Hyper-gradient in bilevel optimization: efficient computation by Krylov subspace and enhanced investigation in reinforcement learning

Yan Yang [♥] [★] Bin Gao [♥] Ya-xiang Yuan [♥]

[♥] Academy of Mathematics and Systems Science, CAS

[★] University of Chinese Academy of Sciences

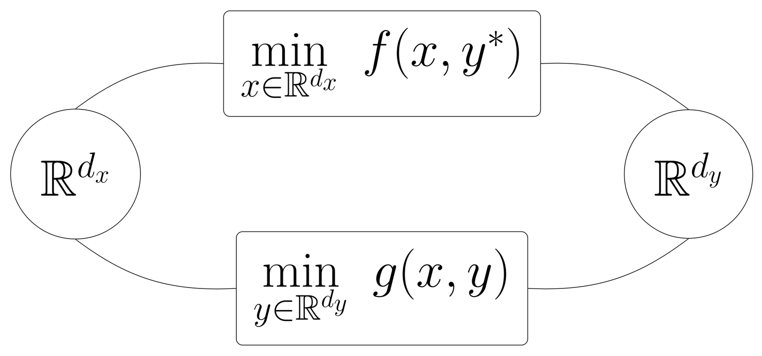
Bilevel optimization problem (BiO)

General form

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

▶ $f, g: \mathbb{R}^{d_x}\times\mathbb{R}^{d_y} \longrightarrow \mathbb{R}$, differentiable, g is **strongly convex**

The nested structure **couples** the **upper level** and **lower level**



Approximate implicit differentiation (AID)

Hyper-gradient computation

$$\nabla\varphi(x) = \nabla_x f(x, y^\star(x)) + \nabla_x y^\star(x)^\top \nabla_y f(x, y^\star(x))$$
$$= \nabla_x f(x, y^\star) - \nabla^2_{xy} g(x, y^\star)\left[\nabla^2_{yy} g(x, y^\star)\right]^{-1}\nabla_y f(x, y^\star)$$

Main difficulties

▶ Solving the lower-level problem to obtain $y^*(x)$

★ Approximating the **Hessian inverse vector product**

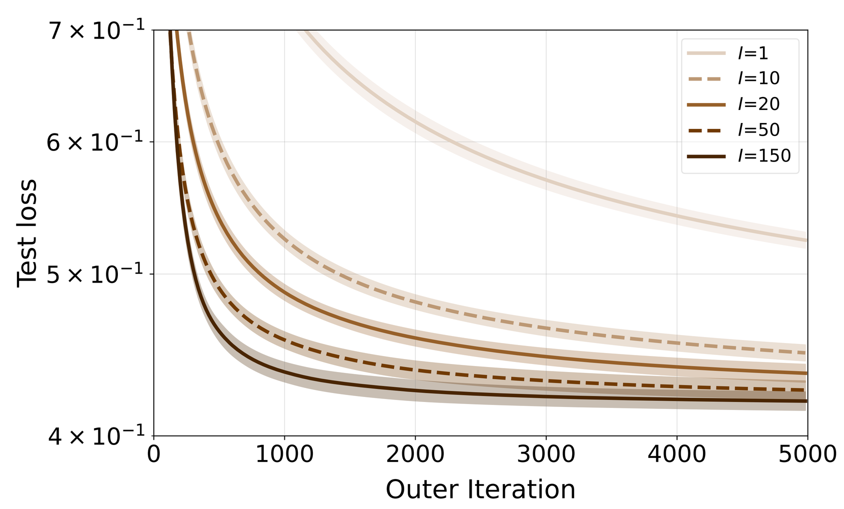
$$v^*(x) := \left[\nabla^2_{yy} g(x, y^*(x))\right]^{-1} \nabla_y f(x, y^*(x))$$

Vanilla update rule

$$1\times: x^+ = x - \beta\left(\nabla_x f(x,y) - \nabla^2_{xy} g(x,y)\left[\nabla^2_{yy} g(x,y)\right]^{-1}\nabla_y f(x,y)\right)$$
$$N\times: y^+ = y - \alpha\nabla_y g(x,y)$$

Motivation

An observation



The more **accurate** v^*, the more **enhanced** descent!

Tackle v^* in BiO

▶ **Approximation**

▶ **Amortization**

Adhere to these two principles

▶ Subspace methods for **Approximation** [Yuan'14]

▶ Subspace iteration for **Amortization** [Yuan'95]

Geometric interpretation of AID: a subspace perspective

$$\min_{v\in\mathcal{S}_k} m_k(v) := \frac{1}{2}v^\top\nabla^2_{yy} g(x_k, y_k)\, v - \nabla_y f(x_k, y_k)^\top v$$
$$\mathcal{S}_k = \mathbb{R}^{d_y} \Longrightarrow v_k = \nabla^2_{yy} g(x_k, y_k)^{-1}\nabla_y f(x_k, y_k) := A_k^{-1} b_k$$

Low-dimensional subspace?

Krylov subspace [Krylov'31] provides a good estimate for $A^{-1}b$ [Carmon and Duchi'18]

$$\mathcal{K}_N(A,b) := \operatorname{span}\left\{b, Ab, A^2 b, \dots A^{N-1} b\right\}$$

Dynamic Lanczos process in BiO

Core principles

▶ Maintain an **orthogonal basis** $Q_j = [q_1,\dots,q_j]$ to construct $\mathcal{S}_k \approx \mathcal{K}_j(A_j, b_j)$

▶ Keep the (approximate) projection matrix T_j **tridiagonal**

▶ Dynamically solve quadratic subproblems:

$$v_k := \arg\min_{v\in\mathcal{S}_k} m_k(v) := \frac{1}{2}v^\top A_k v - b_k^\top v$$

Adapt standard Lanczos process in BiO

$$u_j = A_j q_j - \beta_j q_{j-1}, \quad \alpha_j = q_j^\top u_j \quad \omega_j = u_j - \alpha_j q_j$$
$$\beta_{j+1} = \|\omega_j\| \quad q_{j+1} = \omega_j/\beta_{j+1}$$
$$T_j = \begin{pmatrix} T_{j-1} & \begin{matrix}0\\ \beta_j\end{matrix} \\ \begin{matrix}0 & \beta_j\end{matrix} & \alpha_j \end{pmatrix}.$$

Lanczos process is inherently unstable [Paige'80; Meurant and Strakoš'06]

Two "Res" strategies: LancBiO

Restart mechanism

▶ Restart subspaces each m steps

▶ Mitigate the accumulation of difference among $\{A_1,\dots,A_k\}$

Residual minimization

▶ Minimal residual subproblems

$$\min_{\Delta v\in\mathcal{S}_k} \|(b_k - A_k\bar{v}) - A_k\Delta v\|^2$$

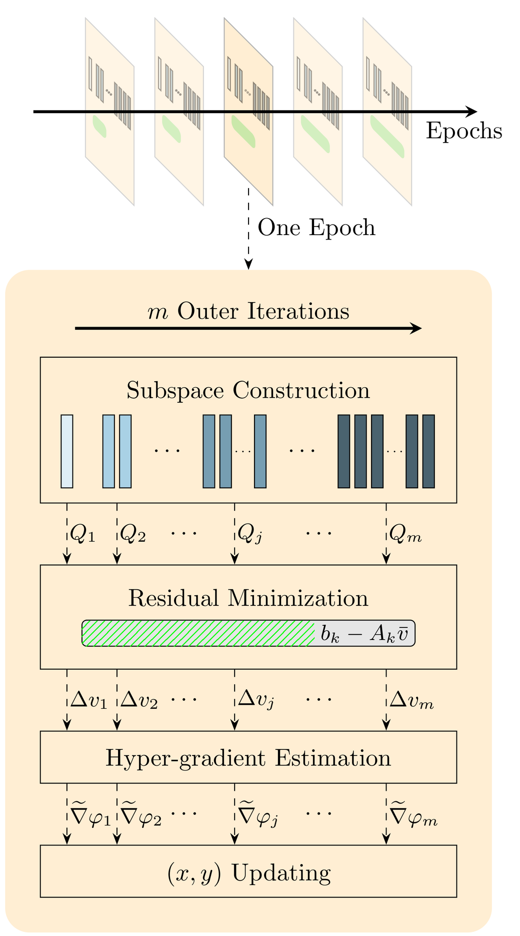
▶ Correct current $\bar{v}$, $v_k = \bar{v} + \Delta v_k$

▶ Collect historical information

Computational complexity

☺ Low-dimensional subproblems

☺ No cost of Hessian projection



Theoretical analysis and numerical experiment

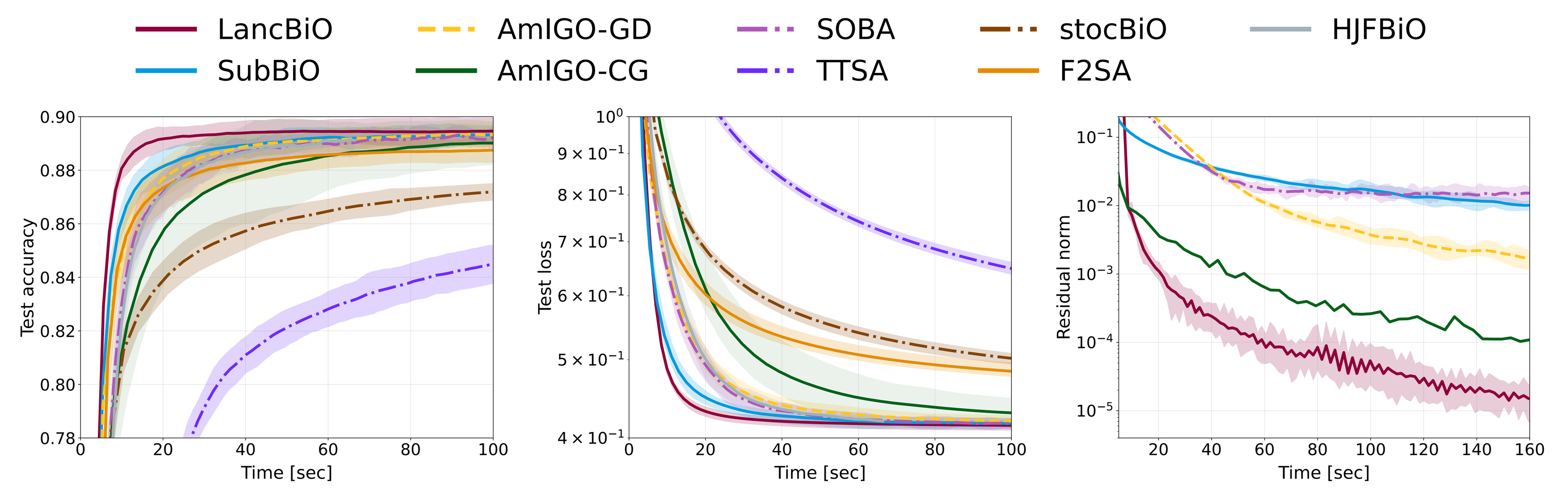
Theorem

With appropriate step sizes θ for y and λ for x, the iterates $\{x_k\}$ satisfy

$$\frac{m}{K(m-m_0)}\sum_{\substack{k=0,\\ (k \bmod m)>m_0}}^{K}\|\nabla\varphi(x_k)\|^2 = \mathcal{O}\left(\frac{m\lambda^{-1}}{K(m-m_0)}\right)$$

where m is the subspace dimension and $m_0 \sim \Omega(\log m)$

Test on MNIST (pollution rate=0.8)



Bilevel reinforcement learning (BiRL)

Parameterized MDP $\mathcal{M}_\tau(x) = (\mathcal{S}, \mathcal{A}, P, r(x), \gamma, \tau)$

▶ Soft value functions $V_s^\pi(x) = \mathbb{E}\left[\sum_{t=0}^\infty \gamma^t\left(r_{s_t a_t}(x) + \tau h(\pi_{s_t})\right) \,\middle|\, s_0 = s, \pi, \mathcal{M}_\tau(x)\right]$

▶ $h(\pi_s) = -\Sigma_a \pi_{sa}\log\pi_{sa}$ and $V^\pi_{\mathcal{M}_\tau(x)}(\rho) = \mathbb{E}_{s\sim\rho}\left[V_s^\pi(x)\right]$ with the initial distribution ρ

Problem formulation

$$\min_{x\in\mathbb{R}^n} \phi(x) := f(x, \pi^*(x)) \quad \text{s.t.} \quad \pi^*(x) = \arg\min_\pi -V^\pi_{\mathcal{M}_\tau(x)}(\rho)$$

where $f(x,\pi) = \mathbb{E}_{d_i\sim\rho(d;\pi)}\left[l(d_1, d_2, \dots, d_I; x)\right]$

Main difficulty

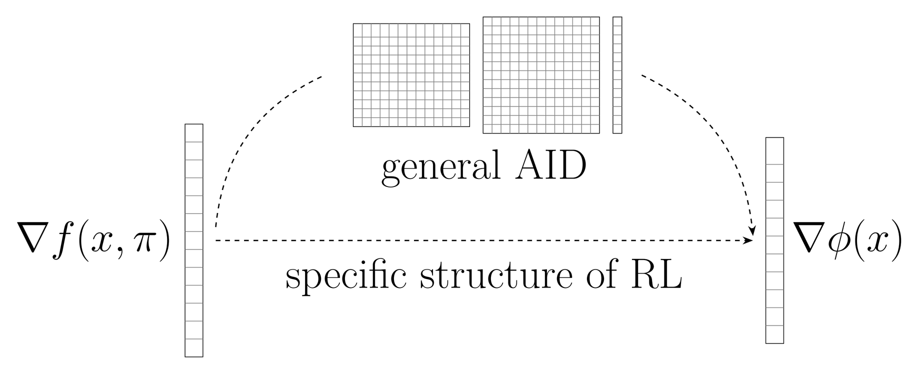
▶ The lower-level problem is inherently **non-convex** [Agarwal et al.'20; Lan'23 MP]

Does hyper-gradient $\nabla\phi(x)$ exist?

Develop hyper-gradient in BiRL

Estimate via sampling first-order information

$$\nabla\phi(x) = \mathbb{E}_{d_i}\left[\nabla l(\mathbf{d}; x)\right] + \tau^{-1}\mathbb{E}_{d_i}\left[l(\mathbf{d}; x)\left(\sum_i\sum_t \nabla\left(Q^*_{s_t^i a_t^i}(x) - V^*_{s_t^i}(x)\right)\right)\right]$$



Comparison of convergence results among biRL methods

Algorithm	Deter. or Stoch.	Conv. Rate	Inner Iter.	Oracle
PARL [Chakraborty et al.'24 ICLR]	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\log\epsilon^{-1})$	1st+2nd
PBRL [Shen et al.'24 ICML]	Deter.	$\mathcal{O}(\lambda\epsilon^{-1})$	$\mathcal{O}(\log\lambda^2\epsilon^{-1})$	1st
HPGD [Thoma et al.'24 NeurIPS]	Stoch.	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\log\epsilon^{-1})$	1st
M-SoBiRL	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(1)$	1st
SoBiRL	Deter.	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\log\epsilon^{-1})$	1st
Stoc-SoBiRL	Stoch.	$\tilde{\mathcal{O}}(\epsilon^{-1.5})$	$\mathcal{O}(\log\epsilon^{-1})$	1st

References

▶ Yan Yang, Bin Gao, Ya-xiang Yuan. *LancBiO: dynamic Lanczos-aided bilevel optimization via Krylov subspace.* ICLR (2025)

▶ Yan Yang, Bin Gao, Ya-xiang Yuan. *Bilevel reinforcement learning via the development of hyper-gradient without lower-level convexity.* AISTATS (2025)

▶ Code and more experiments are publicly available from https://github.com/UCAS-YanYang

yangyan@amss.ac.cn