Bilevel Optimization: Recent Algorithmic & Theoretical Advances, and Emerging Applications in Training LLMs

Mingyi Hong

Department of Electrical and Computer Engineering,

University of Minnesota, Minneapolis

UNIVERSITY OF MINNESOTA

Driven to Discover®

Collaborators (Alphabetical Order)

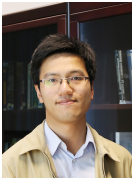
Volkan Cevher
(EPFL)

Alfredo Garcia
(TAMU)

Prashant Khanduri
(WSU)

Sijia Liu
(MSU)

Shoham Sabach
(Technion)



Hoi-To Wai
(CUHK)

Zhaoran Wang
(Northwestern)

Zhuoran Yang
(Yale)

Shuzhong Zhang
(UMN)

Collaborators (Alphabetical Order)

Xiaotian Jiang
(UMN)

Chenliang Li
(TAMU)

Jiaxiang Li
(UMN)

Hadi Reisizadeh
(UMN)

Bingqing Song
(UMN)

Ioannis Tsaknakis
(UMN)

Quan Wei
(UMN)

Siliang Zeng
(UMN)

Related Works (Optimization)

- M. Hong, H.-T. Wai, Z. Wang, Z. Yang, "A Two-Timescale Stochastic Algorithm Framework for Bilevel Optimization: Complexity Analysis and Application to Actor-Critic", **SIOPT**, 33 (1), 2023
- P. Khanduri, S. Zeng, M. Hong, H.-T. Wai, Z. Wang, Z. Yang, "A near-optimal algorithm for stochastic bilevel optimization via double-momentum", **NeurIPS** 2021
- P. Khanduri, I. Tsaknakis, Y. Zhang, J. Liu, S. Liu, J. Zhang, M. Hong, "Linearly Constrained Bilevel Optimization: A Smoothed Implicit Gradient Approach", **ICML** 2023
- X. Jiang, J. Li, M. Hong, S. Zhang, "A Barrier Function Approach for Bilevel Optimization with Coupled Lower-Level Constraints: Formulation, Approximation and Algorithms.", submitted, 2024
- I Tsaknakis, M Hong, S Zhang, "Minimax problems with coupled linear constraints: computational complexity, duality and solution methods", **SIOPT**, 2023

Related Works (Applications)

- S. Zeng, C. Li, A. Garcia and M. Hong, "Maximum-Likelihood Inverse Reinforcement Learning with Finite-Time Guarantees", **NeurIPS**, 2022
- S Zeng, M. Hong, A Garcia, "Structural estimation of markov decision processes in high-dimensional state space with finite-time guarantees, **Operations Research**, 2023
- S. Zeng, C. Li, A. Garcia and M. Hong, "When Demonstrations Meet Generative World Models: A Maximum Likelihood Framework for Offline Inverse Reinforcement Learning", **NeurIPS**, 2023, **(Oral)**
- C Li, S Zeng, Z Liao, J Li, D Kang, A Garcia, M Hong, "Joint demonstration and preference learning improves policy alignment with human feedback", **ICLR** 2025 **(Spotlight)**
- S. Zeng, Y. Liu, H. Rangwala, G. Karypis, M. Hong, R. Fakoor, "From demonstrations to rewards: Alignment without explicit human preferences", 2025
- H Reisizadeh, J Jia, Z Bu, B Vinzamuri, A Ramakrishna, K Chang, V Cevher, S Liu, M Hong, "BLUR: A Bi-Level Optimization Approach for LLM Unlearning", 2025

What is a Bilevel Problem,

& Why Should You Care?

Bilevel Optimization Problems (BLO)

- Let us consider the following **bilevel optimization** (BLO) problem:

$$\begin{aligned} \min_{x \in X \subseteq \mathbb{R}^{d_1}} \quad & \ell(x) := f(x, y^\star(x)) \\ \text{s.t.} \quad & y^\star(x) \in \arg\min_{y \in Y \subseteq \mathbb{R}^{d_2}} g(x, y), \\ & h(x, y^\star(x)) \leq 0 \end{aligned} \tag{B}$$

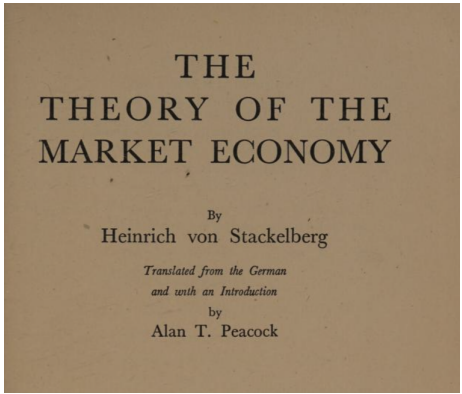
- $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$ are smooth (possibly non-convex) functions
- $h(x, y^\star(x)) \leq 0$ coupled constraints
- $f(\cdot)$ and $g(\cdot)$ are upper- and lower-level objective, respectively
- $\ell(\cdot)$ is the "outer" objective
- A challenging class of problems with a long history and rich structures.

Stackelberg Games

- In the 1934 book [Stackelberg, 1934] and the 1952 book [Stackelberg, 1952], Heinrich von Stackelberg introduced what is now called the Stackelberg game, a strategic game involving two players:

  Leader (L): Moves first, commits to a strategy

  Follower (F): Observes L's decision and chooses its optimal response
- L anticipates F's reaction, and optimizes its outcome
- This maps exactly to the structure of a BLO.

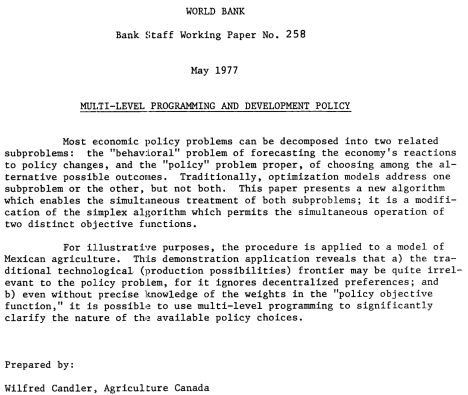

THE
THEORY OF THE
MARKET ECONOMY

By
Heinrich von Stackelberg

Translated from the German
and with an Introduction
by
Alan T. Peacock

Bi-Level Optimization Problems

- Later in 1970's BLO has been formulated in [Bracken-McGill, 1973], with an application to military resource allocation problem
- The name of bi-level/multi-level optimization problem has been formally coined in a World Bank report.



WORLD BANK

Bank Staff Working Paper No. 258

May 1977

MULTI-LEVEL PROGRAMMING AND DEVELOPMENT POLICY

Most economic policy problems can be decomposed into two related subproblems: the "behavioral" problem of forecasting the economy's reactions to policy changes, and the "policy" problem proper, of choosing among the alternative possible outcomes. Traditionally, optimization models address one subproblem or the other, but not both. This paper presents a new algorithm which enables the simultaneous treatment of both subproblems; it is a modification of the simplex algorithm which permits the simultaneous operation of two distinct objective functions.

For illustrative purposes, the procedure is applied to a model of Mexican agriculture. This demonstration application reveals that a) the traditional technological (production possibilities) frontier may be quite irrelevant to the policy problem, for it ignores decentralized preferences; and b) even without precise knowledge of the weights in the "policy objective function," it is possible to use multi-level programming to significantly clarify the nature of the available policy choices.

Prepared by:

Wilfred Candler, Agriculture Canada
and
Roger Norton, Development Research Center

Bilevel Optimization Problems

Starting 1980's BLO has received increased attention, due to applications in transportation, economics, power systems, signal processing, machine learning, etc.

A large body of works

- Hypergradient based methods [Kolstad and Lasdon, 1990, Savard and Gauvin, 1994, Falk and Liu, 1995]
- Penalty based methods [Aiyoshi and Shimizu, 1981, Ishizuka and Aiyoshi, 1992, Case, 1998]
- Lower-level KKT systems (MPEC) [Luo et al., 1996, Outrata et al., 2013]
- Methods for linear lower-level: BnB [Fortuny-Amat and McCarl, 1981]; Complimentary Pivoting [Ben-Ayed and Blair, 1990]; etc.

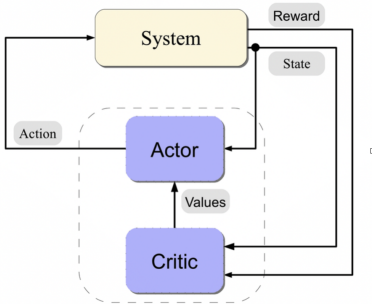
Where Do Bilevel Problems Show Up in the Wild Today?

BLO Applications in machine learning (ML) and AI

- As ML and AI problems getting more complex, many problems start to involve multiple subproblems
- "Empirical loss minimization" based modeling no longer sufficient
- BLO becomes an ideal candidate to formulate these type of problems

Reinforcement Learning and Policy Optimization

- BLO is closely related to the policy optimization problem in classical RL literature, particularly when combined with an actor-critic scheme [Konda and Tsitsiklis, 1999]
- The optimization involved is to find an optimal policy to maximize the expected (discounted) reward.
- Leader = 'actor' optimizes the policy, Follower = 'critic' evaluates the performance of the 'actor' (current policy).



AI Learning from Experience

- A recent article *"Era of Experience"* [Silver and Sutton, 2025] envisions a new AI learning paradigm – continued learning through interactive feedback rather than static datasets.
- This paradigm shift introduces nested learning dynamics:
  - Upper level: learning a reward or preference model from human feedback.
  - Lower level: adapting the model's policy via RL under that reward.

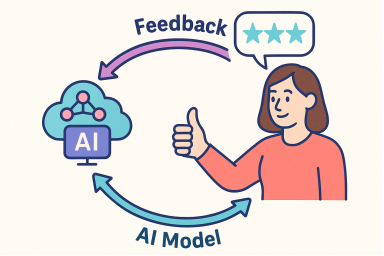
> Furthermore, users could provide feedback during the learning process, such as their satisfaction level, which could be used to fine-tune the reward function. The reward function can then adapt over time, to improve the way in which it selects or combines signals, and to identify and correct any misalignment. This can also be understood as a bi-level optimisation process that optimises user feedback as the top-level goal, and optimises grounded signals from the environment at the low level.[4] In this way, a small amount of human data may facilitate a large amount of autonomous learning.
>
> Silver-Sutton, "Welcome to the Era of Experience", 2025

AI Learning from Experience

- BLO provides the algorithmic and theoretical backbone for this emerging paradigm
- A rough BLO formulation can be written in the following form:

$$\min_{\theta_{\text{reward}}} \ \text{UserSatisfaction}(\pi^\star(\theta_{\text{reward}})) \quad \text{s.t.} \quad \pi^\star(\theta_{\text{reward}}) \in \arg\max_{\pi} R_{\theta_{\text{reward}}}(\pi)$$

- Feedback is provided interactively during training (e.g., preferences, corrections, satisfaction scores).
- R_θ is a reward function, which adapts over time based on streaming preferences
- $\pi^\star(\theta)$ is an AI model, re-optimized continuously to maximize reward.
- More to come soon.



Other applications of BLO

- Hyper-parameter Opt [Maclaurin et al., 2015, Franceschi et al., 2018]
- Neural Architecture Search [Liu et al., 2018, Xue et al., 2021]
- DNN Prunning [Sehwag et al., 2020, Zhang et al., 2022]
- Deep Reinforcement Learning [Gao et al., 2019, Vahdat et al., 2020]
- Wireless Telecommunications [Sun et al., 2021, Gao et al., 2020]
- Data Re-weighting [Pan et al., 2024, Fan et al., 2024]
- LLM unlearning [Reisizadeh et al., 2025]
- .....

A number of recent surveys on related topics [Sinha et al., 2017, Zhang et al., 2024].

Why are BLO Problems Hard?

Challenges of BLO: Theory

- The outer-objective $\ell(x)$ is generally nonconvex even when both levels are strongly convex and unconstrained
- Even if the problem appears to be well-defined (lower-problem is convex), $\ell(x)$ can be non-smooth, even discontinuous
- If the lower-level problem is non-convex, then even more challenging
- BLO problem is Σ_2^p-hard (harder than typical NP-hard problems in the polynomial hierarchy) [Bolte et al., 2025b].

Challenges of BLO: Theory

Upper level problem: $f(x, \mathbf{y}) = y_1$, $x \in \mathbb{R}^1$, $\mathbf{y} \in \mathbb{R}^2$;

$$\textbf{Lower level problem:} \quad \min_{\mathbf{y}} g(x, \mathbf{y}) := xy_1 + y_2$$
$$s.t. \quad y_2 \geq 0$$
$$1 \geq y_1 \geq -1$$

$$\mathbf{y}^*(x) = \begin{cases} (-1, 0) & \text{if } x > 0 \\ [-1, 1] \times \{0\} & \text{if } x = 0 \\ (1, 0) & \text{if } x < 0 \end{cases}$$

$$\textbf{Hyperfunction:} \quad \ell(x) = f(x, \mathbf{y}^*(x)) = \begin{cases} -1 & \text{if } x \geq 0 \\ 1 & \text{if } x < 0 \end{cases}$$

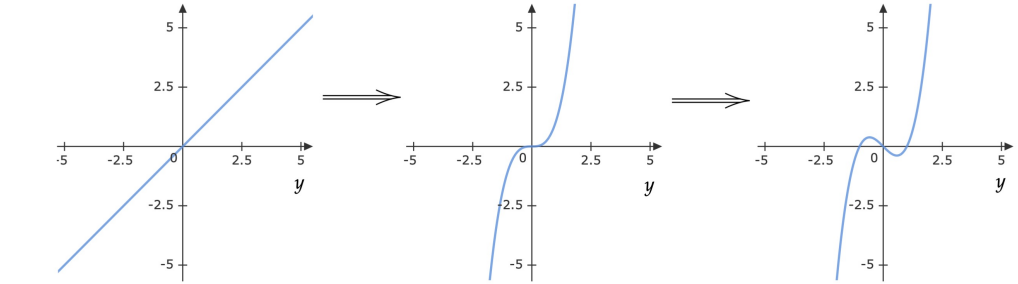
which is not continuous when $x = 0$.

Key Challenges when $g(\cdot)$ is non-convex

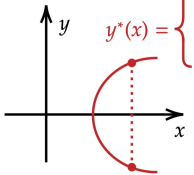
Ill-posedness of the Hyperfunction: $\ell(x)$ practically not computable

Bifurcation phenomenon: Solutions may emerge or disappear abruptly

- (Local) optimal set $S(x)$ may change discontinuously
- When bifurcation occurs, estimating $\|\hat{y} - y^*(x)\|$ becomes difficult due to Hessian degeneracy at $y^*(x)$



$$y^*(x) = \begin{cases} \varnothing & 0 \le x < 1/2 \\ \{0\} & x = 1/2 \\ \left\{ \pm \sqrt{\dfrac{2x-1}{3(3x-2x^2)}} \right\} & 1/2 < x \le 1 \end{cases}$$



- LL: $g(x,y) = (1-2x)y + (3x-2x^2)y^3$
- Bifurcation of stationary points with respect to y (denoted as $y^\star(x)$) occurs at $x = 1/2$

Challenges of BLO: Computation

Classical BLO algorithms are typically complicated:

- Requires Hessian computation [Kolstad and Lasdon, 1990]
- Can only handle deterministic problems [Falk and Liu, 1995]
- Many algorithms require multiple loops [Aiyoshi and Shimizu, 1981, Ghadimi and Wang, 2018, Couellan and Wang, 2016].

Desired properties:

- Lightweight computation
- Can deal with stochastic problems
- (sample/compute) efficient
- Theoretical guarantees.

Efficient Algorithms

& How to Design Them

Getting Started

- Let's get started with a family of tractable BLO problems
- Understand design issues for an efficient algorithm
- Then go over key developments that improve the state-of-the-art.

Algorithm Design for Problem (B)

- Consider the following simplifications:

$$\min_{x \in X \subseteq \mathbb{R}^{d_1}} \ell(x) \quad \Longleftrightarrow \quad \begin{aligned} \min_{x \in X \subseteq \mathbb{R}^{d_1}} \quad & \ell(x) := f(x, y^\star(x)) \\ \text{s.t.} \quad & y^\star(x) = \arg\min_{y \in \mathbb{R}^{d_2}} g(x, y), \end{aligned}$$

  $g(x, y)$ is strongly convex in y and unconstrained; no "coupling" constraints

- $\ell(x)$ is differentiable.
- We are interested in the stochastic setting where

$$f(x, y) := \mathbb{E}_\xi[f(x, y; \xi)], \quad g(x, y) := \mathbb{E}_\zeta[g(x, y; \zeta)].$$

The Hyper-Gradient

- Since $\ell(x)$ becomes differentiable, let's calculate its gradient

$$\nabla_x f(x, y^\star(x)) = \nabla_x f(x, y^\star(x)) + \underbrace{\nabla_x y^\star(x)^\top}_{\text{Jacobian Matrix}} \nabla_y f(x, y^\star(x))$$

Note for lower-level, $\nabla_y g(x, y^\star(x)) = 0$, by Implicit Function Theorem:

$$\nabla^2_{xy} g(x, y^\star(x)) + \nabla_x y^\star(x)^\top \nabla^2_{yy} g(x, y^\star(x)) = 0$$

We obtain:

$$\nabla_x f(x, y^\star) = \nabla_x f(x, y^\star) - \underbrace{\nabla^2_{xy} g(x, y^\star)[\nabla^2_{yy} g(x, y^\star)]^{-1} \nabla_y f(x, y^\star)}_{\text{implicit gradient}}.$$

Fixed point of a system of two coupled equations

- **Challenge**: the above calculation needs $y^\star(x)$, i.e., the lower-level problem has to be solved to global min $\Rightarrow$ generally not possible in practice

Idea: Consider the stationary condition of BLO as finding $(x^\star, y^\star)$ s.t.

$$F(x,y) = 0, \quad G(x,y) = \nabla_y g(x,y) = 0$$

$$\text{where} \quad F(x,y) = \nabla_x f(x,y) - \nabla^2_{xy} g(x,y)[\nabla^2_{yy} g(x,y)]^{-1} \nabla_y f(x,y)$$

We will call the function $F(x,y)$ the "surrogate" gradient.

Stochastic Estimates

- Let $F(\cdot;\xi), G(\cdot;\zeta)$ be the *stochastic estimates* of F, G
- Then the simplest and single-loop algorithm is given by:

$$x_{k+1} = x_k - \alpha_k F(x_k, y_k; \xi_{k+1})$$
$$y_{k+1} = y_k - \beta_k G(x_k, y_k; \zeta_{k+1})$$

  For this to work, we want unbiased estimates of F, G.

- **Challenge**: it is easy to estimate $G(\cdot) = \nabla_y g(\cdot)$, but how about $F(\cdot)$?

$$F(x,y) = \nabla_x f(x,y) - \nabla^2_{xy} g(x,y) \quad \underbrace{[\nabla^2_{yy} g(x,y)]^{-1}}_{\text{can't replace by } \nabla^2_{yy} g(x,y;\zeta)} \quad \nabla_y f(x,y)$$

- Biased estimate?

Stochastic Estimates

Subroutine: estimating $F(x,y)$ – `input: t`

Step 1. Set $p \in \{0, ..., \mathtt{t}-1\}$ uniformly at random.

Step 2. Construct the gradient estimate by (i.e., Neumann series)

$$h_f = \nabla_x f(x,y;\xi^{(1)}) - \nabla^2_{xy} g(x,y;\zeta_0^{(2)}) \left[\frac{\mathtt{t}}{L_g} \prod_{i=1}^{p} \left(I - \frac{c_h}{L_g} \nabla^2_{yy} g(x,y;\xi_i^{(2)}) \right) \right] \nabla_y f(x,y;\xi^{(1)}),$$

where $c_h = \mu_g/(\mu_g^2 + \sigma^2_{gxy})$

- We have (μ_g/L_g is the condition number) [H.-Wai-Wang-Yang-23] [1]

$$\|F(x,y) - \mathbb{E}[h_f]\| = \mathcal{O}((1-\mu_g/L_g)^{\mathtt{t}}), \ \mathbb{E}[\|h_f - \mathbb{E}[h_f]\|^2] = \mathcal{O}(\sigma^2).$$

[1] A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic M Hong, HT Wai, Z Wang, Z Yang, **SIOPT**, 2023

The Two Time-Scale Approximation (TTSA) Algorithm

- In [H.-Wai-Wang-Yang-23], a single-loop algorithm is proposed
- fix `t` to be of the order $\mathcal{O}(\log(1/\epsilon))$.

TTSA Algorithm for BLO – `input: t`

Follow the recursion:

$$x_{k+1} = x_k - \alpha_k h_f^k$$
$$y_{k+1} = y_k - \beta_k \nabla_y g(x_k, y_k; \zeta_{k+1})$$

h_f^k from the previous subroutine with $\mathtt{t} = \mathcal{O}(\log(1/\epsilon))$.

- Takes at most $\mathtt{t} + 2 = \mathcal{O}(\log(1/\epsilon))$ samples at each iteration
- Sample efficient, single-loop, SGD-like, easy to implement
- A **Two timescale stochastic approximation** (TTSA) algorithm, since $\alpha_k/\beta_k \to 0$.

General Assumptions (Informal)

Assumption 1 (upper-level function)

Consider the upper-level function $f(x, y)$ and $\ell(x) = f(x, y^\star(x))$:

1. $\nabla_y f(x, y)$ is Lipschitz in (x, y) + bounded; $\nabla_x f(x, y)$ is Lipschitz in y.

Assumption 2 (lower-level function)

Consider the lower-level function $g(x, y)$:

1. For any $x \in X$, $g(x, y)$ is **strongly convex** in y.
2. The Jacobian/Hessian $\nabla^2_{xy} g(x, y), \nabla^2_{yy} g(x, y)$ are Lipschitz in (x, y). Moreover, $\nabla^2_{xy} g(x, y)$ is bounded.

How to measure convergence?

Tracking Error

For lower-level problem, we care about if y^k **tracks** $y^\star(x^{k-1})$ well:

$$\Delta_y^k = \mathbb{E}[\|y^k - y^\star(x^{k-1})\|^2]$$

Optimality Gap ($\ell(\cdot)$ strongly convex)

For upper-level problem, we care about: $\Delta_x^k = \mathbb{E}[\|x^k - x^\star\|^2]$

Stationarity ($\ell(\cdot)$ convex/non-convex)

When the upper-level problem is possibly non-convex, we care about:

$$\Delta_x^k = \mathbb{E}[\|\nabla \ell(x^k)\|^2]$$

If $X \neq \mathbb{R}^{d_1}$, our result extends to the case with **nearly stationary** solution defined via the Moreau envelope [Davis and Drusvyatskiy, 2018].

Summary of Main Results: Complexity

Main Results

We characterize the **rate of convergence** for TTSA when:

(1) the inner objective $g(x, y)$ is strongly convex in y, and

(2) the outer objective $\ell(x)$ is *strongly convex, convex, weakly convex in* x.

(3) K is the total iteration we run

$\ell(x)$	Constraint	Step Size (α_k, β_k)	Rate (outer)	Rate (Inner)
SC	$X \subseteq \mathbb{R}^{d_1}$	$\mathcal{O}(k^{-1})$, $\mathcal{O}(k^{-2/3})$	$\mathcal{O}(K^{-2/3})$	$\mathcal{O}(K^{-2/3})$
C	$X \subseteq \mathbb{R}^{d_1}$	$\mathcal{O}(K^{-3/4})$, $\mathcal{O}(K^{-1/2})$	$\mathcal{O}(K^{-1/4})$	$\mathcal{O}(K^{-1/2})$
WC	$X \subseteq \mathbb{R}^{d_1}$	$\mathcal{O}(K^{-3/5})$, $\mathcal{O}(K^{-2/5})$	$\mathcal{O}(K^{-2/5})$	$\mathcal{O}(K^{-2/5})$

Key Lemma

Lemma

Let $K \geq 1$ be an integer. Consider sequences of non-negative scalars $\{\Omega^k\}_{k=0}^K$, $\{\Upsilon^k\}_{k=0}^K$, $\{\Theta^k\}_{k=0}^K$. Let $\mathsf{c}_0, \mathsf{c}_1, \mathsf{c}_2$, $\mathsf{d}_0, \mathsf{d}_1, \mathsf{d}_2$ be some positive constants. If the recursion holds

$$\Omega^{k+1} \leq \Omega^k - \mathsf{c}_0\Theta^{k+1} + \mathsf{c}_1\Upsilon^{k+1} + \mathsf{c}_2, \quad \Upsilon^{k+1} \leq (1-\mathsf{d}_0)\Upsilon^k + \mathsf{d}_1\Theta^k + \mathsf{d}_2,$$

for any $k \geq 0$. Then provided that $\mathsf{c}_0 - \mathsf{c}_1\mathsf{d}_1(\mathsf{d}_0)^{-1} > 0, \mathsf{d}_0 - \mathsf{d}_1\mathsf{c}_1(\mathsf{c}_0)^{-1} > 0$, it holds

$$\frac{1}{K}\sum_{k=1}^K \Theta^k \leq \frac{\Omega^0 + \frac{\mathsf{c}_1}{\mathsf{d}_0}\left(\Upsilon^0 + \mathsf{d}_1\Theta^0 + \mathsf{d}_2\right)}{\left(\mathsf{c}_0 - \mathsf{c}_1\mathsf{d}_1(\mathsf{d}_0)^{-1}\right)K} + \frac{\mathsf{c}_2 + \mathsf{c}_1\mathsf{d}_2(\mathsf{d}_0)^{-1}}{\mathsf{c}_0 - \mathsf{c}_1\mathsf{d}_1(\mathsf{d}_0)^{-1}}$$

$$\frac{1}{K}\sum_{k=1}^K \Upsilon^k \leq \frac{\Upsilon^0 + \mathsf{d}_1\Theta^0 + \mathsf{d}_2 + \frac{\mathsf{d}_1}{\mathsf{c}_0}\Omega^0}{\left(\mathsf{d}_0 - \mathsf{d}_1\mathsf{c}_1(\mathsf{c}_0)^{-1}\right)K} + \frac{\mathsf{d}_2 + \mathsf{d}_1\mathsf{c}_2(\mathsf{c}_0)^{-1}}{\mathsf{d}_0 - \mathsf{d}_1\mathsf{c}_1(\mathsf{c}_0)^{-1}}.$$

Summary of Main Results: Sample Complexity

- Consider the sample complexity for TTSA, compare with BSA proposed Ghadimi and Wang [2018]
- TTSA draws $\mathcal{O}(1)$ samples from the lower-level problem; BSA requires solving the lower-level problem to high accuracy

Table: Comparison of total samples needed between BSA [Ghadimi and Wang, 2018] and TTSA

Method	SC	C	WC
BSA	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-4})$	$\mathcal{O}(\epsilon^{-3})$
TTSA	$\mathcal{O}(\epsilon^{-3/2})$	$\mathcal{O}(\epsilon^{-4})$	$\mathcal{O}(\epsilon^{-5/2})$

Discussion: TTSA for Fixed Point Systems

- TTSA ideas can be applied to solve generic systems of two equations
- Classical work Borkar [1997] showed asymptotic convergence requiring $\lim_{k\to\infty} \frac{\alpha_k}{\beta_k} = 0$
- Only asymptotic almost sure results available; function class is much restricted (e.g., the mapping is smooth), additional assumptions of bounded iterates
- Mokkadem et al. [2006] consider restricted form of nonlinear and smooth mappings; Dalal et al. [2018] considers linear TTSA.

Discussion: TTSA for policy optimization

- TTSA can be easily adopted to solve the policy optimization problem in RL
- It will result in an Actor-Critic-type algorithm
  - Actor (upper-level): policy optimization
  - Critic (lower-level): policy evaluation
- In [H.-Wai-Wang-Yang-23] it has been shown that TTSA specializes to a two-timescale natural actor critic (TT-NAC) algorithm
- Applying TTSA analysis to this setting, we obtain

$$\mathbb{E}[\|\ell(\pi^k) - \ell^\star\|^2] = \mathcal{O}(K^{-1/4})$$

What's Next

The TTSA work so far only scratches the surface; many important theoretical/practical questions remain:

- How good is it in practical applications?
- How to specialize these algorithms under special problem structure?
- Can we make it faster?
- Can we weaken the assumptions?

Faster Algorithms

$$\text{Let} \quad x_{k+1} = x_k - \alpha_k h_f^k, \quad y_{k+1} = y_k - \beta_k h_g^k$$

- Let $\eta_k^g, \eta_k^f \in [0,1]$. Replace the gradient estimates by

$$h_k^g = \eta_k^g \underbrace{G(x_k, y_k; \zeta_k)}_{\text{SGD estimate}} + (1-\eta_k^g) \underbrace{\left(h_{k-1}^g + G(x_k, y_k; \zeta_k) - G(x_{k-1}, y_{k-1}; \zeta_k)\right)}_{\text{SARAH estimate Nguyen et al. [2017]}}$$

$$h_k^f = \eta_k^f \underbrace{F(x_k, y_k; \xi_k)}_{\text{SGD estimate}} + (1-\eta_k^f) \underbrace{\left(h_{k-1}^f + F(x_k, y_k; \xi_k) - F(x_{k-1}, y_{k-1}; \xi_k)\right)}_{\text{SARAH estimate Nguyen et al. [2017]}}$$

- **Convex combination** of SGD & SARAH Cutkosky and Orabona [2019] $\Rightarrow$ the **SUSTAIN** algorithm [Khanduri-Zeng-H.-Wai- Wang-Yang 21].
- With step size $\alpha_k = \beta_k \asymp k^{-1/3}, \quad \eta_k^g = \eta_k^f \asymp k^{-2/3}$,

$$\mathbb{E}[\Delta_x^K] = \mathcal{O}(K^{-2/3}) = \text{'an optimal rate'}$$

Other ways to compute Hyper-gradient

- To approximate the Hessian inverse, one can solve the following quadratic equation:

$$v^*(x) \coloneqq \min_{v \in \mathbb{R}^{d_2}} v^\top \nabla^2_{yy} g(x, y^*(x)) v + \nabla_y f(x, y^*(x))^\top v \quad \textbf{(QD)}$$

- **Hyper-gradient**: $F(x, y^*(x)) \coloneqq \nabla_x f(x, y^*(x)) + \nabla^2_{xy} g(x, y^*(x)) v^*$

- **Algorithms**: Solve **(QD)** using: Conjugate gradient (CG) [Ji et al., 2021], SGD Dagréou et al. [2022], Arbel and Mairal [2022].

Can Hessian computations be avoided?

- **Value function (VF) based approach:** Pose the LL problem as a constraint

$$\min_{x\in\mathbb{R}^{d_1}, y\in\mathbb{R}^{d_2}} f(x,y) \quad \text{s.t.} \quad g(x,y) - g^*(x) \leq 0 \qquad \textbf{(VF)}$$

  where $g^*(x) := g(x, y^*(x))$

- Avoids the computation of second-order derivatives
- Fully first-order algorithms, but with weaker convergence guarantees
- **Algorithms:** First baseline Kwon et al. [2023], lower bounds and improved algorithms Kwon et al. [2024], Chen et al. [2023a].

Sample complexity of SOTA Algorithms

- Many stochastic algorithms have been proposed for bilevel optimization since 2021 (under similar assumptions as TTSA):

Algorithm	Approach	Implementation	Batch Size	Convergence
BSA[1] [Ghadimi and Wang, 2018]	AID	Double loop	$\mathcal{O}(1)$	$\mathcal{O}(\epsilon^{-2})$, $\mathcal{O}(\epsilon^{-3})$
TTSA [H. et al 2023]	AID	Single loop	$\mathcal{O}(1)$	$\mathcal{O}(\epsilon^{-5/2})$
F2SA Kwon et al. [2023]	VF	Double loop	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\epsilon^{-7/2})$
F2BA Chen et al. [2023a], Kwon et al. [2024]	VF	Double loop	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\epsilon^{-3})$
stocBiO [Yang et al., 2021]	QD	Double loop	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\epsilon^{-2})$
AmIGO [Arbel and Mairal, 2022]	QD	Double loop	$\mathcal{O}(\epsilon^{-1})$	$\mathcal{O}(\epsilon^{-2})$
SOBA [Dagréou et al., 2022]	QD	Single loop	$\mathcal{O}(1)$	$\mathcal{O}(\epsilon^{-2})$
ALSET [Chen et al., 2021]	AID	Single loop	$\mathcal{O}(1)$	$\mathcal{O}(\epsilon^{-2})$
SVRB [Guo and Yang, 2021]	AID	Single loop	$\mathcal{O}(1)$	$\mathcal{O}(\epsilon^{-3/2})$
SUSTAIN [Khanduri et al., 2021]	AID	Single loop	$\mathcal{O}(1)$	$\mathcal{O}(\epsilon^{-3/2})$

[1] BSA achieves separate convergence guarantees for UL and LL as illustrated on the left and right, resp.

[2] AID refers to the standard approach to compute the Hessian inverse.

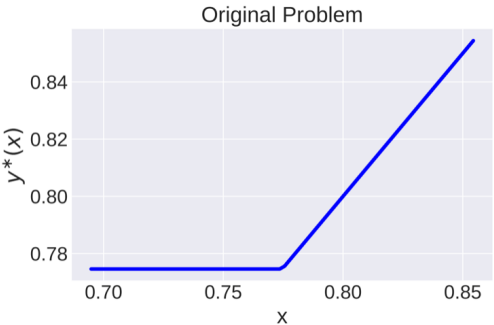
Weaken the Assumptions: Lower Level Constraints

Adding constraints to the lower-level? Then $\ell(x)$ becomes non-smooth

- Smoothing technique [Khanduri et al 23, 25][2]

$$\min_{x \in X \subseteq \mathbb{R}^{d_1}} \bar{\ell}(x) := f(x, \bar{y}^\star(x))$$
$$\text{s.t. } \bar{y}^\star(x) \in \arg\min_{y \in \mathbb{R}^{d_2}, Ay \le b} g(x,y) + \underbrace{q^T y}_{\text{random perturbation}}$$

- The resulting perturbed loss becomes smooth and can be analyzed



[2] A Doubly Stochastically Perturbed Algorithm for Linearly Constrained Bilevel Optimization, P Khanduri, I Tsaknakis, Y Zhang, S Liu, M Hong, arXiv 2025.

Weaken the Assumptions: Lower Level Constraints

Table: Algorithms for linearly constrained LL problems. **Stationarity**: $\|\nabla f\| \leq \epsilon$, **Goldstein**: $(\epsilon, \overline{\delta})$-Goldstein stationarity condition, **Moreau env.**: the gradient of the Moreau envelope. **LE**: Linear equality constraints, **LI**: Linear inequality constraints.

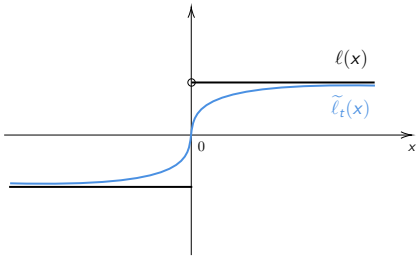
Algorithm	Constraint	Setting	Measure	Convergence
AiPOD Xiao et al. [2023]	LE	Stochastic	Stationarity	$\mathcal{O}(\epsilon^{-4})$
[Lu and Mei, 2024, Algorithm 4]	LI	Deterministic	Stationarity	$\widetilde{\mathcal{O}}(\epsilon^{-7})$ / $\widetilde{\mathcal{O}}(\epsilon^{-4})$
Perturbed Inexact GD Kornowski et al. [2024]	LI	Deterministic	Goldstein	$\mathcal{O}(\epsilon^{-4}\overline{\delta}^{-1})$
[Kornowski et al., 2024, Algorithm 3]	LI	Deterministic	Goldstein	$\mathcal{O}(d_1\epsilon^{-3}\overline{\delta}^{-1})$
[D]SIGD Khanduri et al. [2023]	LI	Deterministic	Stationarity	Asymptotic
[S]SIGD[1] Khanduri et al. [2023]	LI	Stochastic	Moreau env.	$\mathcal{O}(\epsilon^{-4})$
DS-BLO Khanduri et al. [2025]	LI	Stochastic	Goldstein	$\widetilde{\mathcal{O}}(\epsilon^{-4}\overline{\delta}^{-1})$

[1] Under weak convexity assumption.

Weaken the Assumptions: Remove Strong Convexity

- When $g(x,y)$ is not strongly convex in y, $\ell(x)$ can be discontinuous
- In a recent work [Jiang-Li-H.-Zhang 25], we consider the case where the lower-level problem is an LP

$$\min_{x\in\mathcal{X}} \ell(x) := f(x, y^*(x)),$$
$$\text{s.t. } y^*(x) \in \arg\min_{y\in\{y:h_i(x,y)\le 0,\ i=1,\ldots,k\}} g(x,y)$$



- **Solution:** A log-barrier based smooth approximation

$$y_t^*(x) \in \arg\min_{y\in\mathbb{R}^m} \ g(x,y) - t\sum_{i=1}^{k} \log(-h_i(x,y))$$

Jiang, X., Li, J., Hong, M., & Zhang, S. (2024). A Barrier Function Approach for Bilevel Optimization with Coupled Lower-Level Constraints: Formulation, Approximation and Algorithms. arXiv:2410.10670.

Bilevel with LP Lower-level Problem (Approximation)

- Overall problem

$$\min_{x\in\mathcal{X}} \widetilde{\ell}_t(x) := f(x, y^*(x)),$$

$$\text{s.t. } y_t^*(x) \in \arg\min g(x,y) - t\sum_{i=1}^{k} \log(-h_i(x,y))$$

- If $\ell(x)$ is continuous at x, then $\lim_{t\to 0} \widetilde{\ell}_t(x) = \ell(x)$; If $\ell(x)$ is differentiable at x, then $\lim_{t\to 0} \nabla_x \widetilde{\ell}_t(x) = \nabla_x \ell(x)$
- Designed an adaptive algorithm that guarantees $\|\nabla_x \widetilde{\ell}_t(x)\| \le \epsilon$; iteration complexity of $\widetilde{\mathcal{O}}(\epsilon^{-2} t^{-4.5})$
- Overcomes the non-Lipschitz smoothness issue of penalized LL

Weaken the Assumptions: Remove Strong Convexity

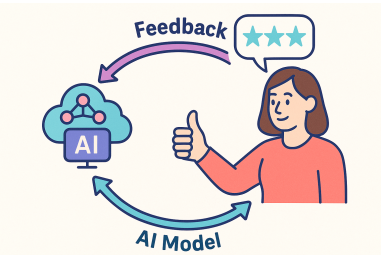
Method	Main Assumption on LL	Rate
IGFM [Chen et al., 2023b]	Convex	$\mathcal{O}(\mathbf{poly}(1/\epsilon))$
[Lu and Mei, 2024, Alg. 4]	Convex	$\widetilde{\mathcal{O}}(\epsilon^{-4})$
V-PBGD [Shen and Chen, 2023]	PL condition	$\widetilde{\mathcal{O}}(\epsilon^{-2})$
[Chen et al., 2025, Algorithm 1]	PL condition	$\widetilde{\mathcal{O}}(\epsilon^{-4})$
PSGD [Masiha et al., 2025]	Local PL condition	$\mathcal{O}(\mathbf{poly}(1/\epsilon))$
SMBG [Bolte et al., 2025a]	Morse type	Asymptotic
TPHSD [Xiao et al., 2025]	No strong assumption	Asymptotic

- Rates are not comparable due to different way of measure optimality, and different assumptions
- **Local PL**: PL condition holds near the stationary point
- **Morse**: every stationary point in y is nondegenerate

Case Study: Aligning LLMs via BLO and TTSA

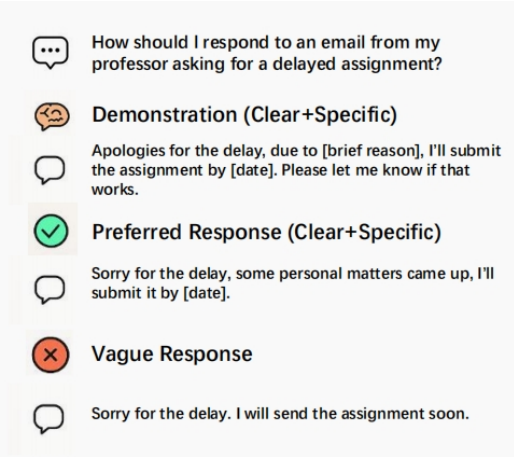
The LLM Alignment Problem

- Recall our previous discussion about improving AI through experience [Silver and Sutton, 2025]
- Human feedback is provided interactively and continuously during training
- A reward model (modeling human preference) is learned over time
- The policy (the AI model parameter) is updated continuously to optimize the reward



The LLM Alignment Problem

- A Special Case: The Large Language Model (LLM) Alignment problem [Ouyang et al 22]
- Ensure that LLM generates texts that conforms human preference and values
- Fixed and offline dataset available, in forms of human demonstrations, preferences, etc.
- Can be modeled by BLO
- **Good News:** Efficient computation of hyper-gradient due to problem structure; Improved alignment results



The RLHF Approach [Ouyang et al 22]

One popular LLM alignment strategy is the Reinforcement Learning with Human Feedback (RLHF); Three main steps (s: state/prompt; a: action/answer):

1. Supervised Fine-tuning (SFT): Collect demonstration data $\{(s, a)\}$ and then train a supervised model from the pre-train model.

2. Reward Model (RM) Training: Collect preference data $\{(s, a_w, a_l)\}$ where $a_w \succ a_l$; train a reward model to classify the preferred response from the non-preferred.

3. Policy Optimization by RL: Given the reward model, optimize the policy by RL.

What's next: BLO for learning from demonstration

Next we plan to:

- Formulate the alignment problem into a BLO
- Explore algorithm design choices
- Results and possible extensions

To simplify things assume for now that only expert demonstration data is available.

Notations

- Data sample $\tau = (s, a)$: a state (prompt) and action (response) pair
- Policy $\pi(a \mid s)$: the probability of choosing an action under state
- Reward $r(s, a; \theta)$: a parameterized function scores a data sample (s, a)
- Expert policy $\pi^E(a \mid s)$: the SFT data is generated from this policy
- **Note:** all results below are applicable to the full MDP setting with multiple stages and interaction with stochastic environment.

BLO for Alignment

- **Task 1 policy optimization**: For a given reward $r(s, a; \theta)$, find the best policy π_θ
- **Task 2 reward estimation**: Find the reward $r(s, a; \theta)$, whose optimal policy matches the expert policy
- The Maximum Likelihood principle: The actions generated by the desired policy should be **most likely** if the ground truth is the expert policy π^E.
- Use BLO to integrate them into a single problem ([Zeng-Li-Garcia-H. 22][3] [Zeng-Garcia-H. 24]) [4]

[3] S. Zeng, C. Li, A. Garcia, & M. Hong. Maximum-likelihood inverse reinforcement learning with finite-time guarantees. **NeurIPS**, 2022.

[4] S. Zeng, M. Hong, A. Garcia, Structural estimation of markov decision processes in high-dimensional state space with finite-time guarantees, **Operations Research**, 2024.

BLO for Alignment

- We consider the following formulation:

$$\max_{\theta} \; L(\theta) := \mathbb{E}_{\tau^{\mathsf{E}} \sim \pi^{\mathsf{E}}} \left[\log \pi_\theta^\star(a \mid s) \right] \qquad \text{(ML-BLO)}$$

$$s.t. \;\; \pi_\theta^\star := \arg\max_{\pi} \; \mathbb{E}_{\tau \sim \pi} \left[r(s, a; \theta) + \mathcal{H}(\pi(\cdot \mid s)) \right]$$

- $\mathcal{H}(\pi(\cdot|s))$ is either a entropy regularizer or KL regularizer
- $\pi_\theta^\star$ denotes the optimal policy when the reward parameter is θ
- **Challenge:** Is the lower-level problem even convex? How to compute the gradient $\nabla L(\theta)$? Hessian computation? Online expert interaction?

Resolve the Challenge: Online to Offline Data

- To avoid online interaction with the expert $\tau \sim \pi^E$, observe the following

$$L(\theta) := \mathbb{E}_{\tau \sim \pi^E}\Big[r(s_t, a_t; \theta)\Big] - \mathbb{E}_{s_0 \sim \rho}\Big[V_\theta(s_0)\Big]$$

  where $V_\theta(s_0)$ denotes the value function for a given reward $r(\cdot; \theta)$.
- Consider the following surrogate loss

$$\widehat{L}(\theta; \mathcal{D}) := \mathbb{E}_{\tau^E \sim \mathcal{D}}\Big[r(s_t, a_t; \theta)\Big] - \mathbb{E}_{s_0 \sim \rho}\Big[V_\theta(s_0)\Big]$$

  where $\mathcal{D}$ is a set of offline demonstration data
- It turns out that we have the following:

$$|L(\theta) - \widehat{L}(\theta; \mathcal{D})| \leq \frac{C_r}{1-\gamma}\sqrt{\frac{\ln(2/\delta)}{2|\mathcal{D}|}}, \text{ with probability } 1-\delta$$

  where C_r is a bound on the reward function.

Resolve the Challenge: Property of the problem

- The policy optimization problem is generally non-convex
- The optimal policy $\pi_\theta^\star$ is unique under each reward function $r(\cdot, \cdot; \theta)$:

$$\pi_\theta^\star(a|s) = \frac{\exp Q_\theta(s,a)}{\sum_{\tilde{a}\in\mathcal{A}} \exp Q_\theta(s,\tilde{a})}$$

where Q_θ is the soft Q-function under the optimal policy $\pi_\theta^\star$

$$Q_\theta(s,a) := r(s,a;\theta) + \mathbb{E}_{(s',a')\sim\pi_\theta^\star}\left[r(s',a';\theta) + \mathcal{H}(\pi_\theta^\star(\cdot|s'))\right]$$

- Need to iteratively compute π and Q to converge to optimal solutions
- But this structure will help us finding closed-form solution for gradients.

Resolve the Challenge: Property of the problem

- Taking the gradient we obtain

$$\begin{aligned}
\nabla_\theta L(\theta) :=& \mathbb{E}_{\tau^{\mathrm{E}}\sim\pi^{\mathrm{E}}}\Big[\nabla_\theta r(s_t, a_t; \theta)\Big] - \mathbb{E}_{s_0\sim\rho}\Big[\nabla_\theta V_\theta(s_0)\Big] \\
=& \mathbb{E}_{\tau^{\mathrm{E}}\sim\pi^{\mathrm{E}}}\Big[\nabla_\theta r(s_t, a_t; \theta)\Big] - \mathbb{E}_{s_0\sim\rho}\Big[\nabla_\theta \log\Big(\sum_{\tilde{a}\in\mathcal{A}} \exp Q_\theta(s_0, \tilde{a})\Big)\Big] \\
=& \mathbb{E}_{\tau^{\mathrm{E}}\sim\pi^{\mathrm{E}}}\Big[\nabla_\theta r(s_t, a_t; \theta)\Big] - \mathbb{E}_{s_0\sim\rho}\Big[\sum_{a\in\mathcal{A}} \pi_\theta(a|s_0)\nabla_\theta Q_\theta(s_0, a)\Big] \\
=& \mathbb{E}_{\tau^{\mathrm{E}}\sim\pi^{\mathrm{E}}}\Big[\nabla_\theta r(s_t, a_t; \theta)\Big] - \mathbb{E}_{\tau\sim\pi_\theta}\Big[\nabla_\theta r(s_t, a_t; \theta)\Big].
\end{aligned}$$

Where the property of the Q function is used

$$V_\theta(s) = \log\Big(\sum_{\tilde{a}\sim\mathcal{A}} \exp Q_\theta(s, \tilde{a})\Big)$$

Approximate Gradient Computation

- Replace the original loss with the surrogate loss, leverage the optimality condition, we obtain a simple gradient expression:

$$\nabla L(\theta) \approx \mathbb{E}_{\tau^{\mathrm{E}} \sim \mathcal{D}} \left[\nabla_\theta r(s, a; \theta) \right] - \mathbb{E}_{\tau \sim \pi_\theta^\star} \left[\nabla_\theta r(s, a; \theta) \right]$$

- Surprisingly simple form! No Hessian computation needed
- **Observation:** The gradient contrasts the reward obtained, by following the experts and moving away from the model's current policy π_θ
- A recent work Yang et al. [2024] has a derivation with more generic loss function
- Note, the second term still depends on the optimal lower-level policy

TTSA-Type Algorithm: A single-step policy optimization

- (**Policy Optimization Step.**) Following TTSA, a single-step soft-policy iteration
- Estimate the soft-Q function $\widehat{Q}(s,a)$
- Update the policy by soft policy iteration [Cen et al 21]:

$$\pi_{k+1}(a|s) \propto \exp\left(\widehat{Q}(s,a)\right), \quad \forall s \in \mathcal{S}, a \in \mathcal{A}.$$

- One step update; $\pi_{k+1}(a \mid s)$ will track the global optimal solution $\pi_\theta^\star(a \mid s)$

TTSA-Type Algorithm: A single-step policy optimization

- (**Reward Optimization Step.**) Recall the gradient expression:

$$\nabla L(\theta) = \mathbb{E}_{\tau^{\mathsf{E}} \sim \pi^{\mathsf{E}}} \Big[\nabla_\theta r(s_t, a_t; \theta) \Big] - \mathbb{E}_{\tau \sim \pi_\theta^\star} \Big[\nabla_\theta r(s_t, a_t; \theta) \Big].$$

Under the current policy π_{k+1}, (biased) stochastic gradient update:

$$\theta_{k+1} := \theta_k + \alpha \Big(h(\theta_k, \tau_k^{\mathsf{E}}) - h(\theta_k, \tau_k) \Big)$$

where $\tau_k^{\mathsf{E}} \sim \mathcal{D}$, $\tau_k \sim \pi_{k+1}$ and $h(\theta, \tau) := \nabla_\theta r(s_t, a_t; \theta)$.

Non-asymptotic Analysis

Theorem (1)

[Zeng-Garcia-H., 2022] By choosing the stepsize $\alpha = \alpha_0 \cdot K^{-\frac{1}{2}}$, it holds:

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|\log\pi_{k+1} - \log\pi_{\theta_k}\|_\infty\right] = \mathcal{O}(K^{-\frac{1}{2}}) + \mathcal{O}(\epsilon_{\text{app}})$$

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left[\|\nabla L(\theta_k)\|^2\right] = \mathcal{O}(K^{-\frac{1}{2}}) + \mathcal{O}(\epsilon_{\text{app}})$$

- Under linear reward parameterization, guarantee of global optimality.

Remarks

- The above work is closely related to a line of research called imitation learning, and inverse reinforcement learning (IRL) [Ziebart et al., 2008, Ross et al., 2011, Pomerleau, 1988]
- How to better learn from expert demonstrations?
- Learning a reward function and a policy together is more generalizable than supervised learning [Ross et al., 2011]
- The above result is first non-asymptotic analysis for IRL algorithm under nonlinear reward parameterization

Application to Fine-Tuning of LLM

- In typical RLHF, SFT step learn from demonstration data using the following:

$$\min_{\pi} \ell_{\mathsf{SFT}}(\pi) := -\mathbb{E}_{\tau \in \mathcal{D}_{\mathsf{SFT}}}[\log(\pi(a \mid s))]$$

  "Clone" the behavior of the expert demonstrator

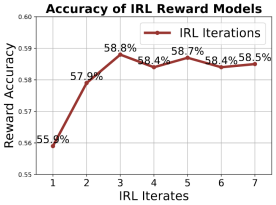
- Learn more generalizable policy? Apply (ML-BLO):

$$\begin{aligned} \min_{\theta} \; & L_{\mathsf{SFT}} = -\mathbb{E}_{\tau \in \mathcal{D}_{\mathsf{SFT}}}[\log(\pi_\theta^\star(a \mid s))] \\ \text{s.t.} \quad & \pi_\theta^\star = \arg\max_{\pi} \mathbb{E}_{(a,x)\sim\pi(\cdot|s)}[r(s,a;\theta) + \mathcal{H}(\pi(\cdot|s))] \end{aligned}$$

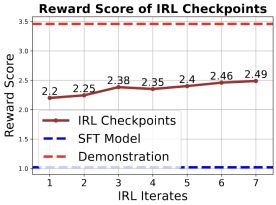
- Algorithm: Alternating between reward update (contrastive learning) and policy update (proximal policy optimization).

Numerical Results [5]

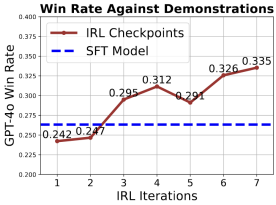
- Train LLM for text summarization tasks using the TL;DR dataset
- Initialize the model by performing SFT on a pretrained 1B parameter Pythia model
- Apply the proposed algorithm (denoted as IRL below), where the RL is done using Proximal Policy Optimization (PPO)



(a) Reward Accuracy



(b) Reward Score



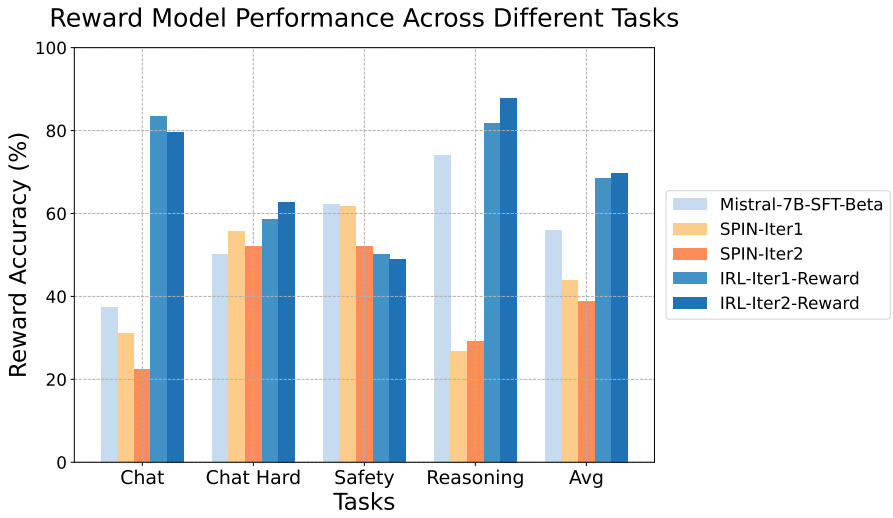
(c) Win Rate

[5] Zeng, S., Liu, Y., Rangwala, H., Karypis, G., Hong, M., & Fakoor, R. (2025). From demonstrations to rewards: Alignment without explicit human preferences.

Numerical Results

- We compare with a few SOTA methods such as SPIN [Chen et al 24] using the UltraChat dataset
- Initial policy Mistral-7b-SFT-Beta 5
- Evaluate both reward models and policy models.

Numerical Results: Reward Bench

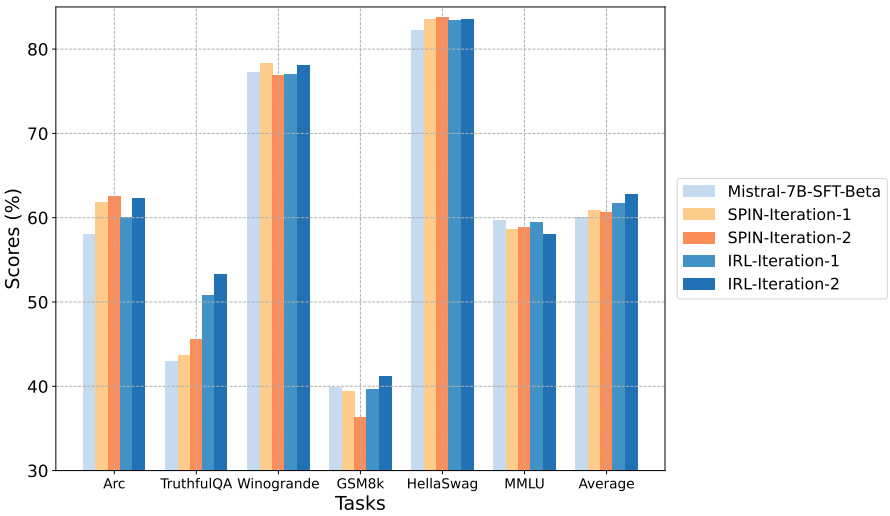


Numerical Results: MT-Bench

- Evaluate different LLM policy models using the Open LLM Leaderboard and MT-Bench

Tasks	First turn	Second turn	Average
`mistral-7b-sft-beta`	5.66	5.09	5.37
`SPIN-Iter1`	6.75	5.56	6.16
`SPIN-Iter2`	3.18	3.41	3.29
`IRL-Iter1-Policy`	6.71	5.96	6.33
`IRL-Iter2-Policy`	**7.01**	**6.19**	**6.60**

Numerical Results: OpenLLM Leaderboard



Extension

- We can further leverage this approach to integrate the three stages of RLHF (SFT, RM, RL) into a single unified stage [Li-Zeng-Li-Garcia-H. 2025] [6]

$$\begin{aligned} \min_{\theta} \quad & L_{\mathsf{SFT}}(\theta; \mathcal{D}_d) + L_{\mathsf{RM}}(\theta; \mathcal{D}_p) \\ \text{s.t.} \quad & \pi_\theta := \arg\max_{\pi} \ \mathbb{E}_{\tau \sim \pi} \left[\left(r(s, a; \theta) + \mathcal{H}(\pi(\cdot|s)) \right) \right] \end{aligned}$$

- The reward and policy learning leverages all the available data
- The algorithm is similar to the two-step bilevel optimization algorithm, alternates between policy optimization and reward optimization.

[6] C Li, S Zeng, Z Liao, J Li, D Kang, A Garcia, M Hong "Learning Reward and Policy Jointly from Demonstration and Preference Improves Alignment", **ICLR (spotlight)**, 2025

Numerical Results: 1B Model for HH dataset

- Policy model: EleutherAI/pythia-1B Reward model: EleutherAI/pythia-1.4B
- Dataset: Anthropic/hh-rlhf; Evaluation: PKU-Alignment/beaver-7b-v3.0-reward

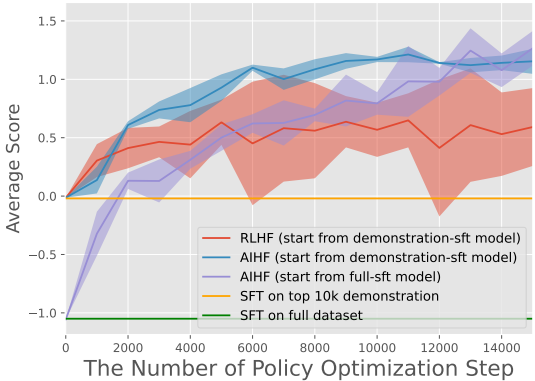


Figure: Helpfulness-controlled Generation on Pythia-1B policy models, where the reward model is trained from Pythia-1.4B models.

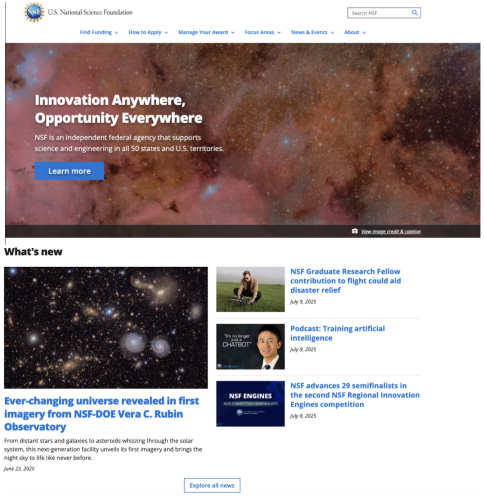
Experiment: Alignment with Demonstration & Preference

- Demonstration Dataset: UltraChat_200k
- Preference Dataset: Ultrafeedback_Binarized

Tasks	Arc Challenge	TruthfulQA MC2	Winogrande	GSM8k	HellaSwag	MMLU	Avg
mistral-7b-sft-beta	54.69%	42.96%	77.27%	39.88%	82.23%	59.72%	59.46%
zephyr-7b-beta	59.64%	55.18%	77.82%	33.51%	84.19%	59.76%	61.68%
SPIN	58.45%	43.66%	78.30%	39.50%	83.59%	58.60%	60.35%
DPO	62.80%	53.17%	**79.40%**	39.20%	85.13%	59.41%	63.19%
IPO	58.02%	48.29%	79.24%	42.91%	83.93%	60.07%	62.08%
AIHF-DPO	61.17%	**60.03%**	79.00%	39.80%	85.71%	60.02%	64.29%
Self-play AIHF	61.77%	58.29%	78.53%	**44.20%**	85.53%	58.66%	64.50%
AIHF	**63.90%**	58.38%	79.24%	40.56%	**86.23%**	**60.18%**	**64.75%**

More Experiments and Discussions

- A tutorial paper for details about applying BLO to LLM alignment [Zeng et al 25] [a]
- An NSF podcast discussing high-level ideas on how to use demonstration data to improve AI



[a] S. Zeng, L. Viano, C. Li, J. Li, V. Cevher, M. Wulfmeier, S. Ermon, A. Garcia, M. Hong, "Aligning Large Language Models with Human Feedback: Mathematical Foundations and Algorithm Design", submitted **IEEE Signal Processing Magazine**, 2025

Conclusions and Future Works

Conclusions and Future Works

Bilevel optimization is a very exciting and vibrant research area, lots of open theoretical and practical problems

- "Universal" algorithms for BLO?
- What kind of generic problem structure we can leverage to simplify implicit gradient computation?
- Finer grained modeling of LLM alignment problems (e.g., multi-turn generation); Enable autonomous self-learning.

Thank You!

References I

Eitaro Aiyoshi and Kiyotaka Shimizu. Hierarchical decentralized systems and its new solution by a barrier method. *IEEE Transactions on Systems, Man and Cybernetics*, (6):444–449, 1981.

Michael Arbel and Julien Mairal. Amortized implicit differentiation for stochastic bilevel optimization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=3PN4iyXBeF.

Omar Ben-Ayed and Charles E Blair. Computational difficulties of bilevel linear programming. *Operations Research*, 38(3):556–560, 1990.

Jérôme Bolte, Quoc-Tung Le, Edouard Pauwels, and Samuel Vaiter. Bilevel gradient methods and morse parametric qualification. *arXiv preprint arXiv:2502.09074*, 2025a.

Jérôme Bolte, Quốc-Tùng Lê, Edouard Pauwels, and Samuel Vaiter. Geometric and computational hardness of bilevel programming. *Mathematical Programming*, pages 1–36, 2025b.

Vivek S Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5):291–294, 1997.

Lori Michelle Case. *An ℓ_1 penalty function approach to the nonlinear bilevel programming problem.* PhD thesis, University of Waterloo, 1998.

He Chen, Jiajin Li, and Anthony Man-cho So. Set smoothness unlocks clarke hyper-stationarity in bilevel optimization. *arXiv preprint arXiv:2506.04587*, 2025.

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

References II

Lesi Chen, Jing Xu, and Jingzhao Zhang. Bilevel optimization without lower-level strong convexity from the hyper-objective perspective. 2023b.

Tianyi Chen, Yuejiao Sun, and Wotao Yin. Closing the gap: Tighter analysis of alternating stochastic gradient methods for bilevel problems. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=OItvP2-i9j.

Nicolas Couellan and Wenjuan Wang. On the convergence of stochastic bi-level gradient methods. *Optimization*, 2016.

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. *Advances in neural information processing systems*, 32, 2019.

Mathieu Dagréou, Pierre Ablin, Samuel Vaiter, and Thomas Moreau. A framework for bilevel optimization that enables stochastic and global variance reduction algorithms. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=wlEOsQ917F.

Gal Dalal, Gugan Thoppe, Balázs Szörényi, and Shie Mannor. Finite sample analysis of two-timescale stochastic approximation with applications to reinforcement learning. In *Conference On Learning Theory*, pages 1199–1233, 2018.

References III

Damek Davis and Dmitriy Drusvyatskiy. Stochastic subgradient method converges at the rate $o(k\hat{}\{-1/4\})$ on weakly convex functions. *arXiv preprint arXiv:1802.02988*, 2018.

James E Falk and Jiming Liu. On bilevel programming, part i: general nonlinear cases. *Mathematical Programming*, 70(1):47–72, 1995.

Simin Fan, Matteo Pagliardini, and Martin Jaggi. Doge: domain reweighting with generalization estimation. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

José Fortuny-Amat and Bruce McCarl. A representation and economic interpretation of a two-level programming problem. *Journal of the operational Research Society*, 32(9):783–792, 1981.

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

Yang Gao, Hong Yang, Peng Zhang, Chuan Zhou, and Yue Hu. Graphnas: Graph neural architecture search with reinforcement learning. *arXiv preprint arXiv:1904.09981*, 2019.

Yulan Gao, Chao Yong, Zehui Xiong, Dusit Niyato, Yue Xiao, and Jun Zhao. A stackelberg game approach to resource allocation for intelligent reflecting surface aided communications. *arXiv preprint arXiv:2003.06640*, 2020.

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

References IV

Zhishuai Guo and Tianbao Yang. Randomized stochastic variance-reduced methods for stochastic bilevel optimization. *arXiv preprint arXiv:2105.02266*, 2021.

Yo Ishizuka and Eitaro Aiyoshi. Double penalty method for bilevel optimization problems. *Annals of Operations Research*, 34(1):73–88, 1992.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International conference on machine learning*, pages 4882–4892. PMLR, 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. *arXiv preprint arXiv:2102.07367*, 2021.

Prashant Khanduri, Ioannis Tsaknakis, Yihua Zhang, Jia Liu, Sijia Liu, Jiawei Zhang, and Mingyi Hong. Linearly constrained bilevel optimization: A smoothed implicit gradient approach. In *International Conference on Machine Learning*, pages 16291–16325. PMLR, 2023.

Prashant Khanduri, Ioannis Tsaknakis, Yihua Zhang, Sijia Liu, and Mingyi Hong. A doubly stochastically perturbed algorithm for linearly constrained bilevel optimization. *arXiv preprint arXiv:2504.04545*, 2025.

Charles D Kolstad and Leon S Lasdon. Derivative evaluation and computational experience with large bilevel mathematical programs. *Journal of optimization theory and applications*, 65(3):485–499, 1990.

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

References V

Guy Kornowski, Swati Padmanabhan, Kai Wang, Zhe Zhang, and Suvrit Sra. First-order methods for linearly constrained bilevel optimization. *arXiv preprint arXiv:2406.12771*, 2024.

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

Jeongyeol Kwon, Dohyun Kwon, and Hanbaek Lyu. On the complexity of first-order methods in stochastic bilevel optimization. *arXiv preprint arXiv:2402.07101*, 2024.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Zhaosong Lu and Sanyou Mei. First-order penalty methods for bilevel optimization. *SIAM Journal on Optimization*, 34(2):1937–1969, 2024.

Zhi-Quan Luo, Jong-Shi Pang, and Daniel Ralph. *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pages 2113–2122. PMLR, 2015.

Saeed Masiha, Zebang Shen, Negar Kiyavash, and Niao He. Superquantile-gibbs relaxation for minima-selection in bi-level optimization. *arXiv preprint arXiv:2505.05991*, 2025.

Abdelkader Mokkadem, Mariane Pelletier, et al. Convergence rate and averaging of nonlinear two-time-scale stochastic approximation algorithms. *The Annals of Applied Probability*, 16(3):1671–1702, 2006.

References VI

Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *International conference on machine learning*, pages 2613–2621. PMLR, 2017.

Jiri Outrata, Michal Kocvara, and Jochem Zowe. *Nonsmooth approach to optimization problems with equilibrium constraints: theory, applications and numerical results*, volume 28. Springer Science & Business Media, 2013.

Rui Pan, Dylan Zhang, Hanning Zhang, Xingyuan Pan, Minrui Xu, Jipeng Zhang, Renjie Pi, Xiaoyu Wang, and Tong Zhang. Scalebio: Scalable bilevel optimization for llm data reweighting. *arXiv preprint arXiv:2406.19976*, 2024.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

Hadi Reisizadeh, Jinghan Jia, Zhiqi Bu, Bhanukiran Vinzamuri, Anil Ramakrishna, Kai-Wei Chang, Volkan Cevher, Sijia Liu, and Mingyi Hong. Blur: A bi-level optimization approach for llm unlearning. *arXiv preprint arXiv:2506.08164*, 2025.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

Gilles Savard and Jacques Gauvin. The steepest descent direction for the nonlinear bilevel programming problem. *Operations Research Letters*, 15(5):265–272, 1994.

References VII

Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. Hydra: Pruning adversarially robust neural networks. *Advances in Neural Information Processing Systems*, 33:19655–19666, 2020.

Han Shen and Tianyi Chen. On penalty-based bilevel gradient descent method. In *International conference on machine learning*, pages 30992–31015. PMLR, 2023.

David Silver and Richard S Sutton. Welcome to the era of experience. *Google AI*, 1, 2025.

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.

H. Stackelberg. *Marktform und Gleichgewicht*. Die Handelsblatt-Bibliothek "Klassiker der Nationalökonomie". J. Springer, 1934. URL https://books.google.com/books?id=wihBAAAAIAAJ.

Heinrich Von Stackelberg. theory of the market economy. *Hodge*, 1952.

Haoran Sun, Wenqiang Pu, Minghe Zhu, Xiao Fu, Tsung-Hui Chang, and Mingyi Hong. Learning to continuously optimize wireless resource in episodically dynamic environment. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4945–4949. IEEE, 2021.

Arash Vahdat, Arun Mallya, Ming-Yu Liu, and Jan Kautz. Unas: Differentiable architecture search meets reinforcement learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11266–11275, 2020.

Nachuan Xiao, Xiaoyin Hu, Xin Liu, and Kim-Chuan Toh. A hybrid subgradient method for nonsmooth nonconvex bilevel optimization. *arXiv preprint arXiv:2505.22040*, 2025.

References VIII

Quan Xiao, Han Shen, Wotao Yin, and Tianyi Chen. Alternating projected sgd for equality-constrained bilevel optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 987–1023. PMLR, 2023.

Chao Xue, Xiaoxing Wang, Junchi Yan, Yonggang Hu, Xiaokang Yang, and Kewei Sun. Rethinking bi-level optimization in neural architecture search: A gibbs sampling perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10551–10559, 2021.

Junjie Yang, Kaiyi Ji, and Yingbin Liang. Provably faster algorithms for bilevel optimization. *arXiv preprint arXiv:2106.04692*, 2021.

Yan Yang, Bin Gao, and Ya-xiang Yuan. Bilevel reinforcement learning via the development of hyper-gradient without lower-level convexity. *arXiv preprint arXiv:2405.19697*, 2024.

Yihua Zhang, Yuguang Yao, Parikshit Ram, Pu Zhao, Tianlong Chen, Mingyi Hong, Yanzhi Wang, and Sijia Liu. Advancing model pruning via bi-level optimization. In *Advances in Neural Information Processing Systems*, 2022.

Yihua Zhang, Prashant Khanduri, Ioannis Tsaknakis, Yuguang Yao, Mingyi Hong, and Sijia Liu. An introduction to bilevel optimization: Foundations and applications in signal processing and machine learning. *IEEE Signal Processing Magazine*, 41(1):38–59, 2024.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.