv preprint arXiv:1906.00331.
Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In: *Proceedings of the IEEE international conference on computer vision*, pp 2980–2988.
Liu, N. N., & Yang, Q. (2008). Eigenrank: a ranking-oriented approach to collaborative filtering. In: *Proceedings of the 31st international ACM SIGIR conference on research and development in information retrieval*, pp 83–90.
Liu, R., Mu, P., Yuan, X., Zeng, S., & Zhang, J. (2020). A generic first-order algorithmic framework for bi-level programming beyond lower-level singleton. In: *Proceedings of the 37th international conference on machine learning*, pp 6305–6315.
Liu, T. Y. (2011). *Learning to rank for information retrieval*. Springer.
Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Math Program, 103*(1), 127–152.
Pasumarthi, R. K., Bruch, S., Wang, X., Li, C., Bendersky, M., Najork, M., Pfeifer, J., Golbandi, N., Anil, R., & Wolf, S. (2019). Tf-ranking: Scalable tensorflow library for learning-to-rank. In: *Proceedings of the 25th ACM SIGKDD conference on knowledge discovery and data mining*, pp 2970–2978.
Pobrotyn, P., & Bialobrzeski, R. (2021). Neuralndcg: Direct optimisation of a ranking metric via differentiable relaxation of sorting. arXiv preprint arXiv:2102.07831.
Pobrotyn, P., Bartczak, T., Synowiec, M., Białobrzeski, R., & Bojar, J. (2020). Context-aware learning to rank with self-attention. arXiv preprint arXiv:2005.10084.
Qi, Q., Luo, Y., Xu, Z., Ji, S., & Yang, T. (2021). Stochastic optimization of area under precision-recall curve for deep learning with provable convergence. *Advances in Neural Information Processing Systems, 34*, 1752–1765.
Qin, T., & Liu, T. Y. (2013). Introducing letor 4.0 datasets. arXiv preprint arXiv:1306.2597.
Qin, T., Zhang, X. D., Tsai, M. F., Wang, D. S., Liu, T. Y., & Li, H. (2008). Query-level loss functions for information retrieval. *Information Processing & Management, 44*(2), 838–855.
Qin, T., Liu, T. Y., & Li, H. (2010). A general approximation framework for direct optimization of information retrieval measures. *Information Retrieval, 13*(4), 375–397.
Qiu, Z. H., Hu, Q., Zhong, Y., Zhang, L., & Yang, T. (2022). Large-scale stochastic optimization of ndcg surrogates for deep learning with provable convergence. In: *Proceedings of the 39th international conference on machine learning*, pp 18122–18152.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. vol 28.
Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., & Huang, J. (2020). Self-supervised graph transformer on large-scale molecular data. *Advances in Neural Information Processing Systems, 33*, 12559–12571.
Singh, M. (2020). Scalability and sparsity issues in recommender datasets: A survey. *Knowledge and Information Systems, 62*(1), 1–43.
Swezey, R., Grover, A., Charron, B., & Ermon, S. (2021). Pirank: Scalable learning to rank via differentiable sorting. *Advances in Neural Information Processing Systems* 34.
Taylor, M., Guiver, J., Robertson, S., & Minka, T. (2008). Softrank: Optimizing non-smooth rank metrics. In: *Proceedings of the 2008 international conference on web search and web data mining*, pp 77–86.
Thonet, T., Cinar, Y. G., Gaussier, E., Li, M., & Renders, J. M. (2022). Listwise learning to rank based on approximate rank indicators. In: *Proceedings of the 36th AAAI conference on artificial intelligence*, pp 8494–8502.
Valizadegan, H., Jin, R., Zhang, R., & Mao, J. (2009). Learning to rank by optimizing NDCG measure. *Advances in Neural Information Processing Systems, 22*, 1883–1891.
Voorhees, E. M. (1999). Natural language processing and information retrieval. In: *International summer school on information extraction*, Springer, pp 32–48.
Wang, C., Zhang, M., Ma, W., Liu, Y., & Ma, S. (2020). Make it a chorus: knowledge-and time-aware item modeling for sequential recommendation. In: *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pp 109–118.
Wang, M., Fang, E. X., & Liu, H. (2017). Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming, 161*(1–2), 419–449.
Wang, X., Li, C., Golbandi, N., Bendersky, M., & Najork, M. (2018). The lambdaloss framework for ranking metric optimization. In: *Proceedings of The 27th ACM international conference on information and knowledge management*, pp 1313–1322.
Wang, X., He, X., Wang, M., Feng, F., & Chua, T. S. (2019). Neural graph collaborative filtering. In: *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pp 165–174.
Wu, M., Chang, Y., Zheng, Z., & Zha, H. (2009). Smoothing dcg for learning to rank: A novel approach using smoothed hinge functions. In: *Proceedings of the 18th ACM conference on information and knowledge management*, p 1923-1926.
Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., & Pande, V. (2018). Moleculenet: A benchmark for molecular machine learning. *Chemical Science, 9*(2), 513–530.
Xia, F., Liu, T. Y., Wang, J., Zhang, W., & Li, H. (2008). Listwise approach to learning to rank: theory and algorithm. In: *Proceedings of the 25th international conference on machine learning*, pp 1192–1199.
Xu, J., & Li, H. (2007). Adarank: a boosting algorithm for information retrieval. In: *Proceedings of the 30th international ACM SIGIR conference on research and development in information retrieval*, pp 391–398.
Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks? arXiv preprint arXiv:1810.00826.
Yang, T., & Lin, Q. (2018). Rsg: Beating subgradient method without smoothness and strong convexity. *Journal of Machine Learning Research, 19*(6), 1–33.
Yeh, J. Y., Lin, J. Y., Ke, H. R., Yang, W. P. (2007). Learning to rank for information retrieval using genetic programming. In: *Proceedings of SIGIR 2007 workshop on learning to rank for information retrieval*.
Yuan, T., Cheng, J., Zhang, X., Qiu, S., & Lu, H. (2014). Recommendation by mining multiple user behaviors with group sparsity. In: *Twenty-Eighth AAAI conference on artificial intelligence*.
Yuan, Z., Yan, Y., Sonka, M., & Yang, T. (2020). Robust deep auc maximization: A new surrogate loss and empirical studies on medical image classification. arXiv preprint arXiv:2012.03173.
Yuan, Z., Zhu, D., Qiu, Z. H., Li, G., Wang, X., & Yang, T. (2023). Libauc: A deep learning library for x-risk optimization. In: *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp 5487–5499.
Zhou, D., Xu, P., & Gu, Q. (2020). Stochastic nested variance reduction for nonconvex optimization. *The Journal of Machine Learning Research, 21*(1), 4130–4192.