ScaleBiO: Scalable Bilevel Optimization for LLM Data Reweighting

Rui Pan[1*] , **Dylan Zhang**[1*] , **Hanning Zhang**[1*] , **Xingyuan Pan**[1*] , **Minrui Xu**[2] ,
Jipeng Zhang[2] , **Renjie Pi**[2], **Xiaoyu Wang**[2] , **Tong Zhang**[1]

[1]University of Illinois Urbana-Champaign

[2]The Hong Kong University of Science and Technology

{ruip4, shizhuo2, hanning5, xp12}@illinois.edu

{mxubh, jzhanggr, rpi, maxywang}@ust.hk

tongzhang@tongzhang-ml.org

Abstract

Bilevel optimization has shown its utility across various machine learning settings, yet most algorithms in practice require second-order information, making it challenging to scale them up. Only recently, a paradigm of first-order algorithms has emerged in the theoretical literature, capable of effectively addressing bilevel optimization problems. Nevertheless, the practical efficiency of this paradigm remains unverified, particularly in the context of large language models (LLMs). This paper introduces the first scalable instantiation of this paradigm called **ScaleBiO**, focusing on bilevel optimization for large-scale LLM data reweighting. By combining with a recently proposed memory-efficient training technique called LISA, our novel algorithm allows the paradigm to scale to $\sim$30B-sized LLMs on $8\times$H100 GPUs, marking the first successful application of bilevel optimization under practical scenarios for large-sized LLMs. Empirically, extensive experiments on data reweighting verify the effectiveness of ScaleBiO for different-scaled models, including Llama-3-8B, Gemma-2-9B, Qwen-2-7B, and Qwen-2.5-32B, where bilevel optimization succeeds in instruction-following and math reasoning tasks, outperforming several popular baselines, including uniform sampling, influence-aware data filtering, and reference-model-based sampling methods. Theoretically, ScaleBiO ensures the optimality of the learned data weights, along with a convergence guarantee matching the conventional first-order bilevel optimization paradigm on smooth and strongly convex objectives.

1 Introduction

Data quality plays a crucial role in the success of Large Language Models (LLMs) (Gunasekar et al., 2023; Yang et al., 2024a; Dubey et al., 2024). Among various techniques for improving data quality, data reweighting has gained increasing attention for advancing LLMs, particularly in areas such as enhancing fairness (Roh et al., 2021, 2020, 2023), accelerating pre-training (Xia et al., 2024a; Xie et al., 2024), strengthening training robustness (Jain et al., 2024), and boosting transfer learning (Xia et al., 2024b). It is widely acknowledged that data reweighting and filtering techniques can lead to significant improvements across a diverse range of tasks (Gunasekar et al., 2023; Xu et al., 2024; Hu et al., 2024; Yang et al., 2024a; Liu et al., 2024).

On the other hand, Bilevel Optimization (BO) has emerged as a prominent area of research for solving this data re-weighting task, which draws substantial attention due to its effectiveness in numerous machine learning applications, such as hyperparameter optimization (Domke, 2012; Maclaurin et al., 2015; Franceschi et al., 2017; Lorraine et al., 2020), meta-learning (Andrychowicz et al., 2016; Franceschi et al., 2018; Rajeswaran et al., 2019) and reinforcement learning (Konda and Tsitsiklis, 1999; Hong et al., 2020). In its standard formulation, bilevel optimization involves a two-level hierarchical structure with inner-outer dependence,

$$
\begin{aligned}
\min_{\lambda \in \Lambda} \quad & \mathcal{L}(\lambda) = L_1(\lambda, w_*(\lambda)) \\
\text{s.t.} \quad & w_*(\lambda) = \arg\min_w L_2(\lambda, w). \quad (1)
\end{aligned}
$$

For example, on data reweighting tasks, λ are weights of different data sources, w represents the trainable model parameters, $w_*(\lambda)$ means the optimal parameters trained on a weighted dataset, while outer function L_1 and inner function L_2 stand for validation and training losses, respectively.

Despite the inherent flexibility and applicability of bilevel optimization across a wide range of problems, its extensive utilization in large-scale problems has been relatively limited thus far. The primary obstacle hindering the scalability of bilevel optimization arises from the interdependence between the upper-level and lower-level problems.

---

* Equal Contribution.

31959

The natural gradient-based iterative method of solving Problem (1) is to compute (or estimate) the hyper-gradient

$$\frac{\partial \mathcal{L}(\lambda)}{\partial \lambda} = \frac{\partial L_1(w_*(\lambda), \lambda)}{\partial \lambda} + \frac{\partial L_1(w_*, \lambda)}{\partial w_*} \frac{\partial w_*(\lambda)}{\partial \lambda}, \quad (2)$$

where the main challenge lies in efficiently computing or approximating the derivative $\partial w_*(\lambda)/\partial \lambda$ in (2). There is a line of research (Domke, 2012; Pedregosa, 2016; Grazzi et al., 2020; Lorraine et al., 2020; Franceschi et al., 2017; Shaban et al., 2019; Grazzi et al., 2020; Ghadimi and Wang, 2018; Hong et al., 2020; Yang et al., 2021; Ji et al., 2021; Chen et al., 2022) have been tempted to address this challenge. However, these works all require the computation of Hessian, Jacobian, or their products with vectors, which can be computationally expensive and memory-intensive for large-scale problems. Recently, Kwon et al. (2023) proposed a fully first-order method for stochastic bilevel optimization via only the first-order gradient oracle. This approach addresses the challenges associated with second-order computations and offers promising potential for stochastic bilevel optimization.

Despite these groundbreaking advancements in algorithms and theory, the practical performance of theoretically-optimal bilevel optimization algorithms in large-scale real-world settings has yet to be thoroughly investigated. Aiming to close this gap, this paper considers a practical scenario where LLMs are fine-tuned with different sources of datasets. We identify a significant challenge in determining the optimal sampling weights for each data source. For instance, Wang et al. (2024) have demonstrated that LLMs' task-specific performance degrades in the presence of certain training datasets. However, the inclusion and combination of various datasets should intuitively enhance the models' overall performance with proper sampling weights. This data-task misalignment poses a primary challenge in training LLMs with multiple data sources:

How to balance each data source in the training dataset to obtain optimal performance?

Various methods have been proposed in attempting to address this challenge. However, they either rely on intuitive preset (Zhou et al., 2024; Muennighoff et al., 2022; Du et al., 2022b; Almazrouei et al., 2023) or lacks theoretical guarantees (Xia et al., 2024b; Xie et al., 2024; Xia et al., 2024a), leading

to suboptimal sampling weights. To this end, we test theoretical-optimal bilevel optimization in data re-weighting tasks for LLMs, aiming to overcome the limitations of existing methods. Our primary contributions are summarized as follows:

- We propose the first scalable and theoretically-optimal instantiation of bilevel optimization on large-sized LLM training problems, which is capable of scaling to models with ~ 30 billion parameters.

- We successfully bridge the gap between theoretical advancements in bilevel optimization and their application in data reweighting, allowing the optimal data weights to be learnable for large-scale LLMs.

- We provide both experimental and theoretical results to demonstrate the effectiveness of ScaleBiO. Empirically, ScaleBiO outperforms popular data filtering/reweighting baselines, including uniform sampling, LESS (Xia et al., 2024b), and RHO-LOSS (Mindermann et al., 2022a), surpassing them by a non-trivial margin of $1\% - 9\%$ in GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021b). This superiority of ScaleBiO also holds in instruction following tasks. Theoretically, ScaleBiO's convergence guarantee matches the results of Kwon et al. (2023) on smooth and strongly convex objectives.

2 Related Work

2.1 Bilevel Optimization

Traditional bilevel optimization algorithms are majorly categorized into two classes: 1) approximate implicit differentiable (AID) methods (Domke, 2012; Pedregosa, 2016; Grazzi et al., 2020; Lorraine et al., 2020), or 2) iterative differentiable (ITD) methods (Domke, 2012; Maclaurin et al., 2015; Franceschi et al., 2017; Shaban et al., 2019; Grazzi et al., 2020). Both approaches follow a two-loops manner and require huge computational cost for large-scale problems. To reduce the cost, attempts in stochastic bilevel optimization have been made (Ghadimi and Wang, 2018; Hong et al., 2020; Ji et al., 2021; Chen et al., 2022; Khanduri et al., 2021), which significantly improve the efficiency of traditional methods, but still lack practicality for large-scale settings due to the requirements of second-order information, such as Jacobian- and

Method	Description	Task	Model	Size
RMD (Bengio, 2000)	2-nd order, deterministic	hyperparameter optimization	Linear	<1M
CG (Grazzi et al., 2020)	2-nd order, deterministic	equilibrium models	CNN	<1M
stocBiO (Ji et al., 2021)	2-nd order, stochastic	meta learning	CNN	<1M
FdeHBO (Yang et al., 2023)	1-st order, stochastic	hyper-representation	LeNet	<1M
BOME (Liu et al., 2022)	1-st order, stochastic	data hyper-cleaning	Linear	<1M
SOBA (Dagréou et al., 2022)	2-nd order, stochastic	data reweighting	Transformers	7M
PZOBO (Sow et al., 2022)	1-st order, stochastic	few-shot meta-learning	ResNet	12M
SAMA (Choe et al., 2023)	2-nd order, stochastic	noisy fine-tuning	BERT	110M
BFTSS (Somayajula et al., 2023)	1-st order, stochastic	task-dependent structure learning	BERT	336M
(FG)^{2}U (Shen et al., 2024)	1-st order, stochastic	online adaptation	GPT-2-XL	1.5B
ScaleBiO (Ours)	1-st order, stochastic	data reweighting	Qwen-2.5-32B	**32B**

Table 1: In this table, we compare the maximal model size implemented in their original paper, where 'M' stands for million and 'B' stands for billion. We also summarize their methods in Description and report the task they tested.

Hessian-vector products for estimating the hyper-gradient. Sow et al. (2022); Yang et al. (2023) attempt to approximate the Jacobian matrix $\nabla y^*(x)$ in (2) by finite differences, but the finite-different estimation can be sensitive to the selection of the smoothing constant and may suffer from some numerical issues in practice (Jorge and Stephen, 2006).

Recently, a new paradigm of fully first-order penalty-based methods has been introduced, which reformulate the inner-level problem into the optimality constraint (Liu et al., 2022; Kwon et al., 2023; Chen et al., 2023). Liu et al. (2022) first find the hypergradient only involving first-order information, while the method only applies to deterministic functions. Kwon et al. (2023) introduced a first-order gradient-based approach that avoids the estimations of Hessian or Jacobian. This method is easily adapted and extended to stochastic bilevel optimization settings. Chen et al. (2023) provided the near-optimal sample complexity, which improves the theoretical result of (Kwon et al., 2023) in the deterministic bilevel optimization. These results verify the effectiveness of the proposed paradigm in theory, yet its practical applications in large-scale LLM settings remain unexplored.

On the practical side, bilevel optimization has been explored in various NLP tasks. Somayajula et al. (2023) use bilevel optimization to learn the task-dependent similarity structure. Although their approach demonstrates effectiveness on BERT models (Devlin et al., 2018), the finite difference approximation suffers from high error and therefore lacks the scalability in LLMs with billions of pa-

rameters. Grangier et al. (2024) adopt SOBA (Dagréou et al., 2022) to modify the training data distributions for language modeling under domain shift. However, the algorithm still requires gradient approximation and Hessian-vector products, posing challenges to scalability and engineering for large-scale problems. We summarize typical bilevel algorithms and their model sizes in Table 1, where to the best of our knowledge, no approach listed in the table has been successfully applied to over 3B-sized LLM models.

2.2 Data Reweighting

The proportion of training data sources significantly affects the performance of large language models (Du et al., 2022a; Xie et al., 2023). To this end, various methods have been proposed to reweight data sources for optimal training data mixture. For example, Mindermann et al. (2022b) utilizes the loss gap between a trained model and a base model to identify learnable data samples, assigning them higher weights on the fly. Thakkar et al. (2023) propose to use a self-influence score to guide the reweighting in mini-batch during pre-training. Xia et al. (2024a) leverages reference losses on validation sets and adjusts the weights dynamically, adding minimal overhead to standard training. DoReMi (Xie et al., 2024) applies distributionally robust optimization (DRO) to tuning the domain weights without knowledge of downstream tasks. Nevertheless, none of the aforementioned methods ensures the optimality of the learned data weights, let alone scalable experiments on over 30B-sized models.

Algorithm 1 ScaleBiO for high-dimensional and large-scale minimax problems

---
1: **Input:** step-sizes $\{\eta_u, \eta_\omega, \eta_\lambda\}$, penalty α, and initialization λ_0, u_0, w_0
2: **for** $k = 0 : K - 1$ **do**
3:     Uniformly and independently select two j_k, r_k block coordinates from $\{1, 2, \cdots, J\}$, respectively
4:     Generating i.i.d. samples $\{D_{\text{tr}}^k, D_{\text{val}}^k\}$ from training dataset D_{tr} and validation dataset D_{val}
5:     $u_{k+1}^{j_k} = u_k^{j_k} - \alpha \eta_u \nabla_{j_k} L_2(\lambda_k, u_k; D_{\text{tr}}^k)$
6:     $u_{k+1} = u_k + U_{j_k}(u_{k+1}^{j_k} - u_k^{j_k})$ ▷ Map the permuted block parameters back
7:     $w_{k+1}^{r_k} = w_k^{r_k} - \eta_w \left(\nabla_{r_k} L_1(\lambda_k, w_k; D_{\text{val}}^k) + \alpha \nabla_{r_k} L_2(\lambda_k, w_k; D_{\text{tr}}^k) \right)$
8:     $w_{k+1} = w_k + W_{r_k}(w_{k+1}^{r_k} - w_k^{r_k})$ ▷ Map the permuted block parameters back
9:     $\lambda_{k+1} = \lambda_k - \eta_\lambda \left(\nabla L_1(\lambda_k, w_k; D_{\text{val}}^k) + \alpha \left(\nabla L_2(\lambda_k, w_k; D_{\text{tr}}^k) - \nabla L_2(\lambda_k, u_k; D_{\text{tr}}^k) \right) \right)$
10: **end for**
11: **Output:** (λ_K, w_K, u_K)

---

3 Methods

In this section, we elaborate on our ScaleBiO method for finding the optimal sampling weights when training large-scale LLMs. We first formulate this problem as a bilevel optimization problem in Section 3.1 and then develop an efficient training method for our formulation in Section 3.2.

3.1 Problem Formulation

Suppose that m data sources are available for training, e.g. Alpaca (Taori et al., 2023), FLAN (Wei et al., 2021), and ShareGPT (Chiang et al., 2023), where each source S_i is a set of n_i examples $S_i = \{a_1^i, a_2^i, \ldots, a_{n_i}^i\}$. The desired dataset mixture can be obtained by assigning each data source S_i a sampling weight p_i that satisfies $\sum_{i=1}^m p_i = 1$.

Accordingly, each data source S_i contributes $p_i|\mathcal{D}_{\text{trn}}|$ samples to the training dataset $\mathcal{D}_{\text{trn}}$, where the sampling weights can be optimized to minimize the model's loss on validation set $\mathcal{D}_{\text{val}}$. This leads to the following bilevel optimization problem:

$$\min_{p \in \Lambda} L_{\text{val}}(w^*(p))$$

$$\text{s.t. } w^*(p) = \arg\min_w \sum_{i=1}^m \frac{p_i}{n_i} \sum_{j=1}^{n_i} L_{\text{trn}}(w, a_j^i)$$

where w denotes the parameters of LLM, $\{p_i\}$ is the probability distribution over m data sources, L_{val} and L_{trn} respectively denote the language modeling loss on D_{val} and D_{trn}. To ensure non-negativity of the sampling weights $\{p_i\}_{i=1}^m$, an additional trainable variable $\lambda \in \mathbb{R}^m$ is introduced to represent $p_i = e^{\lambda_i} / \sum_{j=1}^m e^{\lambda_j}$.

3.2 Fully First-order Hypergradient Method

Recent advancements in the theoretical literature of bilevel optimization allow scalable methods to be developed. The main idea is actually quite similar to merging digits in radix sort. The first step is to decouple two "digit" terms and view the inner-level problem in (1) as a higher-order digit,

$$\min_{\lambda \in \Lambda, w} \quad L_1(\lambda, w)$$

$$\text{s.t.} \quad L_2(\lambda, w) - \min_u L_2(\lambda, u) = 0. \quad (3)$$

Here the auxiliary variable u is introduced to detach the inner-outer dependency, which transforms the inner problem $w_*(\lambda) = \arg\min_w L_2(\lambda, w)$ to be the constraint $L_2(\lambda, w) - \min_u L_2(\lambda, u) = 0$. By prioritizing the "high-order" constraint term of (3) with multiplier $\alpha > 0$, the minimax formulation in Kwon et al. (2023); Lu and Mei (2023) is recovered:

$$\min_{\lambda \in \Lambda, w} \max_u \mathcal{L}^\alpha(\lambda, w, u). \quad (4)$$

where

$$\mathcal{L}^\alpha(\lambda, w, u) = L_1(\lambda, w) + \alpha \left(L_2(\lambda, w) - L_2(\lambda, u) \right)$$

In this way, the approximation of both inner constraint and outer optimum can be obtained during the same optimization process, and α controls the priority. When $\alpha \to \infty$, the bilevel problem (1) is equivalent to the minimax problem (4) under certain smoothness assumptions.

To precisely describe the optimality of the minimax problem with the stationarity of the bilevel problem, the following notations in (4) are overloaded and defined as

$$\Phi^\alpha(\lambda, w) := \max_u \mathcal{L}^\alpha(\lambda, w, u); \quad (5)$$

$$u_*(\lambda) := \arg\max_u \mathcal{L}^\alpha(\lambda, w, u); \quad (6)$$

$$\Gamma^\alpha(\lambda) := \min_w \Phi^\alpha(\lambda, w); \quad (7)$$

$$w_*^\alpha(\lambda) = \arg\min_w \Phi^\alpha(\lambda, w). \quad (8)$$

Additionally, the following assumptions are made for the proposed minimax problem throughout this paper.

Assumption 1. *Suppose that*

(1) $L_1(\lambda, w)$ *is twice continuously differentiable, ℓ_{10}-Lipschitz continuous in w; ℓ_{11}-gradient Lipschitz.*

(2) $L_2(\lambda, w)$ *is ℓ_{21}-gradient Lipschitz, ℓ_{22}-Hessian Lipschitz, and μ_2-strongly convex in w.*

Lemma 1. *Under Assumption 1, if $\alpha > 2\ell_{11}/\mu_2$, we have*

$$|\mathcal{L}(\lambda) - \Gamma^\alpha(\lambda)| \leq \mathcal{O}\left(\frac{1}{\alpha}\right) \qquad (9)$$

$$\|\nabla\mathcal{L}(\lambda) - \nabla\Gamma^\alpha(\lambda)\| \leq \mathcal{O}\left(\frac{1}{\alpha}\right) \qquad (10)$$

$$\|\nabla^2\Gamma^\alpha(\lambda)\| \leq \mathcal{O}(\kappa^3) \qquad (11)$$

where the condition number κ is defined by $\max\{\ell_{10}, \ell_{11}, \ell_{21}, \ell_{22}\}/\mu_2$.

Under Assumption 1, as indicated by Lemma 1, if α goes to infinity, the stationary point of the minimax problem (4) is also a stationary point of the bilevel problem (1). Intuitively, it is like finding a way to the same peak of the mountain with distinct paths, where bilevel optimization suggests a winding road, and minimax utilizes a helicopter.

3.3 Proposed Algorithm

To solve this reformulated large-scale min-max problem, we introduce ScaleBiO in Algorithm 1, a single-loop framework that is capable of scaling up to 30B-sized models. To further reduce memory consumption, randomized block coordinate methods are employed to update the inner variables u, w (Nesterov, 2012; Pan et al., 2024), where $U_j, W_j \in \mathbb{R}^{d \times d_j}$ denotes the block matrices that map the permutation of parameters back to model weights. The optimizer choice varies depending on the backbone model, where Adam or AdamW (Kingma and Ba, 2015; Loshchilov, 2017) is much preferable for LLMs. The penalty coefficient α is predefined with a large factor that ensures the min-max solution is a good approximation of the original bilevel problem.

3.4 Theoretical Results

In this part, we provide a convergence analysis of Algorithm 1, explaining how fast the algorithm can

reach a desired stationary point. Before showing the details of theoretical results, we introduce the notations for partitions. Let $\{x^1, x^2, \cdots, x^J\}$ with $x^j \in \mathbb{R}^{d_j \times 1}$ be J non-overlapping blocks of x. Let the matrix $U_j \in \mathbb{R}^{d \times d_j}$ be d_j columns of a $d \times d$ permutation matrix U corresponding to j block coordinates in x. For any partition of x and U,

$$x = \sum_{j=1}^{J} U_j x^j, \quad x_j = U_j^T x. \qquad (12)$$

The essential lemmas are available in Appendix C to show the theoretical properties of minimax objective $\mathcal{L}^\alpha$ in (4), as well as its optimizers u_* and w_*^α. Lemma 1 provides clear evidence that $\Gamma^\alpha(\lambda)$ is smooth with parameter $\ell_\Gamma = \mathcal{O}(\kappa^3)$ which is independent on the multiplier α.

Theorem 1. *Suppose that Assumptions 1 holds and the parameter α and step-sizes $\eta_u, \eta_w, \eta_\lambda$ are properly chosen such that*

$$\alpha = K^{1/7}, \eta_u = \eta_w = \frac{\eta_0}{K^{4/7}}, \eta_\lambda = \frac{\eta_0^\lambda}{K^{5/7}}.$$

Consider Algorithm 1, if $\alpha \geq \ell_{11}/\mu_2$, for $\eta_0^\lambda \leq 1/(8\ell_\Gamma)$, $\eta_0 \leq 8J/\mu_2$ and $\eta_0/\eta_0^\lambda \geq 6\sqrt{2}\kappa^2 J$, then

$$\mathbb{E}\left[\left\|\nabla\mathcal{L}(\tilde{\lambda})\right\|^2\right] \leq \mathcal{O}\left(\frac{1}{K^{2/7}}\right) \qquad (13)$$

where $\tilde{\lambda}$ is uniformly chosen from $\{\lambda_k\}_{k=1}^K$.

When considering the batch size $B = \mathcal{O}(1)$, the complexity of finding an ϵ-stationary point of Algorithm 1 is $\mathcal{O}(\epsilon^{-7})$, which matches that of (Kwon et al., 2023). The proof of Theorem 1 is provided in Appendix D.

4 Experiments

To verify the effectiveness of ScaleBiO, two types of experiment are conducted: (1) *Small Scale Experiments* in Section 4.1, which offers intuitions for understanding ScaleBiO's theoretical properties in toy settings, and (2) *Real-World Application Experiments* in Section 4.2 that validate its scalability in larger-sized models on instruction-following and mathematical reasoning tasks.

4.1 Small Scale Experiments

To understand the properties of ScaleBiO, experiments with GPT-2 (124M) are conducted on three tasks with synthetic datasets: data denoising, multilingual training, and instruction-following finetuning. Full details are available in Appendix B.1.
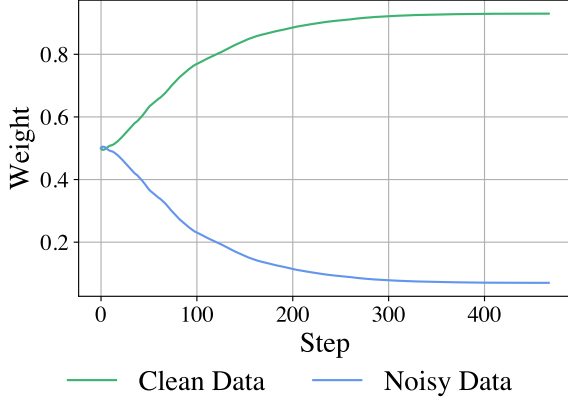
Figure 1: **Data denoising with GPT-2**: weights for noisy data and clean data.

Figure 2: **Multilingual reweighting with GPT-2**: weights of Chinese and English. Training set: 1:1; Validation set: 6:4.

4.1.1 Data Denoising

In this experiment, ScaleBiO's data denoising ability is tested, where noisy samples are expected to be assigned with zero weights. The validation dataset, denoted as $\mathcal{D}_{val}$, comprises 1000 clean samples randomly selected from the Alpaca dataset (Taori et al., 2023). The training dataset, $\mathcal{D}_{trn}$, is derived from two distinct sources: the first includes 1000 clean samples also from Alpaca, while the second incorporates 9000 samples from Alpaca that have been artificially corrupted with synthetic noise, where the outputs are replaced with ".".

Figure 1 demonstrates that our approach has a robust capability to mitigate the influence of harmful data sources via automatic data denoising, where ScaleBiO assigns minimal weight to noisy data sources, effectively filtering the irrelevant samples.

4.1.2 Multilingual Training

It is also intriguing to check if ScaleBiO can recover optimal sampling weights for more general distributions. To this end, the multilingual training experiments are introduced, where the validation data $\mathcal{D}_{val}$ comprises 600 random samples from Alpaca-GPT4-ZH (Peng et al., 2023) and 400 random samples from Alpaca-GPT4-EN (Peng et al., 2023). Hence, the underlying optimal weight is 6:4. In contrast, the training set $\mathcal{D}_{trn}$ has a 1:1 mix ratio, which consists of 40,000/40,000 random examples from Alpaca-GPT4-EN and Alpaca-GPT4-ZH, respectively.

As shown in Figure 2, ScaleBiO nearly replicates the optimal 6:4 ratio after reweighting the training data. This serves as another concrete proof that ScaleBiO is capable of adapting training data weights optimally to downstream validation datasets.
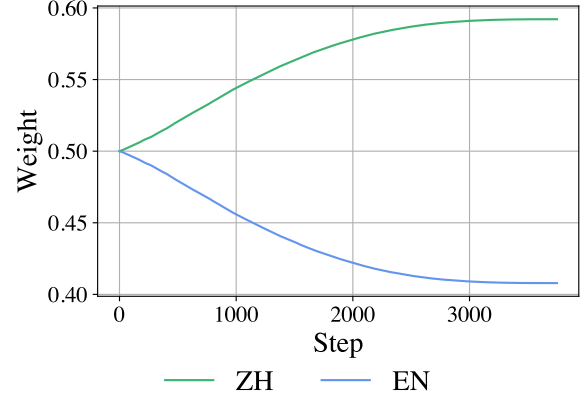
4.1.3 Instruction Following

In instruction-following fine-tuning tasks, there is a fundamental tradeoff between diversity and quality. To verify if ScaleBiO can deduce these implicit weights of low- and high-quality datasets, experiments are conducted on instruction-following tasks with GPT-2, where Alpaca and Alpaca-GPT4 (Peng et al., 2023) are employed. Here Alpaca-GPT4 shares the same instructions and input as Alpaca, whose high quality is distinguished by its outputs generated from a more sophisticated model GPT-4 (Achiam et al., 2023). The validation data for bilevel optimization $\mathcal{D}_{val}$ consists of 1000 random samples from Alpaca-GPT4, while the training data $\mathcal{D}_{trn}$ consists of 2 separate parts: 1000 random samples from Alpaca-GPT4 and 9000 random samples from Alpaca.
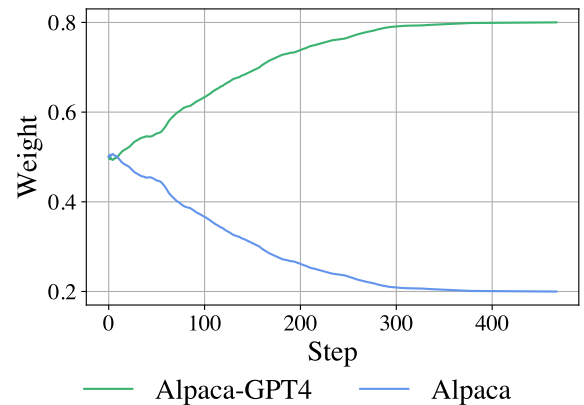

Figure 3: **Instruction Following with GPT-2**: weights for Alpaca-GPT4 and Alpaca.

Method	Model		
	Llama-3-8B	Qwen-2-7B	Gemma-2-9B
SOBA (Dagréou et al., 2022)	OOM	OOM	OOM
Uniform Weighting	6.11	6.66	5.31
LESS (Xia et al., 2024b)	6.06	7.18	7.20
RHO-LOSS (Mindermann et al., 2022a)	6.89	7.34	7.38
ScaleBiO	**7.12**	**7.76**	**7.51**

Table 2: **Instruction Following**. Here all methods are evaluated in MT-Bench (Zheng et al., 2023b) with GPT-4o LLM judge, where scores range from 0 to 10. OOM stands for out of memory.

As shown in Figure 3, although Alpaca-GPT4 accounts for only a small proportion of the training data (10%), it is highlighted by ScaleBiO, revealing that it can effectively up-weights the high-quality data source, leading to improved model outcomes.

4.2 Real-World Application Experiments

In this section, ScaleBiO is tested in real-world data reweighting applications, including instruction following and mathematical reasoning tasks, which demonstrates its scalability and empirical benefits in practice.

4.2.1 Instruction Following

In the instruction following setting, ScaleBiO is validated under the real-world scenario where the data collection is conducted in a non-filtered fashion, e.g. datasets of weak correlations with the downstream task may be included.

Setup Three ∼7B-sized LLMs, including Llama-3-8B (Dubey et al., 2024), Qwen-2-7B (Yang et al., 2024b), and Gemma-2-9B (Team et al., 2024) are evaluated in the widely adopted benchmark of MT-Bench (Zheng et al., 2023b), where a GPT-4o judge (Hurst et al., 2024) is employed to score the generated responses of each model on 80 high-quality multi-turn questions. Different aspects of the model, such as writing, role play, and STEM, are scored by the GPT-4o judge and averaged in the final MT-Bench score.

The training portfolio comprises ∼4.2M total samples from 18 different sources, as detailed in Table 10 in Appendix B.2. All datasets are collected in a task-agnostic fashion, where datasets necessary for general instruction following tasks, but have weak correlations to MT-Bench are also included. One typical example of such datasets is multi-lingual conversations.

To form the training set, all data reweighting methods are required to assign weights to 18 sources and extract 10K samples from the 4.2M

portfolio. The target model will be trained on the training set and evaluated to produce the final MT-Bench score.

Results As shown in Table 2, ScaleBiO is the only bilevel algorithm capable of yielding meaningful weights across data sources. On top of that, SaleBiO outperforms popular influence-aware data filtering method LESS (Xia et al., 2024b) and reference-model-based data reweighting approach RHO-LOSS (Mindermann et al., 2022a), both are considered strong non-bilevel baselines in the data reweighting literature.

4.2.2 Mathematical Reasoning

To further demonstrate ScaleBiO's data reweighting ability under scenarios with refined dataset sources, a training portfolio similar to Dong et al. (2024) is adopted for mathematical reasoning, where datasets detailed in Table 3 are proven to be conducive to downstream math tasks. Here coding and instruction following datasets are considered necessary, which allow the LLM to learn minimum reasoning and instruction following abilities for answering mathematical questions.

Dataset	#Samples
hkust-nlp/dart-math-uniform	591K
Open-Orca/SlimOrca	518K
openbmb/UltraInteract_sft	289K
TIGER-Lab/MathInstruct	262K
microsoft/orca-math-word-problems-200k	200K
WizardLMTeam/WizardLM_evol_instruct_V2_196k	196K
ise-uiuc/Magicoder-Evol-Instruct-110K	110K
anon8231489123/ShareGPT_Vicuna_unfiltered	94K
teknium/GPTeacher-General-Instruct	89K
teknium/GPT4-LLM-Cleaned	55K
Total	**2.4M**

Table 3: Dataset for **Mathematical Reasoning**.

Setup Similar to Section 4.2.1, three models of Llama-3-8B, Qwen-2-7B and Gemma-2-9B are employed. For evaluation, the standard math

Method	GSM8K (Cobbe et al., 2021)			MATH (Hendrycks et al., 2021b)		
	Llama-3-8B	Qwen-2-7B	Gemma-2-9B	Llama-3-8B	Qwen-2-7B	Gemma-2-9B
SOBA	OOM	OOM	OOM	OOM	OOM	OOM
Uniform Weighting	53.6	65.0	56.3	14.2	36.7	24.8
RHO-LOSS	53.8	70.7	56.9	13.6	38.8	25.0
LESS	52.5	71.6	57.9	14.0	38.9	28.3
ScaleBiO	**56.2**	**74.1**	**59.4**	**15.1**	**41.7**	**30.0**

Table 4: **Mathematical Reasoning**. Here all metrics are accuracies ranging from 0 to 100. OOM stands for out of memory.

benchmark of GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021b) are utilized. The reweighting methods are expected to extract 20K samples from the given 10 sources to form the training set, with the target model fine-tuned on the set and evaluated to produce the final accuracy.

Results As shown in Table 4, ScaleBiO consistently outperforms all baselines across different models and benchmarks, by a non-trivial margin of 1%-9%, which demonstrates ScaleBiO's superiority in reweighting task-oriented datasets.

Method	GSM8K	MATH
LESS	OOM	OOM
RHO-LOSS	OOM	OOM
Uniform Weighting	78.1	54.0
ScaleBiO	**87.1**	**59.8**

Table 5: **Large-Scale Mathematical Reasoning** on Qwen-2.5-32B (Yang et al., 2024a). Here all metrics are accuracies ranging from 0 to 100. OOM stands for out of memory.

To further validate ScaleBiO's scalability in even larger-sized LLMs, Qwen-2.5-32B (Yang et al., 2024a) is adopted in the same setting. As presented in Table 5, ScaleBiO is the only data reweighting implementation capable of scaling up to this size. Here LESS and RHO-LOSS both run out of GPU memories due to their non-scalable implementation or requirement for extra reference models. In contrast, ScaleBiO has the same space complexity as full parameter fine-tuning, allowing it to be applicable in any single-node training scenarios.

5 Discussion

Existence of Optimal Datasets? As ScaleBiO is capable of learning optimal task-orient data weights for different models, it serves as a great tool to inspect data weight transferability across different models. As it is unsurprising to find

that Llama-3-8B-learned data weights can be transferred to Llama-3-70B and still yield certain improvement (Table 6), it is more intriguing to observe that the learned data weights vary significantly across different model families, as shown in Table 7 of Appendix A.1.

Model	MT-Bench score	
Llama-3-8B → Llama-3-70B	*Uniform Weighting*	7.85
	ScaleBiO	**8.05**

Table 6: MT-Bench results of Llama-3-70B with transfer trained weights from Llama-3-8B.

It is worth noticing that the weight difference is much smaller inside the same model family. This phenomenon is conjectured to stem from the difference in LLMs' pre-training dataset distributions, where the strengths of different models may vary from each other and need distinct datasets to adapt to the same downstream task. In that case, optimal dataset weights across models would be impossible for small-sized dataset settings, leaving model-dependent reweighting be the only choice.

6 Conclusion

In this paper, we propose ScaleBiO, the first bilevel optimization instantiation that is capable of scaling to over 30B-sized LLMs on data reweighting tasks. Theoretically, ScaleBiO ensures optimality of the learned data weights and enjoys the same convergence guarantees as conventional first-order penalty-based bilevel optimization algorithms on smooth and strongly convex objectives. Empirically, ScaleBiO enables data reweighting on >30B-sized models, bringing forth an efficient data filtering and selection pipeline for improving model performance on various downstream tasks.

Limitations

The proposed algorithm of ScaleBiO has yet to be verified in large-scale pre-training settings, where a huge amount of computation resources are required for conducting such experiments. We hope the success of ScaleBiO in large-scale fine-tuning settings can be the first step towards this direction.

The potential risks of ScaleBiO are the same as other data reweighting techniques, where optimizing the sampling weights on a single loss metric may lead to models that neglect other aspects, such as safety or ethics. In that case, multi-objective losses and post-training alignments are highly recommended to compensate for this deficiency.

The positive aspect of ScaleBiO is that it helps reweight data more effectively, thus allowing the training cost of large language models to be further reduced.

Ethical Considerations

In conducting our experiments on a diverse set of datasets for instruction following, we have given careful consideration to ethical concerns that may arise. Our work involves datasets such as ShareGPT, OpenOrca, WildChat, AlpacaChat, LMSYS-Chat, Airoboros, etc. We list the license for each dataset in the Appendix and ensure compliance with the licensing agreements for each dataset. Furthermore, all these data sources are publicly available and do not involve privacy issues.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. 2016. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, 29.

Yoshua Bengio. 2000. Gradient-based optimization of hyperparameters. *Neural computation*, 12(8):1889–1900.

Ning Bian, Hongyu Lin, Yaojie Lu, Xianpei Han, Le Sun, and Ben He. 2023. Chatalpaca: A multi-turn dialogue corpus based on alpaca instructions. https://github.com/cascip/ChatAlpaca.

Lesi Chen, Yaohua Ma, and Jingzhao Zhang. 2023. Near-optimal nonconvex-strongly-convex bilevel optimization with fully first-order oracles. *arXiv preprint arXiv:2306.14853*.

Tianyi Chen, Yuejiao Sun, Quan Xiao, and Wotao Yin. 2022. A single-timescale method for stochastic bilevel optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2466–2488. PMLR.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. Blog post.

Sang Keun Choe, Sanket Vaibhav Mehta, Hwijeen Ahn, Willie Neiswanger, Pengtao Xie, Emma Strubell, and Eric P. Xing. 2023. Making scalable meta learning practical. *ArXiv*, abs/2310.05674.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. 2022. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. *Advances in Neural Information Processing Systems*, 35:26698–26710.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li,

Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

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

Justin Domke. 2012. Generic methods for optimization-based modeling. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 318–326, La Palma, Canary Islands. PMLR.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022a. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022b. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the*

60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 320–335.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Jon Durbin. 2023. Airoboros: using large language models to fine-tune large language models. https://huggingface.co/datasets/jondurbin/airoboros-3.2.

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. 2017. Forward and reverse gradient-based hyperparameter optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1165–1173. PMLR.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. 2018. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577. PMLR.

Saeed Ghadimi and Mengdi Wang. 2018. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*.

David Grangier, Pierre Ablin, and Awni Hannun. 2024. Bilevel optimization to learn training distributions for language modeling under domain shift. In *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*.

Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. 2020. On the iteration complexity of hypergradient computation. In *International Conference on Machine Learning*, pages 3748–3758. PMLR.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. 2023. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021c. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. 2020. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170*.

Zijian Hu, Jipeng Zhang, Rui Pan, Zhaozhuo Xu, Shanshan Han, Han Jin, Alay Dilipbhai Shah, Dimitris Stripelis, Yuhang Yao, Salman Avestimehr, et al. 2024. Fox-1 technical report. *arXiv preprint arXiv:2411.05281*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Saachi Jain, Kimia Hamidieh, Kristian Georgiev, Andrew Ilyas, Marzyeh Ghassemi, and Aleksander Madry. 2024. Data debiasing with datamodels (d3m): Improving subgroup robustness via data selection. *arXiv preprint arXiv:2406.16846*.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. 2021. Bilevel optimization: Convergence analysis and enhanced design. In *International conference on machine learning*, pages 4882–4892. PMLR.

Nocedal Jorge and J Wright Stephen. 2006. Numerical optimization.

Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. 2021. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. *Advances in neural information processing systems*, 34:30271–30283.

Diederik P Kingma and Jimmy Lei Ba. 2015. ADAM: A method for stochastic optimization. In *International Conference on Learning Representations*.

Vijay Konda and John Tsitsiklis. 1999. Actor-critic algorithms. *Advances in neural information processing systems*, 12.

Jeongyeol Kwon, Dohyun Kwon, Stephen Wright, and Robert D Nowak. 2023. A fully first-order method for stochastic bilevel optimization. In *International Conference on Machine Learning*, pages 18083–18113. PMLR.

Wing Lian, Guan Wang, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Slimorca: An open dataset of gpt-4 augmented flan reasoning traces, with verification.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Bo Liu, Mao Ye, Stephen Wright, Peter Stone, and Qiang Liu. 2022. Bome! bilevel optimization made easy: A simple first-order approach. In *Advances in neural information processing systems*, volume 35, pages 17248–17262.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. 2020. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pages 1540–1552. PMLR.

I Loshchilov. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Zhaosong Lu and Sanyou Mei. 2023. First-order penalty methods for bilevel optimization. *arXiv preprint arXiv:2301.01716*.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. 2015. Gradient-based hyperparameter optimization through reversible learning. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2113–2122, Lille, France. PMLR.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, et al. 2022a. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, pages 15630–15649. PMLR.

Sören Mindermann, Jan M Brauner, Muhammed T Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N Gomez, Adrien Morisot, Sebastian Farquhar, et al. 2022b. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, pages 15630–15649. PMLR.

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. 2024. Orca-math: Unlocking the potential of slms in grade school math. *Preprint*, arXiv:2402.14830.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *Preprint*, arXiv:2306.02707.

Yu Nesterov. 2012. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362.

Rui Pan, Xiang Liu, Shizhe Diao, Renjie Pi, Jipeng Zhang, Chi Han, and Tong Zhang. 2024. Lisa: Layerwise importance sampling for memory-efficient large language model fine-tuning. *arXiv preprint arXiv:2403.17919*.

Fabian Pedregosa. 2016. Hyperparameter optimization with approximate gradient. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 737–746, New York, New York, USA. PMLR.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. 2019. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32.

Yuji Roh, Kangwook Lee, Steven Whang, and Changho Suh. 2021. Sample selection for fair and robust training. *Advances in Neural Information Processing Systems*, 34:815–827.

Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. 2020. Fairbatch: Batch selection for model fairness. *arXiv preprint arXiv:2012.01696*.

Yuji Roh, Weili Nie, De-An Huang, Steven Euijong Whang, Arash Vahdat, and Anima Anandkumar. 2023. Dr-fairness: Dynamic data ratio adjustment for fair training on real and generated data.

Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. 2019. Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1723–1732. PMLR.

Qianli Shen, Yezhen Wang, Zhouhao Yang, Xiang Li, Haonan Wang, Yang Zhang, Jonathan Scarlett, Zhanxing Zhu, and Kenji Kawaguchi. 2024. Memory-efficient gradient unrolling for large-scale bi-level optimization. *arXiv preprint arXiv:2406.14095*.

Sai Ashish Somayajula, Lifeng Jin, Linfeng Song, Haitao Mi, and Dong Yu. 2023. Bi-level finetuning with task-dependent similarity structure for low-resource training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8569–8588.

Daouda Sow, Kaiyi Ji, and Yingbin Liang. 2022. On the convergence theory for hessian-free bilevel algorithms. In *Advances in Neural Information Processing Systems*, volume 35, pages 4136–4149.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

"Teknium". 2023. Gpteacher general-instruct. https://huggingface.co/datasets/teknium/GPTeacher-General-Instruct.

Megh Thakkar, Tolga Bolukbasi, Sriram Ganapathy, Shikhar Vashishth, Sarath Chandar, and Partha Talukdar. 2023. Self-influence guided data reweighting for language model pre-training. *arXiv preprint arXiv:2311.00913*.

Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. 2024. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. 2024. How far can camels go? exploring the state of instruction tuning on open resources. *Advances in Neural Information Processing Systems*, 36.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2024a. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *The Twelfth International Conference on Learning Representations*.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024b. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. 2024. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. 2023. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36:34201–34227.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. 2024. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *arXiv preprint arXiv:2406.08464*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024a. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024b. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Junjie Yang, Kaiyi Ji, and Yingbin Liang. 2021. Provably faster algorithms for bilevel optimization. *Advances in Neural Information Processing Systems*, 34:13670–13682.

Yifan Yang, Peiyao Xiao, and Kaiyi Ji. 2023. Achieving $\mathcal{O}(\epsilon^{-1.5})$ complexity in Hessian/Jacobian-free stochastic bilevel optimization. *Advances in Neural Information Processing Systems*, 36.

Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, Zhenghao Liu, Bowen Zhou, Hao Peng, Zhiyuan Liu, and Maosong Sun. 2024. Advancing llm reasoning generalists with preference trees. *Preprint*, arXiv:2404.02078.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. 2023. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric. P Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2023a. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *Preprint*, arXiv:2309.11998.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

A Additional Experiments

A.1 Data Weights across Model Families

Table 7 shows the learned data weights from ScaleBiO for different backbone models under the instruction following setting.

Llama-3-8B		Llama-3-13B[1]		Qwen-2-7B		Gemma-2-9B		GPT-NeoX-20B		Yi-34B	
source	weight	source	weight	source	weight	source	weight	source	weight	source	weight
WildChat	0.711	WildChat	0.711	SlimOrca	0.945	Alpaca-pt	0.198	Airoboros	0.986	ShareGPT4	0.627
Airoboros	0.154	ShareGPT4	0.137	LMSYS-Chat	0.008	Alpaca-ko	0.180	ShareGPT4	0.005	Airoboros	0.111
ChatAlpaca	0.119	ChatAlpaca	0.021	ShareGPT4	0.004	Alpaca-it	0.080	ChatAlpaca	0.003	WildChat	0.105
Total	0.984	Total	0.869	Total	0.957	Total	0.458	Total	0.994	Total	0.843

Table 7: Data sources with **top-3** weights for LLaMA-3-8B, LLaMA-3-13B, Qwen-2-7B, Gemma-2-9B, GPT-NeoX-20B and Yi-34B in **Instruction Following** tasks.

Llama-3-8B		Qwen-2-7B		Gemma-2-9B	
source	weight	source	weight	source	weight
TIGER-Lab/MathInstruct	0.131	TIGER-Lab/MathInstruct	0.132	TIGER-Lab/MathInstruct	0.121
teknium/GPT4-LLM-Cleaned	0.119	ise-uiuc/Magicoder-Evol-Instruct-110K	0.125	DART-Math	0.114
anon8231489123/ShareGPT_Vicuna_unfiltered	0.107	teknium/GPTeacher-General-Instruct	0.102	openbmb/UltraInteract_sft	0.110
Total	0.357	Total	0.359	Total	0.345

Table 8: Data sources with **top-3** weights for LLaMA-3-8B, Qwen-2-7B, Gemma-2-9B in **Mathematical Reasoning** tasks.

A.2 Mathematical Reasoning: Stronger Benchmarks

We conducted additional experiments on mathematical reasoning using stronger benchmarks and a smaller but higher-quality dataset.

Setup Specifically, we collect 8K prompts uniformly from DART-Math (Tong et al., 2024), Ultra-Interact (Yuan et al., 2024), MathInstruct (Yue et al., 2023), and Orca-Math (Mitra et al., 2024). We then use Deepseek-R1 (DeepSeek-AI et al., 2025) to generate responses with thinking paths to construct question-answer pairs. After obtaining the data, we select 4K training samples using the ScaleBiO method alongside baseline methods and fine-tune the DeepSeek-R1-Distill-Qwen-1.5B model (DeepSeek-AI et al., 2025). The fine-tuned models are then evaluated on the reference sets of AIME24, AIME25, and AIMO25, which contain 30, 30, and 10 questions, respectively.

Method	Method (pass@1)			Method (cons@64)		
	AIME 2024	**AIME 2025**	**AIMO 2025**	**AIME 2024**	**AIME 2025**	**AIMO 2025**
Uniform	26.7	20.0	10.0	**33.3**	33.3	**30.0**
LESS	26.7	20.0	10.0	**33.3**	**36.7**	**30.0**
RHO-LOSS	30.0	20.0	**20.0**	**33.3**	33.3	**30.0**
ScaleBiO	**33.3**	**26.7**	**20.0**	**33.3**	**36.7**	**30.0**

Table 9: Comparison of methods on AIME2024, AIME2025, and AIMO2025 datasets under pass@1 and cons@64 metrics for DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI et al., 2025).

Results As shown in Table 9, ScaleBiO consistently outperforms the baseline methods across the three benchmarks under the **pass@1** metric. For the **Cons@64** accuracy, ScaleBiO achieves performance

---

[1] https://huggingface.co/Replete-AI/Llama-3-13B

comparable to the baselines. We conjecture that the narrowing gap is due to the limited diversity of the small-sized dataset, which we expect to improve with the inclusion of a larger amount of data. In summary, ScaleBiO demonstrates stable and competitive performance on challenging benchmarks across different evaluation metrics, highlighting the effectiveness of our data selection method.

B Experimental Details

B.1 Small Scale Experiments

Throughout our small-scale experiments, we use GPT-2 (Radford et al., 2019) with 124 million parameters as the backbone model. For bilevel optimization hyperparameters, we set the learning rate to 10^{-2} for sampling weights λ and 10^{-5} for models u, w. We run our algorithm for 3 epochs with a batch size of 64 and alpha of 10 while adopting AdamW (Loshchilov and Hutter, 2017) for optimization.

B.2 Large Scale Experiments

Datasets	#Samples	Kind	License
AlpacaGPT4 (Peng et al., 2023)	52K	Instruction	Apache-2.0
ShareGPT4 (Chiang et al., 2023)	6K	Conversation	Apache-2.0
SlimOrca (Lian et al., 2023)	518K	Instruction	MIT
AlpacaChat (Bian et al., 2023)	20K	Conversation	Apache-2.0
OpenOrcaGPT4 (Mukherjee et al., 2023)	1M	Instruction	MIT
WildChat (Zhao et al., 2024)	1M	Conversation	AI2 ImpACT
LMSYS-Chat (Zheng et al., 2023a)	1M	Conversation	LMSYS-Chat-1M
GPTeacher ("Teknium", 2023)	89K	Instruction	MIT
Airoboros (Durbin, 2023)	59K	Conversation	CC-BY-4.0
Alpaca-es[2]	52K	Instruction	CC-BY-4.0
Alpaca-de[3]	50K	Instruction	Apache-2.0
Alpaca-ja[4]	52K	Instruction	CC-BY-NC-SA-4.0
Alpaca-ko[5]	50K	Instruction	CC-BY-NC-4.0
Alpaca-ru[6]	30K	Instruction	CC-BY-4.0
Alpaca-it[7]	52K	Instruction	CC-BY-NC-SA-4.0
Alpaca-fr[8]	55K	Instruction	Apache-2.0
Alpaca-zh[9]	49K	Instruction	CC-BY-4.0
Alpaca-pt[10]	52K	Instruction	CC-BY-NC-4.0

Table 10: Training data sources for the **Instruction Following** task.

---

[2] https://huggingface.co/datasets/bertin-project/alpaca-spanish
[3] https://huggingface.co/datasets/mayflowergmbh/alpaca-gpt4_de
[4] https://huggingface.co/datasets/fujiki/japanese_alpaca_data
[5] https://huggingface.co/datasets/Bingsu/ko_alpaca_data
[6] https://huggingface.co/datasets/IlyaGusev/ru_turbo_alpaca
[7] https://huggingface.co/datasets/mchl-labs/stambecco_data_it
[8] https://huggingface.co/datasets/jpacifico/French-Alpaca-dataset-Instruct-55K
[9] https://huggingface.co/datasets/llm-wizard/alpaca-gpt4-data-zh
[10] https://huggingface.co/datasets/dominguesm/alpaca-data-pt-br

Instruction Following   Our training data consists of 18 distinct sources as detailed in Table 10. We collect 9 high-quality datasets and 9 multilingual Alpaca datasets which serve as irrelevant data sources. For each data source, we preprocess by filtering out conversations/instructions that exceed the max length (1024 tokens in our experiments). For our reference dataset $\mathcal{D}_{\text{val}}$ that corresponds to loss L_1, we prompt GPT4 using the prompt

"Help me generate 3 sets of 2-turn instructions to evaluate the {category} ability of LLMs. The instructions for the second turn need to be highly relevant to the first turn. The following is an example.\n\n\n EXAMPLE:{example}\n TURN1:{turn1}\n TURN2:{turn2}\n".

Here *{category}* represents one of the 8 categories in MT-Bench and *{example}* is one example from MT-Bench. In this way, we obtain a reference dataset with 1,200 samples with a similar distribution to MT-Bench. Furthermore, additional 600 samples generated in similar fashions are adopted for hyperparameter tuning for all methods.

Concerning the data reweighting and training process, we first sample 3,000 data from each source for reweighting. Then we sample 10,000 data according to the weights at the end of bilevel optimization to train the backbone model.

For ScaleBiO, the data reweighting process lasts for 3 epochs with α equals to 100 and initial learning rate 10^{-2} for weights λ. The learning rates of models u, w are set to be the same and searched in range $\{10^{-6}, 2 \times 10^{-6}, 3 \times 10^{-6}, 4 \times 10^{-6}, 5 \times 10^{-6}, 6 \times 10^{-6}, 8 \times 10^{-6}, 10^{-5}\}$. For all the fine-tuning processes, we train the LLM for 1 epoch with an initial learning rate of 8×10^{-6} and a global batch size of 64. Throughout our experiments, we adopt randomized coordinate descent with AdamW (Pan et al., 2024) and bfloat16 precision for efficient training and inference. Our experiments are conducted on 8 NVIDIA H100 80GB GPUs, where the total computational cost is around ~6K GPU hours. The multi-GPU feature of ScaleBiO is enabled by Pytorch's FSDP (Zhao et al., 2023).

For baselines, all of them are free to utilize the additional 1,800 MT-Bench-styled samples to ensure a fair comparison with ScaleBiO, where

- Uniform Weighting: directly sample 10,000 / 18 ≈ 5556 samples from each source, along with the additional MT-styled 1,200 samples to conduct supervised fine-tuning.

- LESS: we stick to settings in its original paper (Xia et al., 2024b), which adopts a warm-up training setup of learning rate 10^{-5}, batch size 32, maximum sequence length of 1024, number of epochs 4, optimizer Adam with linear decay learning rate schedule. The LoRA setup is also similar, with $r = 128$, $\alpha = 512$, dropout = 0.1.

- RHO-LOSS: a training setup of learning rate 10^{-5}, batch size 32, maximum sequence length 1024, number of epochs 1, optimizer of Adam with cosine decay learning rate schedule. Here the same reference model Qwen-2-1.5B is employed for different settings, which according to the original paper (Mindermann et al., 2022a), is fine given the algorithm's non-sensitiveness to the reference model.

Mathematical Reasoning   The validation set comes from the validation sets (if available) or training sets (if validation is not available) from the validation sources presented in Table 11, where 280 samples are

---

Datasets	#Samples	Kind	License
DART-Math[11] (Tong et al., 2024)	591K	Math	MIT
SlimOrca[12] (Lian et al., 2023)	518K	Instruction	MIT
openbmb/UltraInteract_sft[13] (Yuan et al., 2024)	289K	Reasoning	MIT
TIGER-Lab/MathInstruct[14] (Yue et al., 2023)	262K	Reasoning	MIT
microsoft/orca-math-word-problems-200k[15] (Mitra et al., 2024)	200K	Math	MIT
WizardLMTeam/WizardLM_evol_instruct_V2_196k[16]	196K	Instruction	MIT
ise-uiuc/Magicoder-Evol-Instruct-110K[17]	110K	Coding	Apache-2.0
anon8231489123/ShareGPT_Vicuna_unfiltered[18]	94K	Instruction	Apache-2.0
teknium/GPTeacher-General-Instruct[19]	89K	Instruction	MIT
teknium/GPT4-LLM-Cleaned[20]	55K	Instruction	Apache-2.0
GSM8K (Cobbe et al., 2021)	7.5K	Math	MIT
Competition Math[21] (Hendrycks et al., 2021c)	12.5K	Math	MIT
bigcode/self-oss-instruct-sc2-exec-filter-50k[22]	50.7K	Coding	ODC-By
cais/mmlu[23] (Hendrycks et al., 2020, 2021a)	116K	Science	MIT
ARC-Easy (Clark et al., 2018)	5.2K	Instruction	CC-BY-SA-4.0
ARC-Challenge (Clark et al., 2018)	2.6K	Instruction	CC-BY-SA-4.0

Table 11: Training (above the line) and validation data sources (below the line) for the **Mathematical Reasoning** task.

randomly chosen from each source and form a validation set with $280 \times 6 = 1680$ samples. For ScaleBiO, the dataset is proportionally split into two sets with $1,400$ samples and 280 samples individually, where the former is treated as the $\mathcal{D}_{\text{val}}$ for L_1 in reweighting and the latter is adopted for hyperparameter tuning. The whole validation set is available to other baselines. Other settings and statistics remain the same as the instruction-following task.

C Important Lemmas

Suppose Assumption 1 hold, the functions $\mathcal{L}^{\alpha}(\lambda, w, u)$ and $\Gamma^{\alpha}(\lambda)$ satisfy the following properties.

Lemma 2. *Under Assumption 1, the followings hold:*

(i) $\mathcal{L}^{\alpha}(\lambda, w, u)$ *is* $\mu_2 \alpha$*-strongly concave w.r.t.* u*;*

(ii) $\mathcal{L}^{\alpha}(\lambda, w, u)$ *is* $\mu_2 \alpha / 2$*-strongly convex w.r.t.* w *if* $\alpha > 2\ell_{11}/\mu_2$*.*

The results of Lemma 2 can be found in (Kwon et al., 2023) and Lemma B.1 of (Chen et al., 2023). From Lemma B.7 in (Chen et al., 2023), the following result holds for $\Gamma^{\alpha}(\lambda)$:

Lemma 3. *Under Assumption 1, if* $\alpha > 2\ell_{11}/\mu_2$*, then* $\Gamma^{\alpha}(\lambda)$ *is* ℓ_{Γ}*-smooth, where* $\ell_{\Gamma} = \mathcal{O}(\kappa^3)$ *is a constant that is independent on* α*.*

Moreover, the functions $w_*^{\alpha}(\lambda)$ and $u_*(\lambda)$ satisfy the following properties.

Lemma 4. *Under Assumption 1, we have*

$$\|w_*^{\alpha}(\lambda) - w_*(\lambda)\| \leq \frac{C_0}{\alpha}$$

where $C_0 = \ell_{10}/\mu_2$*.*

The result in Lemma 4 follows from Lemma B.2 of (Chen et al., 2023).

Lemma 5. *Under Assumption 1, if* $\alpha > 2\ell_{11}/\mu_2$*, then we have*

(i) $u_*(\lambda)$ *is κ-Lipschitz continuous;*

(ii) $w_*^\alpha(\lambda)$ *is $\ell_{u_*,0}$-Lipschitz continuous where $\ell_{u_*,0} = 3\kappa$.*

where the condition number $\kappa = \max\{\ell_{10}, \ell_{11}, \ell_{21}, \ell_{22}\}/\mu_2$

Claim (i) in Lemma 5 can be found in Lemma 2.2 of (Ghadimi and Wang, 2018) and Claim (ii) implies from Lemma 3.2 (setting $\lambda_1 = \lambda_2$) of (Kwon et al., 2023).

Lemma 6. *Under Assumption 1, if $\alpha > 2\ell_{f,1}/\mu_g$, then $u_*(\lambda)$ is $\ell_{\nabla u_*}$-smooth where $\ell_{\nabla_*} = \mathcal{O}\left(\frac{\kappa^2}{\mu_2}(\ell_{21}+1)\right)$ where the condition number $\kappa = \max\{\ell_{10}, \ell_{11}, \ell_{21}, \ell_{22}\}/\mu_2$*

Following Lemma A.3 of (Kwon et al., 2023) and recalling the Lipschitz continuous property of $u_*(\lambda)$ from Lemma 5, we have this claim is correct.

D   Proofs of Theorem 1

Proof. We sample the function $\mathcal{L}^\alpha$ by the following mini-batch approximation $\mathcal{L}_{D_k}^\alpha$ per iteration:

$$\mathcal{L}_{D_k}^\alpha := L_1(\lambda, w; D_{\mathrm{val}}^k) + \alpha\left(L_2(\lambda, w; D_{\mathrm{tr}}^k) - L_2(\lambda, u; D_{\mathrm{tr}}^k)\right) \tag{14}$$

where D_{tr}^k is i.i.d. from the training dataset D_{train}, D_{val}^k are i.i.d. from the validation dataset D_{val} and independent with D_{train}. We use $\mathcal{F}_k$ to denote the random information before the iteration (λ_k, w_k, u_k), that is $\mathcal{F}_k := \sigma\left(\{(\lambda_k, \omega_k, u_k), D_{k-1}, \cdots, D_1\}\right)$. We use $\mathcal{C}_k = \sigma\left(\{j_1, j_2 \cdots, j_{t-1}; r_1, r_2, \cdots, r_{t-1}\}\right)$ to denote the random information of variables u, w for the randomized block coordinates before the iteration k.

We recall the iterating formula of λ in the stochastic version of the minimax algorithm that $\lambda_{k+1} - \lambda_k = -\eta_\lambda \nabla_\lambda \mathcal{L}_{D_k}^\alpha(\lambda_k, w_k, u_k)$. At each iteration,

$$\mathbb{E}[\nabla\mathcal{L}_{D_k}^\alpha(\lambda_k, \omega_k, u_k) \mid \mathcal{F}_k] = \nabla\mathcal{L}^\alpha(\lambda_k, \omega_k, u_k). \tag{15}$$

By the smoothness of Γ^α (see Lemma 3), we have

$$\Gamma^\alpha(\lambda_{k+1}) \le \Gamma^\alpha(\lambda_k) + \langle\nabla\Gamma^\alpha(\lambda_k), \lambda_{k+1} - \lambda_k\rangle + \frac{\ell_\Gamma}{2}\|\lambda_{k+1} - \lambda_k\|^2$$

$$= \Gamma^\alpha(\lambda_k) - \eta_\lambda\langle\nabla\Gamma^\alpha(\lambda_k), \nabla_\lambda L_{D_k}^\alpha(\lambda_k, \omega_k, u_k)\rangle + \frac{\ell_\Gamma\eta_\lambda^2}{2}\left\|\nabla_\lambda\mathcal{L}_{D_k}^\alpha(\lambda_k, \omega_k, u_k)\right\|^2. \tag{16}$$

Taking conditional expectation w.r.t. $\mathcal{F}_k, \mathcal{C}_k$ on the above inequality, we have

$$\mathbb{E}[\Gamma^\alpha(\lambda_{k+1}) \mid \mathcal{F}_k, \mathcal{C}_k]$$

$$\le \Gamma^\alpha(\lambda_k) - \eta_\lambda\langle\nabla\Gamma^\alpha(\lambda_k), \mathbb{E}[\nabla_\lambda\mathcal{L}_{D_k}^\alpha(\lambda_k, \omega_k, u_k) \mid \mathcal{F}_k, \mathcal{C}_k]\rangle + \frac{\ell_\Gamma\eta_\lambda^2}{2}\mathbb{E}\left[\left\|\nabla_\lambda\mathcal{L}_{D_k}^\alpha(\lambda_k, \omega_k, u_k)\right\|^2 \mid \mathcal{F}_k, \mathcal{C}_k\right]$$

$$\le \Gamma^\alpha(\lambda_k) - \eta_\lambda\langle\nabla\Gamma^\alpha(\lambda_k), \nabla_\lambda\mathcal{L}^\alpha(\lambda_k, \omega_k, u_k)\rangle + \frac{\ell_\Gamma\eta_\lambda^2}{2}\mathbb{E}\left[\left\|\nabla_\lambda\mathcal{L}_{D_k}^\alpha(\lambda_k, \omega_k, u_k)\right\|^2 \mid \mathcal{F}_k, \mathcal{C}_k\right] \tag{17}$$

where the inequality follows the fact that $\mathcal{L}_{D_k}^\alpha$ is an unbiased estimation of $\mathcal{L}^\alpha$ and

$$\mathbb{E}\left[\left\|\nabla_\lambda\mathcal{L}_{D_k}^\alpha(\lambda_k, \omega_k, u_k)\right\|^2 \mid \mathcal{F}_k, \mathcal{C}_k\right]$$

$$= \mathbb{E}\left[\left\|\nabla_\lambda\mathcal{L}_{D_k}^\alpha(\lambda_k, \omega_k, u_k) - \nabla_\lambda\mathcal{L}^\alpha(\lambda_k, \omega_k, u_k) + \nabla_\lambda\mathcal{L}^\alpha(\lambda_k, \omega_k, u_k)\right\|^2 \mid \mathcal{F}_k\right]$$

$$\le \mathbb{E}\left[\left\|\nabla_\lambda\mathcal{L}_{D_k}^\alpha(\lambda_k, \omega_k, u_k) - \nabla_\lambda\mathcal{L}^\alpha(\lambda_k, \omega_k, u_k)\right\|^2 \mid \mathcal{F}_k\right] + \left\|\nabla_\lambda\mathcal{L}^\alpha(\lambda_k, \omega_k, u_k)\right\|^2$$

$$\le \frac{\sigma_1^2 + 2\alpha^2\sigma_2^2}{B} + \left\|\nabla_\lambda\mathcal{L}^\alpha(\lambda_k, \omega_k, u_k)\right\|^2 \tag{18}$$

where the variance of the minibatch stochastic gradients (with batch size B) is bounded

$$\mathbb{E}\left[\left\|\nabla L_1(\lambda, w; D_{\text{val}}^k) - \nabla L_1(\lambda, w)\right\|^2\right] \leq \frac{\sigma_1^2}{B}, \quad \mathbb{E}\left[\left\|\nabla L_2(\lambda, w; D_{\text{tr}}^k) - \nabla L_2(\lambda, w)\right\|^2\right] \leq \frac{\sigma_2^2}{B}, \quad (19)$$

then

$$\mathbb{E}\left[\left\|\nabla_\lambda \mathcal{L}_{D_k}^\alpha(\lambda_k, \omega_k, u_k) - \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, \omega_k, u_k)\right\|^2 \mid \mathcal{F}_k\right]$$

$$= \mathbb{E}\left[\left\|\nabla_\lambda L_1(\lambda_k, w_k; D_{\text{val}}^k) - \nabla_\lambda L_1(\lambda_k, w_k)\right\|^2 + \alpha^2 \left\|\nabla_\lambda L_2(\lambda_k, w_k; D_{\text{tr}}^k) - \nabla_\lambda L_2(\lambda_k, w_k)\right\|^2 \mid \mathcal{F}_k\right]$$

$$+ \alpha^2 \mathbb{E}\left[\left\|\nabla_\lambda L_2(\lambda_k, u_k; D_{\text{tr}}^k) - \nabla_\lambda L_2(\lambda_k, u_k)\right\|^2 \mid \mathcal{F}_k\right]$$

$$\leq \frac{\sigma_1^2 + 2\alpha^2 \sigma_2^2}{B}. \tag{20}$$

Applying the above results, we have

$$\mathbb{E}[\Gamma^\alpha(\lambda_{k+1}) \mid \mathcal{F}_k, \mathcal{C}_k] \leq \Gamma^\alpha(\lambda_k) - \eta_\lambda \left\langle \nabla\Gamma^\alpha(\lambda_k), \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, \omega_k, u_k)\right\rangle + \frac{\ell_\Gamma \eta_\lambda^2}{2}\left\|\nabla_\lambda \mathcal{L}^\alpha(\lambda_k, \omega_k, u_k)\right\|^2$$

$$+ \frac{\ell_\Gamma \eta_\lambda^2}{2B}\left(\sigma_1^2 + 2\alpha^2 \sigma_2^2\right). \tag{21}$$

Let $\delta_k = \|u_k - u_*(\lambda_k)\|^2$ and $r_k = \|w_k - w_*^\alpha(\lambda_k)\|^2$. The inner product term of RHS of (21) is estimated as follows:

$$- \left\langle \nabla\Gamma^\alpha(\lambda_k), \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, \omega_k, u_k)\right\rangle$$

$$= - \left\langle \nabla\Gamma^\alpha(\lambda_k), \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, \omega_k, u_k) - \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_*(\lambda_k))\right\rangle$$

$$- \left\langle \nabla\Gamma^\alpha(\lambda_k), \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_*(\lambda_k)) - \nabla_\lambda \Phi^\alpha(w_*^\alpha(\lambda_k), \lambda_k) + \nabla_\lambda \Phi^\alpha(w_*^\alpha(\lambda_k), \lambda_k)\right\rangle$$

$$\overset{(a)}{=} - \left\langle \nabla\Gamma^\alpha(\lambda_k), \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_k) - \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_*(\lambda_k))\right\rangle$$

$$- \left\langle \nabla\Gamma^\alpha(\lambda_k), \nabla_\lambda \Phi^\alpha(\lambda_k, w_k) - \nabla_\lambda \Phi^\alpha(\lambda_k, w_*^\alpha(\lambda_k)) + \nabla\Gamma^\alpha(\lambda_k)\right\rangle$$

$$= - \|\nabla\Gamma^\alpha(\lambda_k)\|^2 - \left\langle \nabla\Gamma^\alpha(\lambda_k), \nabla_\lambda \mathcal{L}^\alpha(u_k, \omega_k, \lambda_k) - \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_*(\lambda_k))\right\rangle$$

$$- \left\langle \nabla\Gamma^\alpha(\lambda_k), \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_*(\lambda_k)) - \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_*^\alpha(\lambda_k), u_*(\lambda_k))\right\rangle$$

$$\overset{(b)}{\leq} - \frac{1}{2}\|\nabla\Gamma^\alpha(\lambda_k)\|^2 + \|\nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_k) - \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_*(\lambda_k))\|^2$$

$$+ \|\nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_*(\lambda_k)) - \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_*^\alpha(\lambda_k), u_*(\lambda_k))\|^2$$

$$\overset{(c)}{\leq} - \frac{1}{2}\|\nabla\Gamma^\alpha(\lambda_k)\|^2 + \alpha^2 \ell_{21}^2 \|u_k - u_*(\lambda_k)\|^2 + 2\left(\ell_{11}^2 + \alpha^2 \ell_{21}^2\right)\|\omega_k - \omega_*^\alpha(\lambda_k)\|^2$$

$$= - \frac{1}{2}\|\nabla\Gamma^\alpha(\lambda_k)\|^2 + \alpha^2 \ell_{21}^2 \delta_k + 2\left(\ell_{11}^2 + \alpha^2 \ell_{21}^2\right) r_k \tag{22}$$

where (a) uses the optimality of Φ over w that $\nabla_\lambda \Phi^\alpha(w_*^\alpha(\lambda_k), \lambda_k) = \nabla\Gamma^\alpha(\lambda_k) = \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_*^\alpha(\lambda_k), u_*(\lambda_k))$, (b) follows from the Cauchy-Schwartz inequality and (c) uses the smoothness of L_1 and L_2. Next we turn to estimate the norm of gradient $\nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_k)$ as follows

$$\|\nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_k)\|^2 = \|\nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_k) - \nabla\Gamma^\alpha(\lambda_k) + \nabla\Gamma^\alpha(\lambda_k)\|^2$$

$$\leq 2\left(\|\nabla\Gamma^\alpha(\lambda_k)\|^2 + \|\nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_k) - \nabla\Gamma^\alpha(\lambda_k)\|^2\right)$$

$$\leq 2\|\nabla\Gamma^\alpha(\lambda_k)\|^2 + 4\|\nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_k) - \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_*(\lambda_k))\|^2$$

$$+ 4\|\nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_k, u_*(\lambda_k)) - \nabla_\lambda \mathcal{L}^\alpha(\lambda_k, w_*^\alpha(\lambda_k), u_*(\lambda_k))\|^2$$

$$\overset{(a)}{\leq} 2\|\nabla\Gamma^\alpha(\lambda_k)\|^2 + 4\alpha^2 \ell_{11}^2 \|u_k - u_*(\lambda_k)\|^2 + 8\left(\ell_{11}^2 + \alpha^2 \ell_{21}^2\right)\|\omega_k - \omega_*^\alpha(\lambda_k)\|^2$$

$$= 2\|\nabla\Gamma(\lambda_k)\|^2 + 4\alpha^2 \ell_{11}^2 \delta_k + 8\left(\ell_{11}^2 + \alpha^2 \ell_{21}^2\right) r_k \tag{23}$$

where (a) uses the smoothness of objectives L_1, L_2. Incorporating the above inequalities (22) and (23) into (21) gives

$$\mathbb{E}[\Gamma^\alpha(\lambda_{k+1}) \mid \mathcal{F}_k, \mathcal{C}_k] \leq \Gamma^\alpha(\lambda_k) - \frac{\eta_\lambda}{2} \|\nabla\Gamma^\alpha(\lambda_k)\|^2 + \frac{\ell_\Gamma \eta_\lambda^2}{2} \left(2\|\nabla\Gamma^\alpha(\lambda_k)\|^2 + 4\alpha^2 \ell_{11}^2 \delta_k + 8\left(\ell_{11}^2 + \alpha^2 \ell_{21}^2\right) r_k \right)$$

$$+ \eta_\lambda \left(\alpha^2 \ell_{11}^2 \delta_k + 2\left(\ell_{11}^2 + \alpha^2 \ell_{21}^2\right) r_k \right) + \frac{\ell_\Gamma \eta_\lambda^2}{2} \left(\sigma_1^2 + 2\alpha^2 \sigma_2^2 \right). \tag{24}$$

Then, we focus on estimating δ_k and r_k. For the inner variables u, w, we use the randomized block coordinates method with total J blocks and each block is uniformly chosen. By the strong concavity of $\mathcal{L}^\alpha$ with respect to u, we first achieve the following evaluations for δ_k:

$$\mathbb{E}\left[\|u_{k+1} - u_*(\lambda_k)\|^2 \mid \mathcal{F}_k, \mathcal{C}_k \right] = \mathbb{E}\left[\|u_k - \alpha\eta_u U_{j_t}\nabla_u L_2\left(\lambda_k, u_k; \mathcal{D}_k^{\mathrm{tr}}\right) - u_*(\lambda_k)\|^2 \mid \mathcal{F}_k, \mathcal{C}_k \right]$$

$$= \|u_*(\lambda_k) - u_k\|^2 - 2\alpha\eta_u \mathbb{E}\left[\langle u_k - u_*(\lambda_k), \nabla_u L_2(\lambda_k, u_k; \mathcal{D}_k^{\mathrm{tr}}) \rangle_{j_t} \mid \mathcal{F}_k, \mathcal{C}_k \right]$$

$$+ \alpha^2 \eta_u^2 \mathbb{E}\left[\|U_{j_t}\nabla_u L_2(u_k, \lambda_k; \mathcal{D}_k^{\mathrm{tr}})\|^2 \mid \mathcal{F}_k, \mathcal{C}_k \right]$$

$$\overset{(a)}{=} \|u_*(\lambda_k) - u_k\|^2 - \frac{2\alpha\eta_u}{J} \langle u_k - u_*(\lambda_k), \nabla_u L_2(\lambda_k, u_k) \rangle + \frac{\alpha^2 \eta_u^2}{J} \mathbb{E}\left[\|\nabla_u L_2(\lambda_k, u_k; \mathcal{D}_k^{\mathrm{tr}})\|^2 \mid \mathcal{F}_k \right]$$

$$\overset{(b)}{\leq} \|u_*(\lambda_k) - u_k\|^2 - \frac{2\eta_u \alpha}{J} \left(L_2(\lambda_k, u_k) - L_2(\lambda_k, u_*(\lambda_k)) + \frac{\mu_2}{2}\|u_*(\lambda_k) - u_k\|^2 \right)$$

$$+ \frac{\alpha^2 \eta_u^2}{J} \mathbb{E}\left[\|\nabla_u L_2(\lambda_k, u_k; \mathcal{D}_k^{\mathrm{tr}})\|^2 \mid \mathcal{F}_k \right]$$

$$\overset{(c)}{=} \|u_*(\lambda_k) - u_k\|^2 - \frac{2\eta_u \alpha}{J} \left(L_2(\lambda_k, u_k) - L_2(\lambda_k, u_*(\lambda_k)) + \frac{\mu_2}{2}\|u_*(\lambda_k) - u_k\|^2 \right)$$

$$+ \frac{\alpha^2 \eta_u^2}{J} \mathbb{E}\left[\|\nabla_u L_2(\lambda_k, u_k; \mathcal{D}_k^{\mathrm{tr}}) - \nabla_u L_2(\lambda_k, u_k)\|^2 \mid \mathcal{F}_k \right] + \frac{\alpha^2 \eta_u^2}{J}\|\nabla_u L_2(\lambda_k, u_k)\|^2$$

$$\overset{(d)}{\leq} \|u_*(\lambda_k) - u_k\|^2 - \frac{2\eta_u \alpha}{J} \left(L_2(\lambda_k, u_k) - L_2(\lambda_k, u_*(\lambda_k)) + \frac{\mu_2}{2}\|u_*(\lambda_k) - u_k\|^2 \right)$$

$$+ \frac{\alpha^2 \eta_u^2}{J} \mathbb{E}\left[\|\nabla_u L_2(\lambda_k, u_k; \mathcal{D}_k^{\mathrm{tr}}) - \nabla_u L_2(\lambda_k, u_k)\|^2 \mid \mathcal{F}_k \right] + \frac{2\ell_{21}\eta_u^2 \alpha^2}{J} \left(L_2(\lambda_k, u_k) - L_2(\lambda_k, u_*(\lambda_k)) \right)$$

$$\overset{(e)}{\leq} \left(1 - \frac{\alpha\mu_2\eta_u}{J}\right) \|u_*(\lambda_k) - u_k\|^2 + \frac{\alpha^2 \eta_u^2 \sigma_2^2}{JB}. \tag{25}$$

where (a) use the truth that since the j_k block coordinate is uniformly chosen from $\{1, 2, \cdots, J\}$, we have

$$\mathbb{E}\left[\langle u_k - u_*(\lambda_k), \nabla_u L_2(\lambda_k, u_k; \mathcal{D}_k^{\mathrm{tr}}) \rangle_{j_t} \mid \mathcal{F}_k, \mathcal{C}_k \right] = \frac{1}{J}\mathbb{E}\left[\langle u_k - u_*(\lambda_k), \nabla_u L_2(\lambda_k, u_k; \mathcal{D}_k^{\mathrm{tr}}) \rangle \mid \mathcal{F}_k \right]$$

$$= \frac{1}{J} \langle u_k - u_*(\lambda_k), \nabla_u L_2(\lambda_k, u_k) \rangle \tag{26}$$

and

$$\mathbb{E}\left[\|U_{j_t}\nabla_u L_2(u_k, \lambda_k; \mathcal{D}_k^{\mathrm{tr}})\|^2 \mid \mathcal{F}_k, \mathcal{C}_k \right] = \frac{1}{J}\mathbb{E}\left[\|\nabla_u L_2(\lambda_k, u_k; \mathcal{D}_k^{\mathrm{tr}})\|^2 \mid \mathcal{F}_k \right] \tag{27}$$

(b) follows from the strong convexity of L_2 w.r.t. u which implies that

$$L_2(\lambda_k, u_*(\lambda_k)) \geq L_2(\lambda_k, u_k) + \langle \nabla_u L_2(\lambda_k, u_k), u_*(\lambda_k) - u_k \rangle + \frac{\mu_2}{2}\|u_k - u_*(\lambda_k)\|^2,$$

(c) uses the relationship $\mathbb{E}\left[\nabla_u L_2(\lambda_k, u_k; D_{\mathrm{tr}}^k) \mid \mathcal{F}_k \right] = \nabla_u L_2(\lambda_k, u_k)$ which induces that

$$\mathbb{E}\left[\|\nabla_u L_2(\lambda_k, u_k; D_k^{\mathrm{str}})\|^2 \mid \mathcal{F}_k \right] = \mathbb{E}\left[\|\nabla_u L_2(\lambda_k, u_k; \mathcal{D}_k^{\mathrm{str}}) - \nabla_u L_2(\lambda_k, u_k)\|^2 \mid \mathcal{F}_k \right] + \|\nabla_u L_2(\lambda_k, u_k)\|^2 \tag{28}$$

and (d) uses the optimality of $u_*(\lambda)$ and the smoothness of L_2 such that

$$L_2(\lambda_k, u_*(\lambda_k)) - L_2(\lambda_k, u_k) \leq L_2(\lambda_k, \tilde{u}) - L_2(\lambda_k, u_k)$$

$$\leq L_2(\lambda_k, u_k) + \langle \nabla_u L_2(\lambda_k, u_k), \tilde{u} - u_k \rangle + \frac{\ell_{21}}{2} \| \tilde{u} - u_k \|^2 - L_2(\lambda_k, u_k)$$

$$= -\frac{1}{2\ell_{21}} \| \nabla_u L_2(\lambda_k, u_k) \|^2 \tag{29}$$

where $\tilde{u} = u_k - \frac{1}{\ell_{21}} \nabla_u L_2(\lambda_k, u_k)$ and (e) uses

$$\mathbb{E}\left[\left\| \nabla_u L_2(\lambda_k, u_k; \mathcal{D}_k^{\text{str}}) - \nabla_u L_2(\lambda_k, u_k) \right\|^2 \mid \mathcal{F}_k \right] \leq \frac{\sigma_2^2}{B}. \tag{30}$$

and $\eta_u \leq 1/(\alpha \ell_{21})$. Then we make the following recursive estimation for δ_k:

$$\delta_{k+1} = \| u_*(\lambda_{k+1}) - u_{k+1} \|^2 = \| u_*(\lambda_{k+1}) - u_*(\lambda_k) + u_*(\lambda_k) - u_{k+1} \|^2$$

$$\overset{(a)}{\leq} (1 + \gamma_1) \| u_*(\lambda_{k+1}) - u_*(\lambda_k) \|^2 + (1 + 1/\gamma_1) \| u_*(\lambda_k) - u_{k+1} \|^2$$

$$\overset{(b)}{\leq} (1 + \gamma_1)\kappa^2 \| \lambda_{k+1} - \lambda_k \|^2 + (1 + 1/\gamma_1) \| u_*(\lambda_k) - u_{k+1} \|^2$$

$$\overset{(c)}{\leq} (1 + \gamma_1)\kappa^2 \| \lambda_{k+1} - \lambda_k \|^2 + (1 + 1/\gamma_1) \left(\left(1 - \frac{\alpha \mu_2 \eta_u}{J} \right) \delta_k + \frac{\alpha^2 \eta_u^2 \sigma_2^2}{JB} \right)$$

$$\overset{(d)}{\leq} (1 + \gamma_1)\kappa^2 \eta_\lambda^2 \| \nabla_\lambda \mathcal{L}^\alpha(u_k, \omega_k, \lambda_k) \|^2 + (1 + 1/\gamma_1) \left(1 - \frac{\alpha \mu_2 \eta_u}{J} \right) \delta_k + (1 + 1/\gamma_1)\frac{\alpha^2 \eta_u^2 \sigma_2^2}{JB}$$

$$\overset{(e)}{\leq} (1 + \gamma_1)\kappa^2 \eta_\lambda^2 \left(2 \| \nabla \Gamma^\alpha(\lambda_k) \|^2 + 4\alpha^2 \ell_{21}^2 \delta_k + 8 \left(\ell_{11}^2 + \alpha^2 \ell_{21}^2 \right) r_k \right) + (1 + 1/\gamma_1) \left(1 - \frac{\alpha \mu_2 \eta_u}{J} \right) \delta_k$$

$$+ (1 + 1/\gamma_1)\frac{\alpha^2 \eta_u^2 \sigma_2^2}{JB}$$

$$= \left(4\alpha^2(1 + \gamma_1)\kappa^2 \eta_\lambda^2 \ell_{21}^2 + (1 + 1/\gamma_1) \left(1 - \frac{\alpha \mu_2 \eta_u}{J} \right) \right) \delta_k + 8(1 + \gamma_1)\kappa^2 \eta_\lambda^2 \left(\ell_{11}^2 + \alpha^2 \ell_{21}^2 \right) r_k$$

$$+ 2(1 + \gamma_1)\kappa^2 \eta_\lambda^2 \| \nabla \Gamma^\alpha(\lambda_k) \|^2 + (1 + 1/\gamma_1)\frac{\alpha^2 \eta_u^2 \sigma_2^2}{JB} \tag{31}$$

where (a) follows from Cauchy-Schwartz inequality with $\gamma_1 > 0$; (b) uses the Lipschitz continuity of u_* from Lemma 5; (c) follows from the inequality (25); (d) uses the iterating formula of λ_{k+1}; (e) follows from the inequality (23).

Since $L_1 + \alpha L_2$ is strongly convex with respect to w with parameter $\alpha \mu_2/2$ if $\alpha \geq 2\ell_{21}/\mu_2$. Similar to δ_k, we can achieve the following result for r_k

$$\mathbb{E}\left[\| \omega_*^\alpha(\lambda_k) - \omega_{k+1} \|^2 \mid \mathcal{F}_k, \mathcal{C}_k \right] \leq \left(1 - \frac{\alpha \mu_2 \eta_w}{2J} \right) r_k + \frac{\eta_w^2 \left(\sigma_1^2 + \alpha^2 \sigma_2^2 \right)}{JB}. \tag{32}$$

Following the same procedure as in (31), we estimate the recursion r_k as below

$$r_{k+1} \leq (1 + \gamma_2) \| \omega_*^\alpha(\lambda_{k+1}) - \omega_*^\alpha(\lambda_k) \|^2 + (1 + \gamma_2^{-1}) \| \omega_*^\alpha(\lambda_k) - \omega_{k+1} \|^2$$

$$\leq (1 + \gamma_2)\kappa^2 \| \lambda_{k+1} - \lambda_k \|^2 + (1 + \gamma_2^{-1}) \left(\left(1 - \frac{\alpha \mu_2 \eta_w}{2J} \right) r_k + \frac{\eta_w^2 \left(\sigma_1^2 + \alpha^2 \sigma_2^2 \right)}{JB} \right)$$

$$\leq \left(4(1 + \gamma_2)\kappa^2 \eta_\lambda^2 \left(\ell_{11}^2 + \alpha^2 \ell_{21}^2 \right) + (1 + 1/\gamma_2) \left(1 - \frac{\alpha \mu_2 \eta_w}{2J} \right) \right) r_k + 8(1 + \gamma_2)\kappa^2 \eta_\lambda^2 \alpha^2 \ell_{21}^2 \delta_k$$

$$+ 2(1 + \gamma_2)\kappa^2 \eta_\lambda^2 \| \nabla \Gamma^\alpha(\lambda_k) \|^2 + (1 + 1/\gamma_2)\frac{\eta_w^2 \left(\sigma_1^2 + \alpha^2 \sigma_2^2 \right)}{JB} \tag{33}$$

where $\gamma_2 > 0$.

31979

We define the Lyapunov function

$$R_k = \Gamma^\alpha(\lambda_k) - \Gamma^\alpha_{\min} + \xi^1_k \delta_k + \xi^k_2 r_k \tag{34}$$

where $\xi^1_k, \xi^2_k > 0$ are non-increasing sequences and $\Gamma^\alpha_{\min}$ is the minimum of Γ^α. We must have $R_k \geq 0$. Incorporating the results of (24), (31), (33) gives

$$\mathbb{E}[R_{k+1} \mid \mathcal{F}_k, \mathcal{C}_k]$$

$$\leq R_k - \frac{\eta_\lambda}{2} \|\nabla\Gamma(\lambda_k)\|^2 + \frac{\ell_\Gamma \eta_\lambda^2}{2} \left(2\|\nabla\Gamma^\alpha(\lambda_k)\|^2 + 4\alpha^2\ell_{21}^2\delta_k + 8\left(\ell_{11}^2 + \alpha^2\ell_{21}^2\right)r_k\right)$$

$$+ \eta_\lambda\left(\alpha^2\ell_{21}^2\delta_k + 2\left(\ell_{11}^2 + \alpha^2\ell_{21}^2\right)r_k\right) + \frac{\ell_\Gamma\eta_\lambda^2}{2}\left(\sigma_1^2 + 2\alpha^2\sigma_2^2\right) + \left(\xi^1_{k+1}\delta_{k+1} - \xi^1_k\delta_k\right) + \left(\xi^2_{k+1}r_{k+1} - \xi^2_k r_k\right)$$

$$\leq R_k - \left(\frac{\eta_\lambda}{2} - \ell_\Gamma\eta_\lambda^2 - 2\xi^1_{k+1}(1+\gamma_1)\kappa^2\eta_\lambda^2 - 2\xi^2_{k+1}(1+\gamma_2)\kappa^2\eta_\lambda^2\right)\|\nabla\Gamma^\alpha(\lambda_k)\|^2 + \phi_1\delta_k + \phi_2 r_k$$

$$+ \frac{\ell_\Gamma\eta_\lambda^2}{2}\left(\sigma_1^2 + 2\alpha^2\sigma_2^2\right) + \xi^1_{k+1}(1+\gamma_1^{-1})\frac{\alpha^2\eta_u^2\sigma_2^2}{JB} + \xi^2_{k+1}(1+\gamma_2^{-1})\frac{\eta_w^2\left(\sigma_1^2 + \alpha^2\sigma_2^2\right)}{JB} \tag{35}$$

where

$$\phi_1 = \xi^1_{k+1}\left(4\alpha^2(1+\gamma_1)\kappa^2\eta_\lambda^2\ell_{21}^2 + (1+1/\gamma_1)\left(1 - \frac{\alpha\mu_2\eta_u}{J}\right)\right) - \xi^1_k + 2\ell_\Gamma\eta_\lambda^2\alpha^2\ell_{21}^2 + \eta_\lambda\alpha^2\ell_{21}^2$$

$$+ 8\xi^2_{k+1}(1+\gamma_2)\kappa^2\eta_\lambda^2\left(\ell_{11}^2 + \alpha^2\ell_{21}^2\right)$$

$$\phi_2 = \xi^2_{k+1}\left(4(1+\gamma_2)\kappa^2\eta_\lambda^2\left(\ell_{11}^2 + \alpha^2\ell_{21}^2\right) + (1+1/\gamma_2)\left(1 - \frac{\alpha\mu_2\eta_w}{2J}\right)\right) - \xi^2_k + 4\ell_\Gamma\eta_\lambda^2\left(\ell_{11}^2 + \alpha^2\ell_{21}^2\right)$$

$$+ 2\eta_\lambda\left(\ell_{11}^2 + \alpha^2\ell_{21}^2\right) + 8\xi^1_{k+1}(1+\gamma_1)\kappa^2\eta_\lambda^2\alpha^2\ell_{21}^2. \tag{36}$$

Let $\eta_u = \eta_\omega = \eta_0/K^a$ and $\eta_\lambda = \eta_\lambda^0/K^b$, and $\alpha = K^c$ where $0 \leq a \leq b$ and $c > 0$, and $\ell = \max\{\ell_{11}, \ell_{21}\}$ then ϕ_1 and ϕ_2 can be re-written as:

$$\phi_1 = \xi^1_{k+1}\left(\frac{4(1+\gamma_1)\kappa^2(\eta_\lambda^0)^2}{K^{2(b-c)}}\ell^2 + (1+1/\gamma_1)\left(1 - \frac{\mu_2\eta_0}{JK^{(a-c)}}\right)\right) - \xi^1_k + \frac{2\ell_\Gamma\ell^2(\eta_\lambda^0)^2}{K^{2(b-c)}} + \frac{\ell^2\eta_\lambda^0}{K^{(b-2c)}}$$

$$+ \frac{8\xi^2_{k+1}(1+\gamma_2)\kappa^2\ell^2(\eta_\lambda^0)^2}{K^{2(b-c)}}$$

$$\phi_2 = \xi^2_{k+1}\left(\frac{4(1+\gamma_2)\kappa^2(\eta_\lambda^0)^2}{K^{2(b-c)}}\ell^2 + (1+1/\gamma_2)\left(1 - \frac{\mu_2\eta_0}{2JK^{(a-c)}}\right)\right) - \xi^2_k + \frac{4\ell_\Gamma\ell^2(\eta_\lambda^0)^2}{K^{2(b-c)}} + \frac{2\ell^2\eta_\lambda^0}{K^{(b-2c)}}$$

$$+ \frac{8\xi^1_{k+1}(1+\gamma_1)\kappa^2\ell^2(\eta_\lambda^0)^2}{K^{2(b-c)}}. \tag{37}$$

In order to achieve $\phi_1 \leq 0$ and $\phi_2 \leq 0$, we might let $\gamma_1 = \gamma_2 = 4JK^{(a-c)}/(\mu_2\eta_0) - 1$, then

$$(1+1/\gamma_1)\left(1 - \frac{\mu_2\eta_0}{JK^{(a-c)}}\right) \leq 1 - \frac{3\mu_2\eta_0}{4JK^{(a-c)}}$$

$$(1+1/\gamma_2)\left(1 - \frac{\mu_2\eta_0}{2JK^{(a-c)}}\right) \leq 1 - \frac{\mu_2\eta_0}{4JK^{(a-c)}}. \tag{38}$$

For $\eta_0 \leq \frac{8J}{\mu_2}$, we have $\frac{\mu_2\eta_0}{4J} \leq \frac{1}{2}$. Consider that ξ^1_k and ξ^2_k are non-increasing sequence, then $\xi^1_k \geq \xi^1_{k+1}$ and $\xi^2_k \geq \xi^2_{k+1}$, we have

$$\phi_1 \leq \xi^1_k\left(1 + \frac{(\eta_\lambda^0)^2\ell^2\kappa^2 J}{\mu_2\eta_0 K^{(2b-c-a)}} - \frac{3\mu_2\eta_0}{4JK^{(a-c)}}\right) - \xi^1_k + \frac{2\ell_\Gamma\ell^2(\eta_\lambda^0)^2}{K^{2(b-c)}} + \frac{\ell^2\eta_\lambda^0}{K^{(b-2c)}} + \xi^2_k\frac{8J(\eta_\lambda^0)^2\ell^2\kappa^2}{\mu_2\eta_0 K^{(2b-c-a)}} \leq 0$$

$$\phi_2 \leq \xi^2_k\left(1 + \frac{(\eta_\lambda^0)^2\ell^2\kappa^2 J}{\mu_2\eta_0 K^{(2b-c-a)}} - \frac{\mu_2\eta_0}{4JK^{(a-c)}}\right) - \xi^2_k + \frac{4\ell_\Gamma\ell^2(\eta_\lambda^0)^2}{K^{2(b-c)}} + \frac{2\ell^2\eta_\lambda^0}{K^{(b-2c)}} + \xi^1_k\frac{8J(\eta_\lambda^0)^2\ell^2\kappa^2}{\mu_2\eta_0 K^{(2b-c-a)}} \leq 0$$

If $\eta_\lambda^0 \leq 1/(2\ell_\Gamma)$ and $\eta_0/\eta_\lambda^0 \geq 6\sqrt{2}\kappa^2 J$, for $b \geq a$ and $k > 1$, then

$$\frac{2\ell_\Gamma\ell^2(\eta_\lambda^0)^2}{K^{2(b-c)}} \leq \frac{\ell^2\eta_\lambda^0}{K^{(b-2c)}}, \frac{9(\eta_\lambda^0)^2\ell^2\kappa^2 J}{\mu_2\eta_0} \leq \frac{\mu_2\eta_0}{8J}.$$

31980

The inequalities of ϕ_1, ϕ_2 can be simplified as

$$\phi_1 \le \xi_k^1 \left(1 - \frac{53\mu_2\eta_0}{72JK^{(a-c)}}\right) - \xi_k^1 + \frac{\ell^2\eta_\lambda^0}{K^{(b-2c)}} + \xi_k^2 \frac{\mu_2\eta_0}{9JK^{(a-c)}} \le 0 \tag{39}$$

$$\phi_2 \le \xi_k^2 \left(1 - \frac{17\mu_2\eta_0}{72JK^{(a-c)}}\right) - \xi_k^2 + \frac{2\ell^2\eta_\lambda^0}{K^{(b-2c)}} + \xi_k^1 \frac{\mu_2\eta_0}{9JK^{(a-c)}} \le 0 \tag{40}$$

We might solve the above inequalities and properly set

$$\xi_k^1 = \frac{-\frac{53\mu_2\eta_0}{72JK^{(a-c)}}\frac{2\ell^2\eta_\lambda^0}{K^{(b-2c)}} - \frac{\ell^2\eta_\lambda^0}{K^{(b-2c)}}\frac{\mu_2\eta_0}{9JK^{(a-c)}}}{\frac{\mu_2\eta_0}{9JK^{(a-c)}}\frac{\mu_2\eta_0}{9JK^{(a-c)}} - \frac{17\mu_2\eta_0}{72JK^{(a-c)}}\frac{53\mu_2\eta_0}{72JK^{(a-c)}}} = \frac{\frac{114\ell^2\eta_\lambda^0}{K^{(b-2c)}}}{\frac{837\mu_2\eta_0}{72JK^{(a-c)}}} = \frac{10\ell^2\eta_\lambda^0}{\mu_2\eta_0}\frac{J}{K^{(b-a-c)}}$$

$$\xi_k^2 = \frac{-\frac{17\mu_2\eta_0}{72JK^{(a-c)}}\frac{\ell^2\eta_\lambda^0}{K^{(b-2c)}} - \frac{2\ell^2\eta_\lambda^0}{K^{(b-2c)}}\frac{\mu_2\eta_0}{9JK^{(a-c)}}}{\frac{\mu_2\eta_0}{9JK^{(a-c)}}\frac{\mu_2\eta_0}{9JK^{(a-c)}} - \frac{17\mu_2\eta_0}{72JK^{(a-c)}}\frac{53\mu_2\eta_0}{72JK^{(a-c)}}} = \frac{3\ell^2\eta_\lambda^0}{\mu_2\eta_0}\frac{J}{K^{(b-a-c)}}$$

to guarantee that $\phi_1 \le 0$ and $\phi_2 \le 0$. Then the main inequality (35) can be estimated as

$$\mathbb{E}[R_{k+1} \mid \mathcal{F}_k, \mathcal{C}_k] \le R_k - \left(\frac{\eta_\lambda}{2} - \ell_\Gamma\eta_\lambda^2 - 2\xi_{k+1}^1(1+\gamma_1)\kappa^2\eta_\lambda^2 - 2\xi_{k+1}^2(1+\gamma_2)\kappa^2\eta_\lambda^2\right)\|\nabla\Gamma^\alpha(\lambda_k)\|^2$$

$$+ \frac{\ell_\Gamma\eta_\lambda^2}{2}\left(\sigma_1^2 + 2\alpha^2\sigma_2^2\right) + \xi_{k+1}^1(1+\gamma_1^{-1})\frac{\alpha^2\eta_u^2\sigma_2^2}{JB} + \xi_{k+1}^2(1+\gamma_2^{-1})\frac{\eta_w^2\left(\sigma_1^2+\alpha^2\sigma_2^2\right)}{JB}.$$

If we set $\eta_\lambda^0 \le 1/(8\ell_\Gamma)$, then $\ell_\Gamma\eta_\lambda^2 \le \frac{\eta_\lambda}{8}$. For $b \ge a$ and $k \ge 1$, if we set $\eta_0/\eta_\lambda^0 \ge 8\sqrt{3}\kappa^2 J$

$$\xi_k^1(1+\gamma_1)\kappa^2\eta_\lambda \le \frac{40(\eta_\lambda^0)^2K^{(a-c)}\ell^2\kappa^2J^2}{\mu_2^2\eta_0^2K^bK^{(b-a-c)}} = \frac{40(\eta_\lambda^0)^2\ell^2\kappa^2J^2}{\mu_2^2\eta_0^2K^{(2b-2a)}} \le \frac{1}{16}$$

$$\xi_k^2(1+\gamma_2)\kappa^2\eta_\lambda \le \frac{12(\eta_\lambda^0)^2K^{(a-c)}\ell^2\kappa^2J^2}{\mu_2^2\eta_0^2K^bK^{(b-a-c)}} = \frac{12(\eta_\lambda^0)^2\ell^2\kappa^2J^2}{\mu_2^2\eta_0^2K^{(2b-2a)}} \le \frac{1}{16}.$$

Then

$$\mathbb{E}[R_{k+1} \mid \mathcal{F}_k, \mathcal{C}_k] \le R_k - \frac{\eta_\lambda}{4}\|\nabla\Gamma^\alpha(\lambda_k)\|^2 + \frac{\ell_\Gamma\eta_\lambda^2}{2}\left(\sigma_1^2+2\alpha^2\sigma_2^2\right) + \frac{8\xi_{k+1}^1}{7}\frac{\alpha^2\eta_u^2\sigma_2^2}{JB} + \frac{3\xi_{k+1}^2}{4}\frac{\eta_w^2\left(\sigma_1^2+\alpha^2\sigma_2^2\right)}{JB}.$$

Telescoping the above inequality gives

$$\mathbb{E}\left[\left\|\nabla\Gamma^\alpha(\tilde\lambda)\right\|^2\right] = \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla\Gamma^\alpha(\lambda_k)\|^2\right]$$

$$\le \frac{4}{K\eta_\lambda}\left(\sum_{k=1}^{T}\mathbb{E}[R_k \mid \mathcal{F}_{k-1}, \mathcal{C}_{k-1}] - \mathbb{E}[R_{k+1} \mid \mathcal{F}_k], \mathcal{C}_k\right)$$

$$+ \frac{4}{K\eta_\lambda}\sum_{k=1}^{K}\left(\frac{\ell_\Gamma\eta_\lambda^2}{2}\left(\sigma_1^2+2\alpha^2\sigma_2^2\right) + \frac{8\xi_{k+1}^1}{7}\frac{\alpha^2\eta_u^2\sigma_2^2}{JB} + \frac{3\xi_{k+1}^2}{4}\frac{\eta_w^2\left(\sigma_1^2+\alpha^2\sigma_2^2\right)}{JB}\right)$$

$$\le \frac{4\mathbb{E}[R_1]K^b}{\eta_\lambda^0 K} + \frac{4K^b}{\eta_\lambda^0}\frac{\ell_\Gamma(\eta_\lambda^0)^2\left(\sigma_1^2+K^{2c}\sigma_2^2\right)}{2K^{2b}}$$

$$+ \frac{4K^b}{\eta_\lambda^0}\left(\frac{80\ell^2\eta_0\eta_\lambda^0K^{2c}K^{-2a}\sigma_2^2}{7\mu_2BK^{(b-a-c)}} + \frac{9\ell^2\eta_0\eta_\lambda^0K^{-2a}\left(\sigma_1^2+K^{2c}\sigma_2^2\right)}{4\mu_2BK^{(b-a-c)}}\right).$$

Recalling the result of Lemma 1 states the relation between the stationarity of the minimax problem and

the original bilevel problem, we have

$$\mathbb{E}\left[\left\|\nabla\mathcal{L}(\tilde{\lambda})\right\|^2\right] = \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla\mathcal{L}(\lambda_k)\|^2\right]$$

$$\leq \frac{2}{K}\sum_{k=1}^{K}\left(\mathbb{E}\left[\|\nabla\mathcal{L}(\lambda_k) - \nabla\Gamma^\alpha(\lambda_k)\|^2\right] + \mathbb{E}\left[\|\nabla\Gamma^\alpha(\lambda_k)\|^2\right]\right)$$

$$\leq \frac{2}{\alpha^2} + \frac{2}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla\Gamma^\alpha(\lambda_k)\|^2\right]$$

$$\leq \frac{2}{K^{2c}} + \frac{8\mathbb{E}[R_1]K^b}{\eta_\lambda^0 K} + \frac{8K^b}{\eta_\lambda^0}\frac{\ell_\Gamma(\eta_\lambda^0)^2\left(\sigma_1^2 + K^{2c}\sigma_2^2\right)}{2K^{2b}}$$

$$+ \frac{8K^b}{\eta_\lambda^0}\left(\frac{80\ell^2\eta_0\eta_\lambda^0 K^{2c}K^{-2a}\sigma_2^2}{7\mu_2 BK^{(b-a-c)}} + \frac{9\ell^2\eta_0\eta_\lambda^0 K^{-2a}\left(\sigma_1^2 + K^{2c}\sigma_2^2\right)}{4\mu_2 BK^{(b-a-c)}}\right).$$

Let $c = 1/7$, $a = 4/7$, and $b = 5/7$, we have

$$\mathbb{E}\left[\left\|\nabla\mathcal{L}(\tilde{\lambda})\right\|^2\right] = \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla\mathcal{L}(\lambda_k)\|^2\right]$$

$$\leq \mathcal{O}\left(\frac{1}{K^{2/7}}\right) + \mathcal{O}\left(\frac{\mathbb{E}[R_1]}{\eta_\lambda^0 K^{2/7}}\right) + \mathcal{O}\left(\frac{(1 + \ell\kappa\eta_0)\sigma_1^2}{BK^{4/7}}\right) + \mathcal{O}\left(\frac{(1 + \ell\kappa\eta_0)\sigma_2^2}{BK^{2/7}}\right).$$

Note that the initial state R_1 can be controlled by a constant which is independent with α:

$$R_1 = \Gamma^\alpha(\lambda_1) - \Gamma_{\min}^\alpha + \xi_1^1\delta_1 + \xi_2^1 r_1$$

$$= \Gamma^\alpha(\lambda_1) - \Gamma_{\min}^\alpha + \mathcal{O}\left(J\kappa\eta_0^\lambda/\eta_0\left(\|w_1 - w_*^\alpha(\lambda_1)\|^2 + \|u_1 - u_*(\lambda_1)\|^2\right)\right) \quad (41)$$

where

$$\Gamma^\alpha(\lambda_1) - \Gamma_{\min}^\alpha \leq \mathcal{L}^\alpha(\lambda_1, w_*^\alpha(\lambda_1), u_*(\lambda_1)) - \mathcal{L}^\alpha(\lambda_*, w_*^\alpha(\lambda_*), u_*(\lambda_*))$$

$$= L_1(\lambda_1, w_*^\alpha(\lambda_1)) - L_1(\lambda^*, w_*^\alpha(\lambda^*)) + \alpha\left(L_2(\lambda_1, w_*^\alpha(\lambda_1)) - L_2(\lambda_1, u_*(\lambda_1))\right)$$

$$+ \alpha\left(L_2(\lambda^*, w_*^\alpha(\lambda^*)) - L_2(\lambda^*, u_*(\lambda^*))\right)$$

$$= L_1(\lambda_1, w_*(\lambda_1)) - L_1(\lambda^*, w_*(\lambda^*)) + L_1(\lambda_1, w_*^\alpha(\lambda_1)) - L_1(\lambda_1, w_*(\lambda_1))$$

$$+ L_1(\lambda^*, w_*(\lambda^*)) - L_1(\lambda^*, w_*^\alpha(\lambda^*)) + \alpha\left(L_2(\lambda_1, w_*^\alpha(\lambda_1)) - L_2(\lambda_1, u_*(\lambda_1))\right)$$

$$+ \alpha\left(L_2(\lambda^*, w_*^\alpha(\lambda^*)) - L_2(\lambda^*, u_*(\lambda^*))\right)$$

$$\leq \mathcal{L}(\lambda_1) - \mathcal{L}(\lambda^*) + \ell_{10}\|w_*^\alpha(\lambda_1) - w_*(\lambda_1)\| + \ell_{10}\|w_*^\alpha(\lambda^*) - w_*(\lambda^*)\|$$

$$+ \alpha\frac{\ell_{21}}{2}\|w_*^\alpha(\lambda_1) - u_*(\lambda_1)\|^2 + \alpha\frac{\ell_{21}}{2}\|w_*^\alpha(\lambda^*) - u_*(\lambda^*)\|^2$$

$$\leq \mathcal{L}(\lambda_1) - \mathcal{L}(\lambda^*) + \frac{2\ell_{10}C_0}{\alpha} + 2\alpha\frac{\ell_{21}}{2}\frac{C_0^2}{\alpha^2}$$

$$\leq \mathcal{L}(\lambda_1) - \mathcal{L}(\lambda^*) + \frac{2\ell_{10}C_0\mu_2}{\ell_{11}} + \frac{\ell_{21}C_0^2\mu_2}{\ell_{11}} = \mathcal{L}(\lambda_1) - \mathcal{L}(\lambda^*) + \mathcal{O}\left(\kappa^2\ell_{21}\right), \quad (42)$$

where by definitions we know $w_*(\lambda) = u_*(\lambda)$ and the first inequality follows from the gradient-Lipschitz of L_2 and the Lipschitz continuity of L_1 in w, and the second inequality uses Lemma 4. The proof is complete. $\square$

31982