Enhancing Bilevel Optimization with Single-Level Learning Techniques

Group G4: David Yang, Liuyuan Jiang, Meltem Tatli, Quan Xiao

1 Introduction

Bilevel optimization has a long history in operations research, mathematics, engineering, and economics communities traced back to (Bracken & McGill, 1973), where the ultimate goal is to minimize a problem that depends on the optimal solutions of another problem. Earlier works on bilevel optimization have inspired prosperous literature on both theory, e.g., (Ye & Zhu, 1995; Vicente & Calamai, 1994; Colson et al., 2007; Sinha et al., 2017) and applications in the transportation network (Marcotte, 1986; Migdalas, 1995), portfolio management (Labbé et al., 1998) and game theory (Stackelberg, 1952); see a seminal textbook (Dempe, 2002).

Recently, bilevel optimization problems have regained significant attention due to their relevance in various real-world large-scale machine learning applications, including hyperparameter optimization (Maclaurin et al., 2015; Franceschi et al., 2017; 2018; Pedregosa, 2016), meta-learning (Finn et al., 2017), representation learning (Arora et al., 2020), reinforcement learning (Sutton & Barto, 2018; Stadie et al., 2020), continual learning (Pham et al., 2021; Borsos et al., 2020), adversarial learning (Zhang et al., 2022a; Robey et al., 2023) and neural architecture search (Liu et al., 2019); see recent survey (Liu et al., 2021; Zhang et al., 2023; Sinha et al., 2017).

In this project, we focus on a specific bilevel optimization problem represented as follows:

$$\min_x \; F(x) := f(x, y^*(x)), \quad \text{s.t.} \quad y^*(x) = \arg\min_y g(x,y) \tag{1}$$

where both upper-level objective f and lower-level objective g are continuously differentiable. We assume $g(x,\cdot)$ is μ_g strongly convex so that $y^*(x)$ is uniquely defined for all x. In practice, f and g can take the form of expectations when encountered with the stochasticity, i.e. $f(x,y) = \mathbb{E}_\zeta[f(x,y;\zeta)], g(x,y) = \mathbb{E}_\phi[g(x,y;\phi)]$.

From the nested optimization perspective, bilevel objective $F(x)$ is an implicit function of x. By the chain rule, $\nabla F(x)$ depends on the implicit gradient of lower-level solution $\nabla y^*(x)$, i.e.

$$\nabla F(x) = \nabla_x f(x, y^*(x)) + \nabla^\top y^*(x) \nabla_y f(x, y^*(x)). \tag{2}$$

Thanks to the implicit function theorem (Ghadimi & Wang, 2018), $\nabla y^*(x)$ has the form of

$$\nabla y^*(x) = -\left[\nabla^2_{yy} g(x, y^*(x))\right]^{-1} \nabla^2_{yx} g(x, y^*(x)). \tag{3}$$

Combining (2) and (3), the gradient of bilevel objective takes the form of

$$\nabla F(x) = \nabla_x f(x, y^*(x)) - \nabla^2_{xy} g(x, y^*(x)) \left[\nabla^2_{yy} g(x, y^*(x))\right]^{-1} \nabla_y f(x, y^*(x)). \tag{4}$$

As a result, advanced by the Hessian inversion estimation techniques, we can estimate the gradient of bilevel objective and employ gradient-type algorithm to solve (1).

However, the major drawback of the nested approaches are that they require the second-order information, which makes them computationally inefficient. In contrast, another line of researches focus on solving bilevel optimization from the constrained optimization perspective. Defining the minimal function value of lower-level objective as the value function $g^*(x) = \min_y g(x,y)$, we can reformulate the bilevel problem as

$$\min_{x,y} \; f(x,y), \quad \text{s.t.} \quad g(x,y) - g^*(x) \le 0. \tag{5}$$

The constraint in (5) restricts $y = y^*(x)$ so that the bilevel problem (1) now is equivalently reverted to (5). Then one can consider minimizing the Lagrangian function of (5) defined by

$$\mathcal{L}_\lambda(x,y) := f(x,y) + \lambda(g(x,y) - g^*(x)) \tag{6}$$

As the constraint in (5) does not preserve a strict feasible point, the optimal Lagrangian multiplier $\lambda = +\infty$. Letting $\mathcal{L}_\lambda^*(x) := \min_y \mathcal{L}_\lambda(x, y)$, Kwon et al. (2023a) showed that

$$\|\nabla F(x) - \nabla \mathcal{L}_\lambda^*(x)\| \leq \mathcal{O}(1/\lambda) \tag{7}$$

which also suggests the optimal Langrangian multiplier is unbounded. On the other hand, the most prevailing feature of $\mathcal{L}_\lambda(x, y)$ is that its gradient is fully first-order, which follows from

$$\nabla g^*(x) = \nabla g(x, y^*(x)) = \nabla_x g(x, y^*(x)) + \nabla^\top y^*(x) \nabla_y g(x, y^*(x)) = \nabla_x g(x, y^*(x)). \tag{8}$$

where the second equality holds according to the lower-level stationary condition $\nabla_y g(x, y^*(x)) = 0$. Denoting l_f as the smoothness constant of f, if $\lambda > 2l_f/\mu_g$, $\mathcal{L}_\lambda(x, y)$ is strongly convex in y. So similar to the derivation of (8), we have

$$\nabla \mathcal{L}_\lambda^*(x) = \nabla_x \mathcal{L}_\lambda(x, y_\lambda^*(x)), \quad \text{where} \quad y_\lambda^*(x) = \arg\min_y \mathcal{L}_\lambda(x, y) \tag{9}$$

This means although $\nabla F(x)$ contains second-order information, we can approximate it via fully first-order derivatives $\nabla \mathcal{L}_\lambda^*(x)$. Therefore, the fully first-order algorithm is designed in (Kwon et al., 2023a) based on the Lagrangian objective $\mathcal{L}_\lambda(x, y)$ with manually increasing λ.

Unlike the single-level optimization where only one stepsize is involved, fully first-order algorithm for bilevel problem requires tunning both upper-level and lower-level learning rates and the stepsize ratio. This poses a significant challenge due to the potential correlation between these learning rates. While adaptive gradient variants of nested bilevel algorithms have been explored in (Fan et al., 2023), there is a vacant for fully first-order bilevel method.

In this project, we boost the fully first-order bilevel method via Adam, aiming at improving the convergence performance. We empirical experiments on 2 commonly seen tasks, namely data hypercleaning and Regularization selection. The results demonstrate that F^{2}SA aided with Adam accelerates the convergence to a target accuracy and the performance of it is relatively stable to both the parameters and the settings.

2 RELATED WORK

Nested bilevel methods. Nested bilevel approaches can be classified into iteration differentiation (ITD) and approximation differentiation (AID) methods. ITD solves the lower level problem by an iterative solver and computes $\nabla F(x)$ by differentiating the lower-level objective through the iterates of the lower-level solver. It can be traced back to (Domke, 2012) and its nonasymptotic convergence rate was proved by (Grazzi et al., 2020). Empirical efforts are taken to reduce the memory cost and propose lightweight library for ease of users (Maclaurin et al., 2015; Grefenstette et al., 2019). However, it is time-consuming to obtain the hypergradient by differentiate through the lower-level optimizer such as lower level multiple steps of gradient descent, especially for large-scale machine learning problems. Different from ITD, AID leverages implicit function theorem and Hessian inversion estimation techniques to estimate $\nabla F(x)$. Existing literature has incorporated Neumann series approximation (Ghadimi & Wang, 2018; Chen et al., 2021b), conjugate gradient descent (Pedregosa, 2016; Ji et al., 2021), and kernel based method (Hataya & Yamada, 2023) to estimate the Hessian inversion. Recent advances include variance reduction and momentum based methods (Khanduri et al., 2021; Yang et al., 2021; Dagréou et al., 2022); warm-started algorithms (Arbel & Mairal, 2022; Li et al., 2022); distributed bilevel approaches (Tarzanagh et al., 2022; Lu et al., 2022; Yang et al., 2022b); adaptive bilevel method (Fan et al., 2023). Nevertheless, all of these methods require second order information.

Fully first order bilevel methods. While the second order bilevel methods have been extensively studied in the literature, efficient first order bilevel methods remained under-explored. Recently, (Liu et al., 2022) has first proposed a fully first order bilevel method by dynamic barrier gradient descent. Subsequently, (Kwon et al., 2023a) put forward a simple yet efficient fully first order bilevel algorithm for bilevel problem with strongly convex lower-level objective by connecting $\nabla F(x)$ with $\nabla \mathcal{L}_\lambda^*(x)$. A concurrent work (Shen & Chen, 2023) studied the relations of bilevel problem with its penalized problem, and proposed a fully first order method for bilevel problem with constrained nonconvex lower-level problem. Very recently, (Kwon et al., 2023b) has extended the algorithm in (Kwon et al., 2023a) to tackle the constrained nonconvex upper-level and lower-level problem. To the best of our knowledge, none of these works consider employing adaptive gradient schemes.

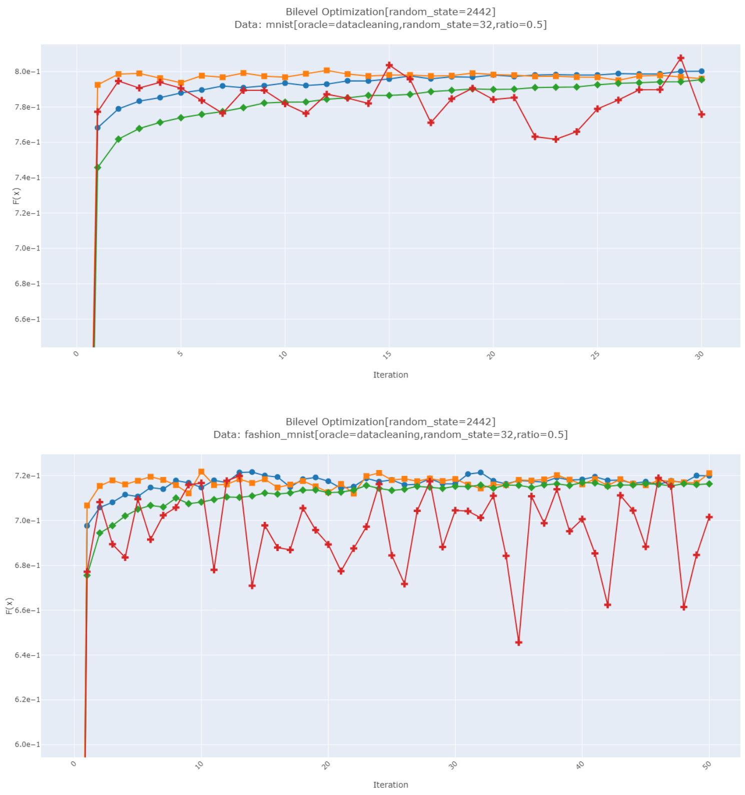


Figure 1: Test accuracy v.s. iteration comparison plot of F^{2}SA with Adam (Orange) and F^{2}SA (Blue), StocBiO (Red), SABA (Green) in hyper-cleaning tasks with corruption rate $p_c = 0.5$ on MNIST (up) and FashionMNIST dataset (bottom). For each algorithm, we plot the median performance over 10 runs. In both experiments, F^{2}SA with Adam converges faster.

Adaptive gradient methods. One limitation of (stochastic) gradient descent is that it scales the gradient uniformly in all directions by a pre-determined sequence of constants (a.k.a. learning rates). This may lead to poor performance when the training data are sparse (Duchi et al., 2011). To address this issue, adaptive learning rate methods have been developed by incorporating the knowledge of the geometry of the past observations to scale the gradient. Adaptive gradient methods have found great success in modern machine learning, and different variants such as RMSProp, AdaGrad, and Adam have been proposed (Tieleman et al., 2012; Kingma & Ba, 2014; Reddi et al., 2019; Ward et al., 2020; Zhang et al., 2022b). Later on, adaptive gradient methods have been integrated in zeroth-order optimization (Chen et al., 2019), distributed learning (Chen et al., 2021a), min-max optimization (Antonakopoulos et al., 2020; Yang et al., 2022a; Huang et al., 2023), and second order bilevel optimization (Fan et al., 2023). As far as we know, adaptive gradient methods have not yet been studied for fully first order bilevel optimization regime.

3 Background

In this section, we first review the update of Adam in single-level optimization and then introduce the fully first-order algorithm (F^{2}SA) for bilevel optimization in (Kwon et al., 2023a).

3.1 Adam

Consider the setting where we optimize $\min_\theta l(\theta)$ and initialize the first and second moment $m_{-1} = 0, v_{-1} = 0$. At each iteration t, Adam first obtains an unbiased gradient estimator g_t and then update

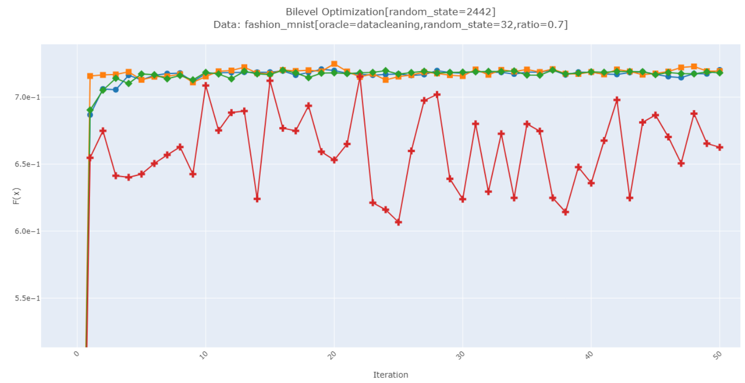


Figure 2: Test accuracy v.s. iteration comparison plot of F^{2}SA with Adam (Orange) and F^{2}SA (Blue), StocBiO (Red), SABA (Green) in hyper-cleaning tasks with corruption rate $p_c = 0.7$ on FashionMNIST dataset. For each algorithm, we plot the median performance over 10 runs. In both experiments, F^{2}SA with Adam converges faster.

the biased first and second moment estimate by

$$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1)g_t, \quad v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2)g_t^2. \tag{10}$$

Subsequently, the bias-corrected first and second moments are computed by

$$\hat{m}_t \leftarrow \frac{m_t}{1-\beta_1^t}, \quad \hat{v}_t \leftarrow \frac{v_t}{1-\beta_2^t}. \tag{11}$$

Finally, the parameter θ is updated by

$$\theta_{t+1} \leftarrow \theta_t - \alpha_k \frac{\hat{m}_{y,t}}{\sqrt{\hat{v}_{y,t}} + \epsilon} \tag{12}$$

where ϵ is a small constant. Algorithm 1 summarizes the process of calculating the first and second moments in Adam.

Algorithm 1 AdamGrad$(m_{t-1}, v_{t-1}, g_t, \beta_1, \beta_2, t)$

```
**Input:** Gradient estimator $g_t$, previous estimates $m_{t-1}$, $v_{t-1}$, parameters $\beta_1, \beta_2$
$m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1)g_t$
$v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2)g_t^2$
$\hat{m}_t \leftarrow \frac{m_t}{1-\beta_1^t}$
$\hat{v}_t \leftarrow \frac{v_t}{1-\beta_2^t}$
**Output:** Updated estimates $(\hat{m}_t, \hat{v}_t)$
```

3.2 F^{2}SA ALGORITHM

In order to estimate $\nabla \mathcal{L}_\lambda^*(x)$, F^{2}SA drives two sequences to chase $y_\lambda^*(x)$ and $y^*(x)$ according to (6), (8) and (9). That is, at each iteration k, we initialize $z_{k,0} = z_k, y_{k,0} = y_k$ and execute T step (stochastic) gradient descent with stepsize $\{\beta_k, \alpha_k\}$ in parallel,

$$z_{k,t+1} \leftarrow z_k - \beta h_{gz}^{k,t}, \quad y_{k,t+1} \leftarrow y_k - \alpha_k(h_{fy}^{k,t} + \lambda_k h_{gy}^{k,t})$$

where $h_{gz}^{k,t}, h_{gy}^{k,t}, h_{fy}^{k,t}$ are the unbiased estimator of $\nabla_y g(x_k, z_{k,t}), \nabla_y g(x_k, y_{k,t}), \nabla_y f(x_k, y_{k,t})$, and $z_{k,t}$ is used to chase $y^*(x_k)$, while $y_{k,t}$ aims to approach $y_\lambda^*(x_k)$. After that, we set $z_{k+1} = z_{k,T}, y_{k+1} = y_{k,T}$ and update x by one step (stochastic) gradient descent with a stepsize ratio ξ,

$$x_{k+1} \leftarrow x_k - \xi \alpha_k (h_{fx}^k + \lambda_k (h_{gxy}^k - h_{gxz}^k))$$

where $h^k_{fx}, h^k_{gxy}, h^k_{gxz}$ are the unbiased estimator of $\nabla_x f(x_k, y_{k+1}), \nabla_x g(x_k, y_{k+1}), \nabla_x f(x_k, z_{k+1})$. By defining the stochastic estimators as

$$
\begin{aligned}
&h_{gz}^{k,t} := \nabla_y g\left(x_k, z_{k,t}; \phi_z^{k,t}\right), h_{fy}^{k,t} := \nabla_y f\left(x_k, y_{k,t}; \zeta_y^{k,t}\right), h_{gy}^{k,t} := \nabla_y g\left(x_k, y_{k,t}; \phi_y^{k,t}\right), \\
&h_{gxy}^{k} := \nabla_x g\left(x_k, y_{k+1}; \phi_{xy}^{k}\right), h_{fx}^{k} := \nabla_x f\left(x_k, y_{k+1}; \zeta_x^{k}\right), h_{gxz}^{k} := \nabla_x g\left(x_k, z_{k+1}; \phi_{xz}^{k}\right)
\end{aligned}
$$

F^{2}SA algorithm is summarized in Algorithm 2.

Algorithm 2 F^{2}SA

```
**Input:** step sizes: $\{\alpha_k, \gamma_k\}$, multiplier difference sequence: $\{\delta_k\}$, inner-loop iteration count: $T$, step-size ratio: $\xi$, initializations: $\lambda_0, x_0, y_0, z_0$
**for** $k = 0$ to $K - 1$ **do**
    $z_{k,0} \leftarrow z_k, y_{k,0} \leftarrow y_k$
    **for** $t = 0$ to $T - 1$ **do**
        $z_{k,t+1} \leftarrow z_{k,t} - \gamma_k h_{gz}^{k,t}$
        $y_{k,t+1} \leftarrow y_{k,t} - \alpha_k (h_{fy}^{k,t} + \lambda_k h_{gy}^{k,t})$
    **end for**
    $z_{k+1} \leftarrow z_{k,T}, y_{k+1} \leftarrow y_{k,T}$
    $x_{k+1} \leftarrow x_k - \xi \alpha_k (h_{fx}^k + \lambda_k (h_{gxy}^k - h_{gxz}^k))$
    $\lambda_{k+1} \leftarrow \lambda_k + \delta_k$
**end for**
```

4 Algorithm: F^{2}SA with Adam

In this section, we propose the new algorithm F^{2}SA with Adam by incorporating Adam into the updates of x, y, z sequence in F^{2}SA, The full algorithm is summarized in Algorithm 3.

Algorithm 3 F^{2}SA with Adam.

```
**Input:** step sizes: $\{\alpha_k, \gamma_k\}$, multiplier difference sequence: $\{\delta_k\}$, inner-loop iteration $T$, initializations: $\lambda_0, x_0, y_0, z_0, m_{x,-1} \leftarrow 0, v_{x,-1} \leftarrow 0$, parameters: $\beta_1, \beta_2, \epsilon, \xi$
**for** $k = 0$ to $K - 1$ **do**
    $z_{k,0} \leftarrow z_k, y_{k,0} \leftarrow y_k$
    $m_{y,-1} \leftarrow 0, v_{y,-1} \leftarrow 0, m_{z,-1} \leftarrow 0, v_{z,-1} \leftarrow 0$
    **for** $t = 0$ to $T - 1$ **do**
        Update $(m_{z,t}, v_{z,t}) = $ AdamGrad$(m_{z,t-1}, v_{z,t-1}, h_{gz}^{k,t}, \beta_1, \beta_2, t)$
        $z_{k,t+1} \leftarrow z_{k,t} - \gamma_k \frac{m_{z,t}}{\sqrt{v_{z,t}} + \epsilon}$
        Update $(m_{y,t}, v_{y,t}) = $ AdamGrad$(m_{y,t-1}, v_{y,t-1}, h_{fy}^{k,t} + \lambda_k h_{gy}^{k,t}, \beta_1, \beta_2, t)$
        $y_{k,t+1} \leftarrow y_{k,t} - \alpha_k \frac{m_{y,t}}{\sqrt{v_{y,t}} + \epsilon}$
    **end for**
    $z_{k+1} \leftarrow z_{k,T}, y_{k+1} \leftarrow y_{k,T}$
    Update $(m_{x,k}, v_{x,k}) = $ AdamGrad$(m_{x,k-1}, v_{x,k-1}, h_{fx}^k + \lambda_k (h_{gxy}^k - h_{gxz}^k), \beta_1, \beta_2, k)$
    $x_{k+1} \leftarrow x_k - \xi \alpha_k \frac{m_{x,k}}{\sqrt{v_{x,k}} + \epsilon}$
    $\lambda_{k+1} \leftarrow \lambda_k + \delta_k$
**end for**
```

As a fully first-order method, F^{2}SA works well in practice when the second-order larger noises introduced may dampen the benefit of considering second-order information. However, it still fails to beat the second-order methods in the rate of convergence.

In this way, it is natural to think about coming up with an enhanced algorithm that brings more second-order information in without having too much additional noise. This is when *Adam* is considered, which approximate the second-order information using first-order term as well as fixing the learning-rate decay issue of AdaGrad, therefore being popular recently.

Methods	Time (sec)		Methods	Time (sec)	
	MNIST	FashionMNIST		MNIST	FashionMNIST
F^{2}SA with Adam	**1.2**	**3.1**	F^{2}SA	4.9	4.5
SABA	8.5	9.2	StocBiO	9.2	8.8

Table 1: Runtime comparison in hyper-cleaning tasks with corruption ratio $p_c = 0.5$ for different methods to achieve the maximum accuracy (i.e, 0.8 for MNIST and 0.72 for FashionMNIST).

Parameters	F^{2}SA	F^{2}SA with Adam
$\{\gamma_k, \alpha_k\}$	{1, 0.5, 0.1, 0.05, 0.01, 0.005}	{0.01, 0.005, 0.001, 0.0009, 0.0005}
ξ	{0.5, 1, 5}	{0.5, 1, 5}
δ_k	{0.01, 0.1, 0.5, 1}	{0.01, 0.1, 0.5, 1}
β_1	/	{0.9, 0.85, 0.8, 0.75, 0.7}
β_2	/	{0.999, 0.99, 0.98, 0.95,0.9, 0.85}
batch size	{16, 32, 64}	{16, 32, 64}

Table 2: Search grid for parameters in F^{2}SA and F^{2}SA with Adam.

5 Numerical Experiments

In this section, we compare F^{2}SA with Adam with other bilevel methods on two tasks. The baseline bilevel methods we choose are F^{2}SA (Kwon et al., 2023a), and two state-of-the-art second order bilevel methods, StocBiO (Ji et al., 2021) and SABA (Dagréou et al., 2022). Two bilevel tasks we tested are the data hyper-cleaning task (Franceschi et al., 2017) on the MNIST and the FashionMNIST dataset, and the regularization selection task (Franceschi et al., 2018) on the Ijcnn1 dataset (Prokhorov, 2001).

Our implementation of the F^{2}SA, SABA, and StocBiO algorithm is based on the code authored by (Dagréou et al., 2022), as made available in their code repository `https://github.com/benchopt/benchmark_bilevel`. We added a new FashionMNIST dataset and the proposed optimizer – F^{2}SA with Adam to it. We also corrected the definition of the accuracy in their repository, as the original one accounted for error percentage instead.

5.1 Data hyper-cleaning

Data hyper-cleaning aims to train a classifier within a corrupted environment but generalize well to clean, unseen data. To do so, we are given training data, where each label in the training dataset is substituted with a random class number according to a specified corruption rate p_c. We are also given the clean validation data to guide the training and the clean testing data. Let x be a vector being trained to label the noisy data and y be the model weight and bias, the objective is given by

$$\begin{aligned}
&\min_x \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(a_i,b_i)\in\mathcal{D}_{\text{val}}} \text{CE}\left(y^*(x); a_i, b_i\right) \\
&\text{s.t. } \; y^*(x) = \arg\min_y \frac{1}{|\mathcal{D}_{\text{tr}}|} \sum_{(a_i,b_i)\in\mathcal{D}_{\text{tr}}} [\sigma(x)]_i \text{CE}\left(y; a_i, b_i\right) + \frac{r}{2}\|y\|^2
\end{aligned}$$

where CE denotes the cross entropy loss and σ denotes sigmoid function. We add regularization $r > 0$ in LL objective to make it strongly convex.

Figure 1 shows the test accuracy over iterations of F^{2}SA with Adam with the state-of-the-art bilevel methods for hyper-cleaning tasks on MNIST and FashionMNIST datasets when $p_c = 0.5$. Besides, we list the exceutation time of different methods to achieve the maximum test accuracy in Table 1. It can be seen that although F^{2}SA with Adam introduces extra computation compared with F^{2}SA, this overhead is low compared with its speed gain. Also it is not suprising that F^{2}SA with Adam

Parameters	F^2SA		F^2SA with Adam	
	MNIST	FashionMNIST	MNIST	FashionMNIST
$\{\gamma_k, \alpha_k\}$	0.05	0.1	0.0009	0.001
ξ	1	1	5	1
δ_k	0.1	0.1	0.1	0.1
β_1	/	/	0.9	0.9
β_2	/	/	0.99	0.99
batch size	64	64	64	64

Table 3: Selected parameters in F^2SA and F^2SA with Adam on MNIST and FashionMNIST.

Parameters	F^2SA	F^2SA with Adam	Parameters	F^2SA	F^2SA with Adam
$\{\gamma_k, \alpha_k\}$	0.1	0.05	ξ	1	1
δ_k	0.1	0.1	β_1	/	0.9
β_2	/	0.85	batch size	64	32

Table 4: Selected parameters in F^2SA and F^2SA with Adam on Ijcnn1.

is faster than the second order bilevel methods SABA and StocBiO as the first-order gradients are computationally lighter. We also test the performance of F^2SA with Adam for higher corruption ratio $p_c = 0.7$ and show the results in Figure 2. It can be seen that F^2SA algorithm also converges the fastest in higher corruption setting.

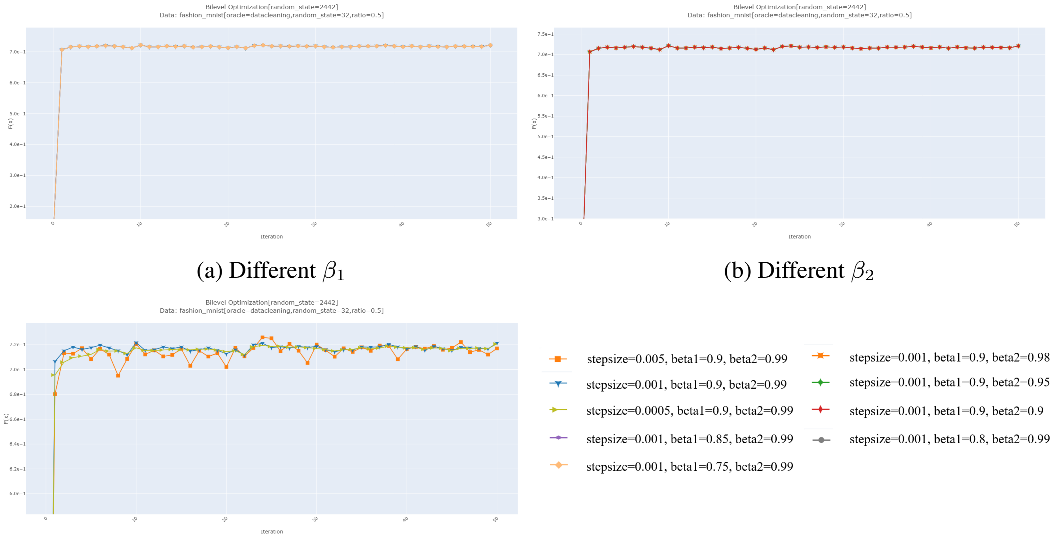
The optimal batch sizes, step sizes, and Adam algorithm's beta values were selected through grid search. The search grid is listed in Table 2 while the selected parameters can be found in Table 3. Both models were trained with the same batch size. However, for FashionMNIST dataset, the step size of F^2SA was 0.1, while the step size of F^2SA with Adam is 0.001, which is 100 times smaller. For MNIST dataset, F^2SA step size was 0.05 and F^2SA with Adam step size was 0.00009. In both experiments, we can see that F^2SA with Adam converges faster.

Figure 3 illustrates the performance comparisons of F^2SA with Adam under various parameters when $p_c = 0.5$. It is observed that the parameters β_1 and β_2 in Adam exhibit low sensitivity, provided that they remain close to 1. In contrast, the stepsizes α and γ are crucial for determining the convergence speed and stability when using F^2SA with Adam. Large step sizes can lead to instability, whereas too small step sizes result in slower convergence. Based on our experiments, we choose the optimal step size of 0.001. In bilevel learning, the optimal step size for F^2SA when used with Adam is found to be 100 times less than that for F^2SA alone. Interestingly, this pattern aligns with the default choice of the PyTorch library for single-level learning tasks, wherein the optimal step size for Adam is similarly 100 times smaller than that for SGD. This consistency underscores a parallel in stepsize scaling between the two learning paradigms.

5.2 Regularization selection

Regularization selection is a hyperparameter optimization task. It aims to find the optimal regularization coefficient x, which is used in training a model y on the training set, such that the learned model achieves the low risk on the validation set. Let CE $(y; a_i, b_i)$ denote the cross entropy loss of the model y on datum a_i and label b_i, and $\mathcal{D}_{\text{val}}$ and $\mathcal{D}_{\text{tr}}$ denote, respectively, the validation and training datasets. Specifically, we aim to solve

$$\begin{aligned} &\min_x \frac{1}{|\mathcal{D}_{\text{val}}|} \sum_{(a_i,b_i)\in\mathcal{D}_{\text{val}}} \mathrm{CE}\left(y^*(x); a_i, b_i\right) \\ &\text{s.t.} \quad y^*(x) = \arg\min_y \frac{1}{|\mathcal{D}_{\text{tr}}|} \sum_{(a_i,b_i)\in\mathcal{D}_{\text{tr}}} \mathrm{CE}\left(y; a_i, b_i\right) + \sum_{i=1}^{|\mathcal{D}_{\text{tr}}|} \exp\left(x_i\right) \|y_i\|^2. \end{aligned} \tag{13}$$



(a) Different β_1

(b) Different β_2

(c) Different stepsize $\alpha = \gamma$

(d) Legend for (a)-(c)

Figure 3: Comparisons of different parameters in F^{2}SA with Adam in hyper-cleaning tasks with $p_c = 0.5$ on FashionMNIST dataset.

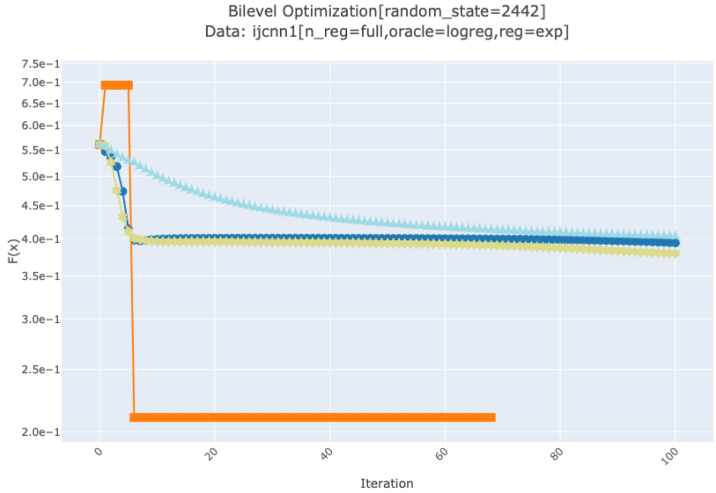


Figure 4: Loss v.s. iteration comparison plot of F^{2}SA with Adam (Orange) and F^{2}SA (Blue), StocBiO (Cyan), SABA (Green) in regularization selection task on Ijcnn1 dataset. For each algorithm, we plot the median performance over 10 runs.

The optimal hyperparameters for F^{2}SA and F^{2}SA with Adam can be found in Table 4 for the Ijcnn1 dataset. We report the loss v.s. iteration plot of all methods with the optimal hyperparameter in Figure 4. It can be seen that though a little unstable at the beginning, adding Adam to F^{2}SA benefits both the convergence speed and the final loss value. This alignes with the observations for the data hyper-cleaning task.

6 Conclusions

In this project, we focus on the bilevel optimization problem and boost a first order bilevel method F^{2}SA by the adaptive gradient method Adam. We conducted two experiments on bilevel learning, data hyper-clearning and regularization selection, to test the performance of the proposed method. Numerical results demonstrate that F^{2}SA aided with Adam accelerates the convergence to a target accuracy and the performance of it is relatively stable to both the parameters and the settings. We can especially observe this in the data hyper-cleaning task as the convergence of the proposed algorithm is faster than base algorithms. In the case of regularization selection, while initally it is less stable than other algorithms, in the long term it achieves the lowest objective value.

References

Kimon Antonakopoulos, E Veronica Belmega, and Panayotis Mertikopoulos. Adaptive extra-gradient methods for min-max optimization and games. *arXiv preprint arXiv:2010.12100*, 2020.

Michael Arbel and Julien Mairal. Amortized implicit differentiation for stochastic bilevel optimization. In *Proc. International Conference on Learning Representations*, virtual, 2022.

Sanjeev Arora, Simon Du, Sham Kakade, Yuping Luo, and Nikunj Saunshi. Provable representation learning for imitation learning via bi-level optimization. In *Proc. International Conference on Machine Learning*, virtual, 2020.

Zalán Borsos, Mojmír Mutnỳ, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. In *Proc. Advances in Neural Information Processing Systems*, virtual, 2020.

Jerome Bracken and James T McGill. Mathematical programs with optimization problems in the constraints. *Operations Research*, 21(1):37–44, 1973.

Tianyi Chen, Ziye Guo, Yuejiao Sun, and Wotao Yin. Cada: Communication-adaptive distributed adam. In *Proc. International Conference on Artificial Intelligence and Statistics*, virtual, 2021a.

Tianyi Chen, Yuejiao Sun, and Wotao Yin. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. *Proc. Advances in Neural Information Processing Systems*, 2021b.

Xiangyi Chen, Sijia Liu, Kaidi Xu, Xingguo Li, Xue Lin, Mingyi Hong, and David Cox. Zo-adamm: Zeroth-order adaptive momentum method for black-box optimization. In *Proc. Advances in Neural Information Processing Systems*, Vancouver, Canada, 2019.

Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, 153(1):235–256, 2007.

Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, 2022.

Stephan Dempe. *Foundations of bilevel programming*. Springer Science & Business Media, 2002.

Justin Domke. Generic methods for optimization-based modeling. In *Proc. International Conference on Artificial Intelligence and Statistics*, 2012.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Chen Fan, Gaspard Choné-Ducasse, Mark Schmidt, and Christos Thrampoulidis. Bisls/sps: Auto-tune step sizes for stable bi-level optimization. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, 2023.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. International Conference on Machine Learning*, Sydney, Australia, 2017.

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *Proc. International Conference on Machine Learning*, Sydney, Australia, 2017.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimilano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *Proc. International Conference on Machine Learning*, Stockholm, Sweden, 2018.

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

Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. On the iteration complexity of hypergradient computation. In *Proc. International Conference on Machine Learning*, virtual, 2020.

Edward Grefenstette, Brandon Amos, Denis Yarats, Phu Mon Htut, Artem Molchanov, Franziska Meier, Douwe Kiela, Kyunghyun Cho, and Soumith Chintala. Generalized inner loop meta-learning. *arXiv preprint arXiv:1910.01727*, 2019.

Ryuichiro Hataya and Makoto Yamada. Nyström method for accurate and scalable implicit differentiation. In *Proc. International Conference on Artificial Intelligence and Statistics*, Valencia, Spain, 2023.

Feihu Huang, Xidong Wu, and Zhengmian Hu. Adagda: Faster adaptive gradient descent ascent methods for minimax optimization. In *Proc. International Conference on Artificial Intelligence and Statistics*, Valencia, Spain, 2023.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *Proc. International Conference on Machine Learning*, virtual, 2021.

Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. In *Proc. Advances in Neural Information Processing Systems*, virtual, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jeongyeol Kwon, Dohyun Kwon, Stephen Wright, and Robert D Nowak. A fully first-order method for stochastic bilevel optimization. In *Proc. International Conference on Machine Learning*, Honolulu, HI, 2023a.

Jeongyeol Kwon, Dohyun Kwon, Steve Wright, and Robert Nowak. On penalty methods for nonconvex bilevel optimization and first-order stochastic approximation. *arXiv preprint arXiv:2309.01753*, 2023b.

Martine Labbé, Patrice Marcotte, and Gilles Savard. A bilevel model of taxation and its application to optimal highway pricing. *Management science*, 44(12-part-1):1608–1622, 1998.

Junyi Li, Bin Gu, and Heng Huang. A fully single loop algorithm for bilevel optimization without hessian inverse. In *Proc. Association for the Advancement of Artificial Intelligence*, virtual, 2022.

Bo Liu, Mao Ye, Stephen Wright, Peter Stone, et al. Bome! bilevel optimization made easy: A simple first-order approach. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, 2022.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: Differentiable architecture search. In *Proc. International Conference on Learning Representations*, New Orleans, LA, 2019.

Risheng Liu, Jiaxin Gao, Jin Zhang, Deyu Meng, and Zhouchen Lin. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Songtao Lu, Siliang Zeng, Xiaodong Cui, Mark Squillante, Lior Horesh, Brian Kingsbury, Jia Liu, and Mingyi Hong. A stochastic linearized augmented lagrangian method for decentralized bilevel optimization. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, 2022.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *Proc. International Conference on Machine Learning*, Lille, France, 2015.

Patrice Marcotte. Network design problem with congestion effects: A case of bilevel programming. *Mathematical programming*, 34(2):142–162, 1986.

Athanasios Migdalas. Bilevel programming in traffic planning: Models, methods and challenge. *Journal of global optimization*, 7(4):381–405, 1995.

Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *Proc. International Conference on Machine Learning*, New York City, NY, 2016.

Quang Pham, Chenghao Liu, Doyen Sahoo, and HOI Steven. Contextual transformation networks for online continual learning. In *Proc. International Conference on Learning Representations*, virtual, 2021.

Danil Prokhorov. Ijcnn 2001 neural network competition. *Slide presentation in IJCNN*, 1(97): 38, 2001. URL `https://www.openml.org/search?type=data&sort=runs&id=1575&status=active`.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

Alexander Robey, Fabian Latorre, George J Pappas, Hamed Hassani, and Volkan Cevher. Adversarial training should be cast as a non-zero-sum game. *arXiv preprint arXiv:2306.11035*, 2023.

Han Shen and Tianyi Chen. On penalty-based bilevel gradient descent method. In *Proc. International Conference on Machine Learning*, Honolulu, HI, 2023.

Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. A review on bilevel optimization: from classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22 (2):276–295, 2017.

Heinrich Von Stackelberg. *The Theory of Market Economy*. Oxford University Press, 1952.

Bradly Stadie, Lunjun Zhang, and Jimmy Ba. Learning intrinsic rewards as a bi-level optimization problem. In *Conference on Uncertainty in Artificial Intelligence*, virtual, 2020.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

Davoud Ataee Tarzanagh, Mingchen Li, Christos Thrampoulidis, and Samet Oymak. FEDNEST: Federated bilevel, minimax, and compositional optimization. In *Proc. International Conference on Machine Learning*, Baltimore, MD, 2022.

Tijmen Tieleman, Geoffrey Hinton, et al. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

Luis N Vicente and Paul H Calamai. Bilevel and multilevel programming: A bibliography review. *Journal of Global optimization*, 5(3):291–306, 1994.

Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. *The Journal of Machine Learning Research*, 21(1):9047–9076, 2020.

Junchi Yang, Xiang Li, and Niao He. Nest your adaptive algorithm for parameter-agnostic nonconvex minimax optimization. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, 2022a.

Junjie Yang, Kaiyi Ji, and Yingbin Liang. Provably faster algorithms for bilevel optimization. In *Proc. Advances in Neural Information Processing Systems*, virtual, 2021.

Shuoguang Yang, Xuezhou Zhang, and Mengdi Wang. Decentralized gossip-based stochastic bilevel optimization over communication networks. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, 2022b.

Jane J Ye and Daoli Zhu. Optimality conditions for bilevel programming problems. *Optimization*, 33(1):9–27, 1995.

Yihua Zhang, Guanhua Zhang, Prashant Khanduri, Mingyi Hong, Shiyu Chang, and Sijia Liu. Revisiting and advancing fast adversarial training through the lens of bi-level optimization. In *Proc. International Conference on Machine Learning*, Baltimore, MD, 2022a.

Yihua Zhang, Prashant Khanduri, Ioannis Tsaknakis, Yuguang Yao, Mingyi Hong, and Sijia Liu. An introduction to bi-level optimization: Foundations and applications in signal processing and machine learning. *arXiv preprint arXiv:2308.00788*, 2023.

Yushun Zhang, Congliang Chen, Naichen Shi, Ruoyu Sun, and Zhi-Quan Luo. Adam can converge without any modification on update rules. In *Proc. Advances in Neural Information Processing Systems*, New Orleans, LA, 2022b.