BILEVEL OPTIMIZATION: CONVERGENCE ANALYSIS AND ENHANCED DESIGN

Anagha Joshi
(UBID 50485475)

University at Buffalo
School of Engineering and Applied Sciences

AGENDA

Introduction

Algorithms

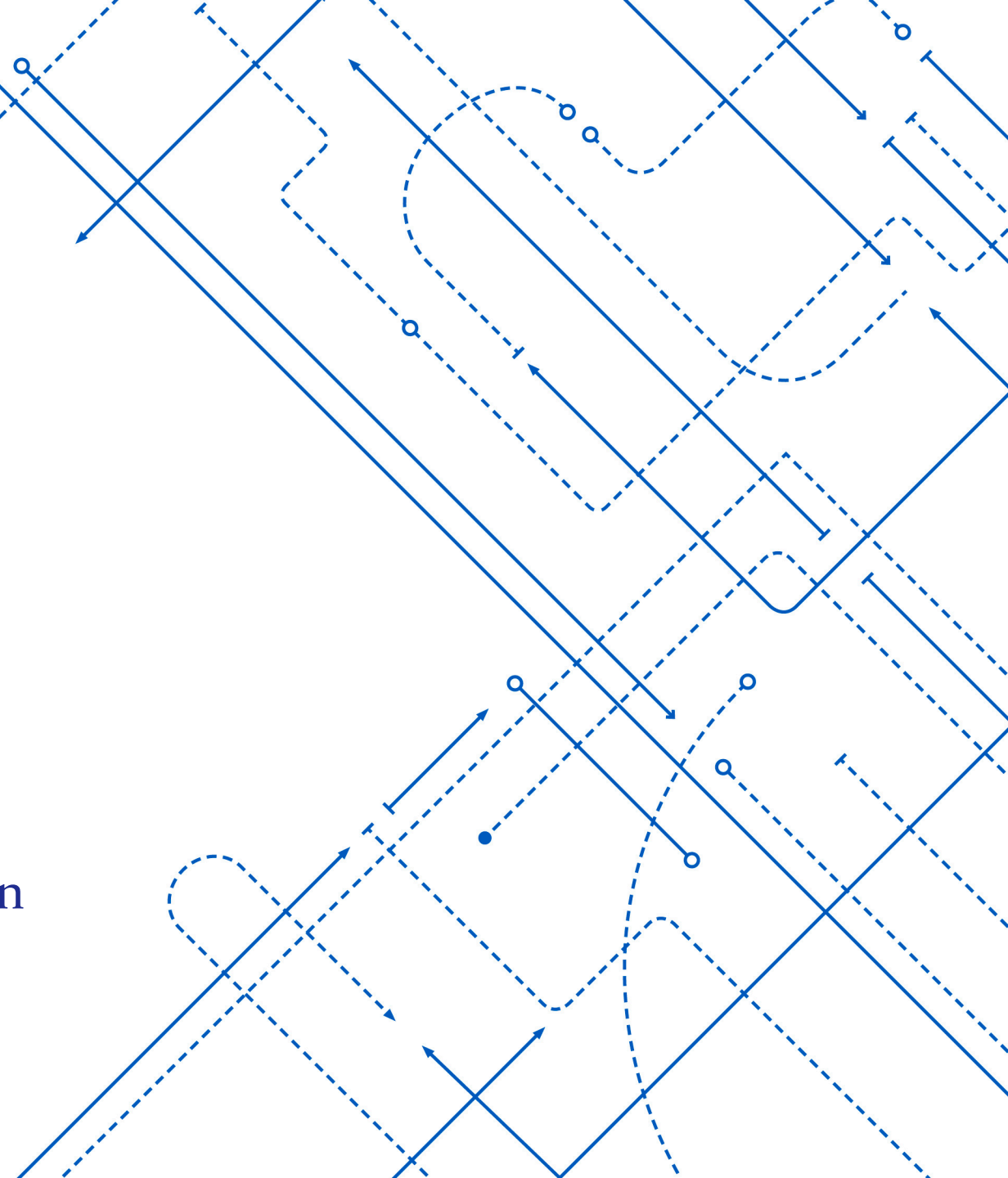
Definitions and Assumptions

Main Results for Bilevel Optimization

Applications to Meta-Learning

Applications to Hyperparameter Optimization

Conclusion



Introduction

- Bilevel optimization has received significant attention recently and become an influential framework in various machine learning applications including meta learning hyperparameter and reinforcement learning.
  A general bilevel optimization takes the following formulation

$$\min_{x \in \mathbb{R}^p} \Phi(x) := f(x, y^*(x))$$
$$\text{s.t.} \quad y^*(x) = \arg\min_{y \in \mathbb{R}^q} g(x, y),$$

- Setting:
  upper -level objective function in nonconvex
  lower-level function g is strongly convex w.r.t y

Introduction

- The first focus of this paper is to develop a comprehensive and sharper theory, which covers a broader class of bilevel optimizers via ITD and AID techniques, and more importantly, improves existing analysis with a more practical parameter selection and order wisely lower computational complexity
- The second focus of this paper is to design a more sample-efficient algorithm for bilevel stochastic optimization, which achieves lower computational complexity by orders of magnitude than BSA and TTSA

Algorithms

Algorithm 1 Bilevel algorithms via AID or ITD

Input: K, D, N, stepsizes α, β, initializations x_0, y_0, v_0.
for $k = 0, 1, 2, ..., K$ **do**
  Set $y_k^0 = y_{k-1}^D$ if $k > 0$ and y_0 otherwise
  for $t = 1, ..., D$ **do**
    Update $y_k^t = y_k^{t-1} - \alpha \nabla_y g(x_k, y_k^{t-1})$
  end for
  Hypergradient estimation via
  AID: 1) set $v_k^0 = v_{k-1}^N$ if $k > 0$ and v_0 otherwise
    2) solve v_k^N from $\nabla_y^2 g(x_k, y_k^D) v = \nabla_y f(x_k, y_k^D)$ via N steps of CG starting from v_k^0
    3) get Jacobian-vector product $\nabla_x \nabla_y g(x_k, y_k^D) v_k^N$ via automatic differentiation
    4) $\widehat{\nabla} \Phi(x_k) = \nabla_x f(x_k, y_k^D) - \nabla_x \nabla_y g(x_k, y_k^D) v_k^N$
  ITD: compute $\widehat{\nabla} \Phi(x_k) = \frac{\partial f(x_k, y_k^D)}{\partial x_k}$ via backpropagation
  Update $x_{k+1} = x_k - \beta \widehat{\nabla} \Phi(x_k)$
end for

Algorithms for Deterministic Bilevel Optimization

Algorithm 2 Stochastic bilevel optimizer (stocBiO)

Input: K, D, Q, stepsizes α and β, initializations x_0 and y_0.
for $k = 0, 1, 2, ..., K$ **do**
  Set $y_k^0 = y_{k-1}^D$ if $k > 0$ and y_0 otherwise
  for $t = 1, ..., D$ **do**
    Draw a sample batch $\mathcal{S}_{t-1}$
    Update $y_k^t = y_k^{t-1} - \alpha \nabla_y G(x_k, y_k^{t-1}; \mathcal{S}_{t-1})$
  end for
  Draw sample batches $\mathcal{D}_F, \mathcal{D}_H$ and $\mathcal{D}_G$
  Compute gradient $v_0 = \nabla_y F(x_k, y_k^D; \mathcal{D}_F)$
  Construct estimate v_Q via Algorithm 3 given v_0
  Compute $\nabla_x \nabla_y G(x_k, y_k^D; \mathcal{D}_G) v_Q$
  Compute gradient estimate $\widehat{\nabla}\Phi(x_k)$ via eq. (6)
  Update $x_{k+1} = x_k - \beta \widehat{\nabla}\Phi(x_k)$
end for

Algorithm 3 Construct v_Q given v_0

Input: Integer Q, samples $\mathcal{D}_H = \{\mathcal{B}_j\}_{j=1}^Q$ and constant η.
for $j = 1, 2, ..., Q$ **do**
  Sample $\mathcal{B}_j$ and compute $G_j(y) = y - \eta \nabla_y G(x, y; \mathcal{B}_j)$
end for
Set $r_Q = v_0$
for $i = Q, ..., 1$ **do**
  $r_{i-1} = \partial\big(G_i(y) r_i\big)/\partial y = r_i - \eta \nabla_y^2 G(x, y; \mathcal{B}_i) r_i$ via automatic differentiation
end for
Return $v_Q = \eta \sum_{i=0}^Q r_i$

Algorithm for Stochastic Bilevel Optimization

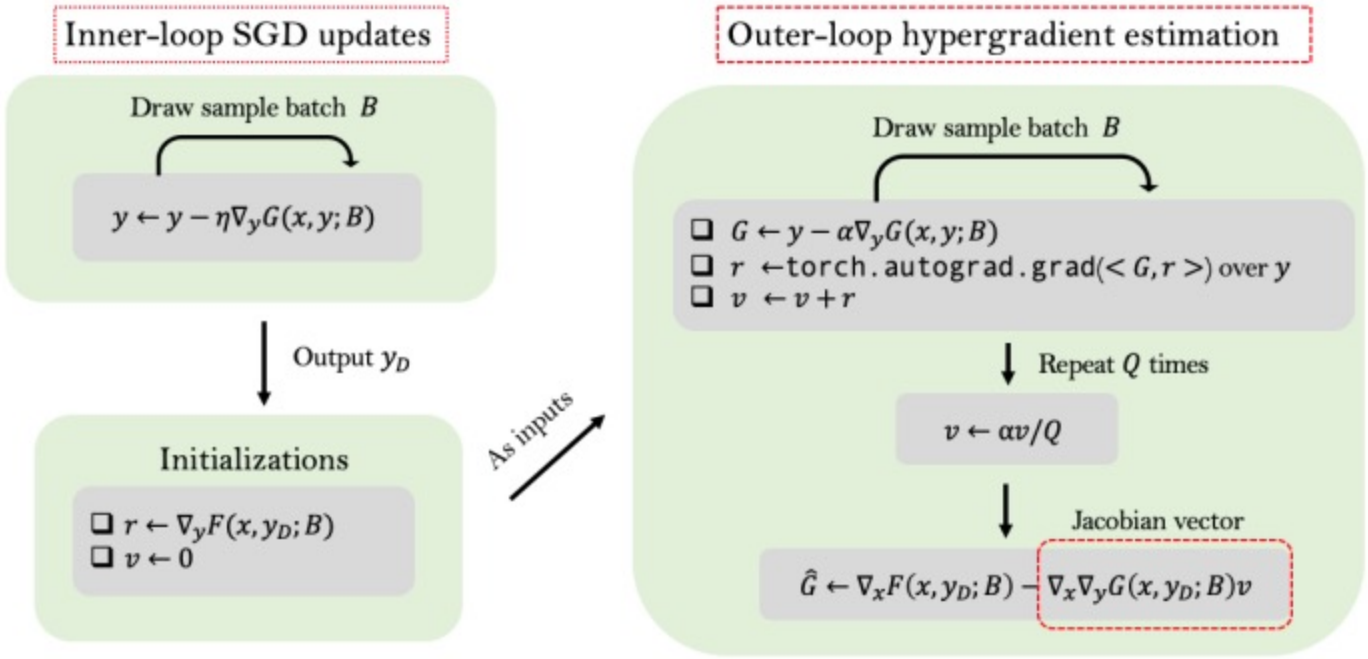


Illustration of hyperparameter estimation in stocBiO algorithm

Definitions and Assumptions

Assumption 1. *The lower-level function $g(x, y)$ is μ-strongly-convex w.r.t. y and the total objective function $\Phi(x) = f(x, y^*(x))$ is nonconvex w.r.t. x. For the stochastic setting, the same assumptions hold for $G(x, y; \zeta)$ and $\Phi(x)$, respectively.*

Since $\Phi(x)$ is nonconvex, algorithms are expected to find an ϵ-accurate stationary point defined as follows.

Definition 1. *We say $\bar{x}$ is an ϵ-accurate stationary point for the objective function $\Phi(x)$ in eq. (2) if $\mathbb{E}\|\nabla\Phi(\bar{x})\|^2 \leq \epsilon$, where $\bar{x}$ is the output of an algorithm.*

Definition 2. *For a function $f(x, y)$ and a vector v, let $\mathrm{Gc}(f, \epsilon)$ be the number of the partial gradient $\nabla_x f$ or $\nabla_y f$, and let $\mathrm{JV}(g, \epsilon)$ and $\mathrm{HV}(g, \epsilon)$ be the number of Jacobian-vector products $\nabla_x \nabla_y g(x, y)v$. and Hessian-vector products $\nabla_y^2 g(x, y)v$. For the stochastic case, similar metrics are adopted but w.r.t. the stochastic function $F(x, y; \xi)$.*

Assumption 2. *The loss function $f(z)$ and $g(z)$ satisfy*

- *The function $f(z)$ is M-Lipschitz, i.e., for any z, z',*

$$|f(z) - f(z')| \leq M\|z - z'\|.$$

- *$\nabla f(z)$ and $\nabla g(z)$ are L-Lipschitz, i.e., for any z, z',*

$$\|\nabla f(z) - \nabla f(z')\| \leq L\|z - z'\|,$$
$$\|\nabla g(z) - \nabla g(z')\| \leq L\|z - z'\|.$$

For the stochastic case, the same assumptions hold for $F(z; \xi)$ and $G(z; \zeta)$ for any given ξ and ζ.

Assumption 3. *Suppose the derivatives* $\nabla_x \nabla_y g(z)$ *and* $\nabla_y^2 g(z)$ *are* τ*- and* ρ*- Lipschitz, i.e.,*

- *For any* z, z', $\|\nabla_x \nabla_y g(z) - \nabla_x \nabla_y g(z')\| \leq \tau \|z - z'\|$.
- *For any* z, z', $\|\nabla_y^2 g(z) - \nabla_y^2 g(z')\| \leq \rho \|z - z'\|$.

For the stochastic case, the same assumptions hold for $\nabla_x \nabla_y G(z; \zeta)$ *and* $\nabla_y^2 G(z; \zeta)$ *for any* ζ.

Assumption 4. *Gradient* $\nabla G(z; \zeta)$ *has a bounded variance, i.e.,* $\mathbb{E}_\xi \|\nabla G(z; \zeta) - \nabla g(z)\|^2 \leq \sigma^2$ *for some* σ.

Main Results

Comparison of bilevel deterministic optimization algorithms.

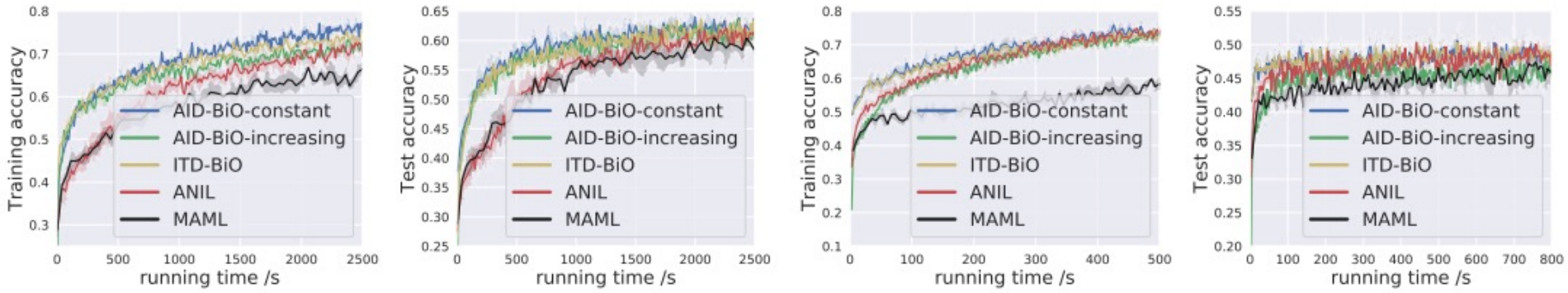
Algorithm	Gc(f,ϵ)	Gc(g,ϵ)	JV(g,ϵ)	HV(g,ϵ)
AID-BiO (Ghadimi & Wang, 2018)	$\mathcal{O}(\kappa^4\epsilon^{-1})$	$\mathcal{O}(\kappa^5\epsilon^{-5/4})$	$\mathcal{O}\left(\kappa^4\epsilon^{-1}\right)$	$\widetilde{\mathcal{O}}\left(\kappa^{4.5}\epsilon^{-1}\right)$
AID-BiO (this paper)	$\mathcal{O}(\kappa^3\epsilon^{-1})$	$\mathcal{O}(\kappa^4\epsilon^{-1})$	$\mathcal{O}\left(\kappa^3\epsilon^{-1}\right)$	$\mathcal{O}\left(\kappa^{3.5}\epsilon^{-1}\right)$
ITD-BiO (this paper)	$\mathcal{O}(\kappa^3\epsilon^{-1})$	$\widetilde{\mathcal{O}}(\kappa^4\epsilon^{-1})$	$\widetilde{\mathcal{O}}\left(\kappa^4\epsilon^{-1}\right)$	$\widetilde{\mathcal{O}}\left(\kappa^4\epsilon^{-1}\right)$

Gc(f,ϵ) and Gc(g,ϵ): number of gradient evaluations w.r.t. f and g. κ : condition number.
JV(g,ϵ): number of Jacobian-vector products $\nabla_x\nabla_y g(x,y)v$. Notation $\widetilde{\mathcal{O}}$: omit $\log\frac{1}{\epsilon}$ terms.
HV(g,ϵ): number of Hessian-vector products $\nabla_y^2 g(x,y)v$.

Comparison of bilevel stochastic optimization algorithms.

Algorithm	Gc(F,ϵ)	Gc(G,ϵ)	JV(G,ϵ)	HV(G,ϵ)
TTSA (Hong et al., 2020)	$\mathcal{O}(\text{poly}(\kappa)\epsilon^{-\frac{5}{2}})^*$	$\mathcal{O}(\text{poly}(\kappa)\epsilon^{-\frac{5}{2}})$	$\mathcal{O}(\text{poly}(\kappa)\epsilon^{-\frac{5}{2}})$	$\mathcal{O}(\text{poly}(\kappa)\epsilon^{-\frac{5}{2}})$
BSA (Ghadimi & Wang, 2018)	$\mathcal{O}(\kappa^6\epsilon^{-2})$	$\mathcal{O}(\kappa^9\epsilon^{-3})$	$\mathcal{O}\left(\kappa^6\epsilon^{-2}\right)$	$\widetilde{\mathcal{O}}\left(\kappa^6\epsilon^{-2}\right)$
stocBiO (this paper)	$\mathcal{O}(\kappa^5\epsilon^{-2})$	$\mathcal{O}(\kappa^9\epsilon^{-2})$	$\mathcal{O}\left(\kappa^5\epsilon^{-2}\right)$	$\widetilde{\mathcal{O}}\left(\kappa^6\epsilon^{-2}\right)$

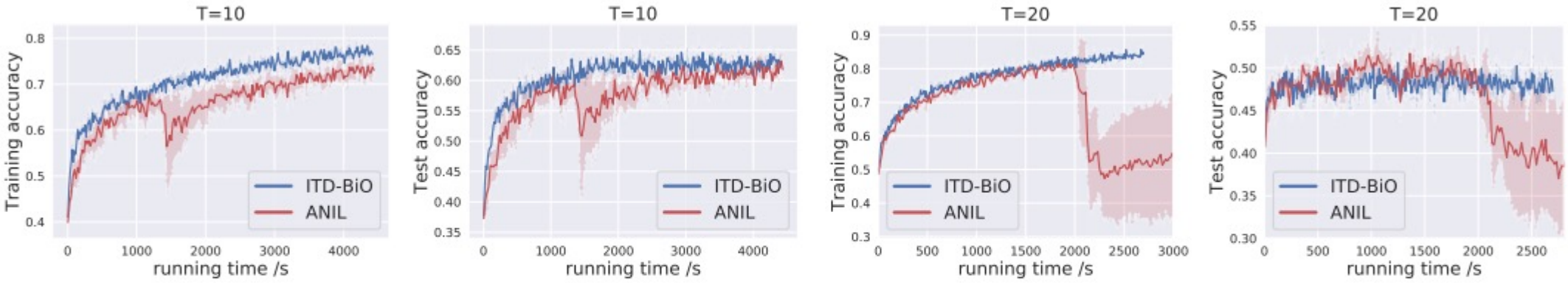
Applications to meta learning



(a) dataset: miniImageNet (b) dataset: FC100

Comparison of various bilevel algorithms on meta-learning. For each dataset, left plot: training accuracy v.s. running time; right plot: test accuracy v.s. running time.

Applications to meta learning

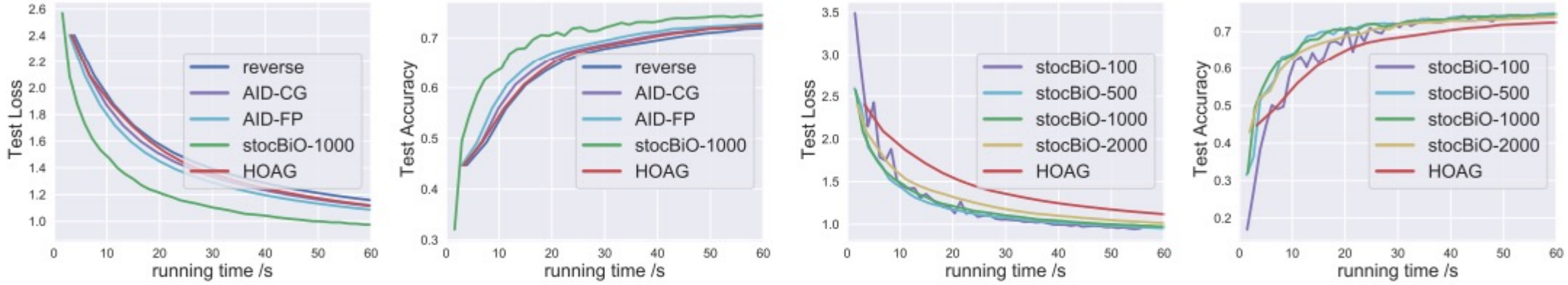


(a) $T = 10$, miniImageNet dataset

(b) $T = 20$, FC100 dataset

Comparison of ITD-BiO and ANIL with a relatively large inner-loop iteration number T.

Applications to Hyperparameter Optimization

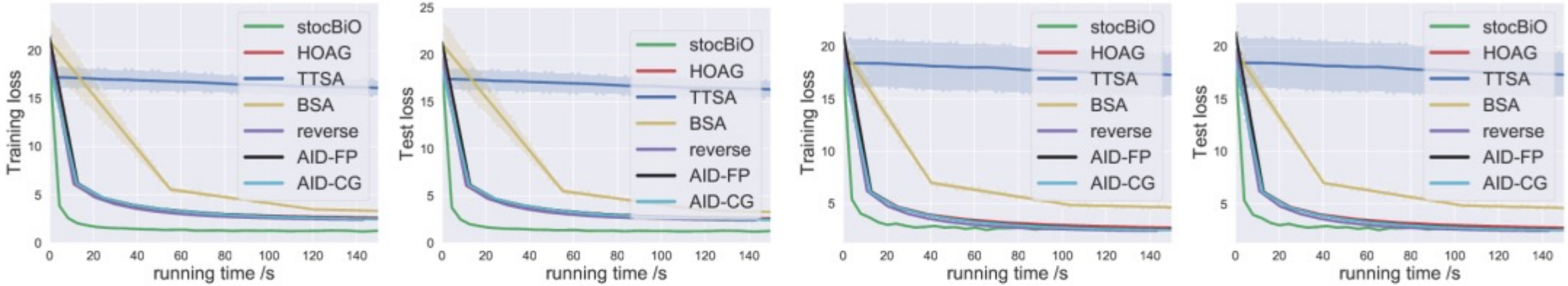


(a) Test loss and test accuracy v.s. running time

(b) Convergence rate with different batch sizes

Comparison of various stochastic bilevel algorithms on logistic regression on 20 Newsgroup dataset.

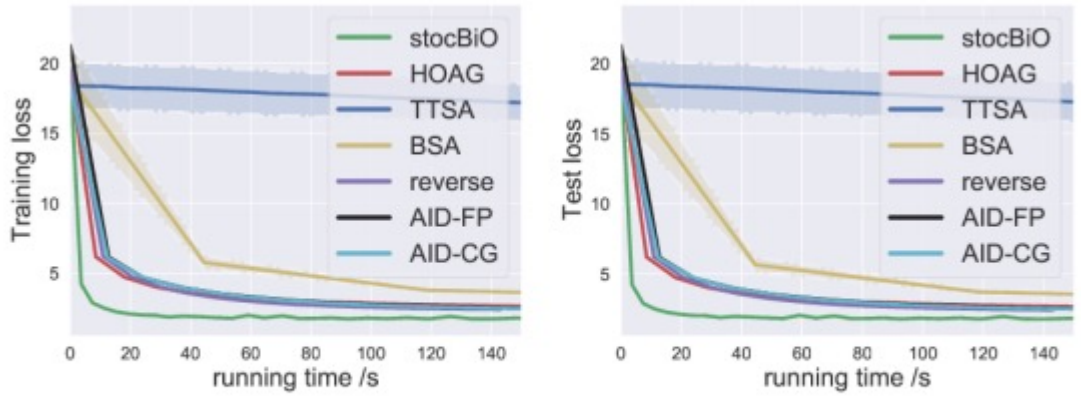
Applications to Hyperparameter Optimization



(a) Corruption rate $p = 0.1$ (b) Corruption rate $p = 0.4$

Comparison of various stochastic bilevel algorithms on hyperparameter optimization at different corruption rates. For each corruption rate p, left plot: training loss v.s. running time; right plot: test loss v.s. running time.

Applications to Hyperparameter Optimization



Convergence of algorithms at corruption rate $p = 0.2$.

Conclusion

- In this paper, a general and enhanced convergence rate analysis for the nonconvex-strongly-convex bilevel deterministic optimization is developed, a novel algorithm for the stochastic setting is proposed and it is showed that its computational complexity outperforms the best-known results order wisely.
- The results also provide the theoretical guarantee for various bilevel optimizers in meta-learning and hyperparameter optimization. The experiments validate theoretical results and demonstrate the superior performance of the proposed algorithm.