# A Generic First-Order Algorithmic Framework for Bi-Level Programming Beyond Lower-Level Singleton

## (Paper ID: 1129)

Risheng Liu[1], Pan Mu[1], Xiaoming Yuan[2], **Shangzhi Zeng[2]**, Jin Zhang[3]

[1] Dalian University of Technology
[2] The University of Hong Kong
[3] Southern University of Science and Technology

**ICML 2020**

Thirty-seventh International Conference on Machine Learning

# Bi-Level Programs (BLPs)

We consider the following **BLP** formulation:

- A **hierarchical optimization** problem, where an optimization problem contains another optimization problem **as the constraint**
- In general, solving BLPs is extremely challenging

$$\min_{\mathbf{x}\in\mathcal{X},\mathbf{y}\in\mathbb{R}^m} F(\mathbf{x},\mathbf{y}),\ \ s.t.\ \mathbf{y}\in\mathcal{S}(\mathbf{x}),\ \ \text{where } \mathcal{S}(\mathbf{x}) := \arg\min_{\mathbf{y}} f(\mathbf{x},\mathbf{y})$$

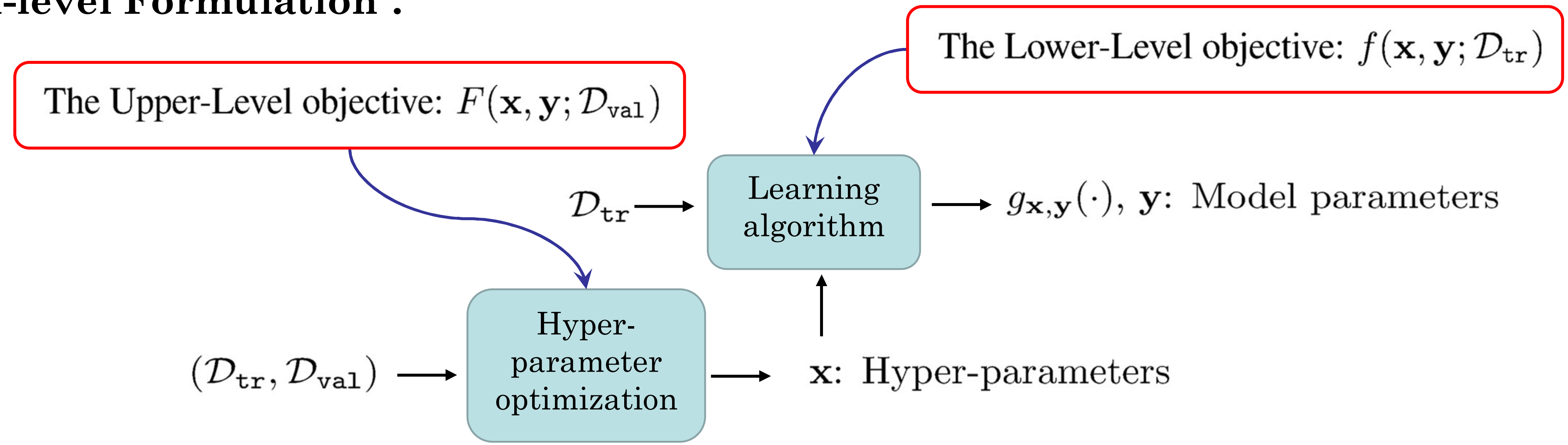
- $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is called Upper-Level (UL) objective
- For every $\mathbf{x}\in\mathcal{X}$, $f(\mathbf{x},\cdot) : \mathbb{R}^m \to \mathbb{R}$ is called Lower-Level (LL) objective

# From single level to bi-level

- **Hyper-parameter optimization**

  Most machine learning problems crucially depend on **some variables that must be decided before learning,** e.g., parameters for regularization, hypothesis space, optimization, preprocessing, etc.

**Bi-level Formulation :**



# From single level to bi-level

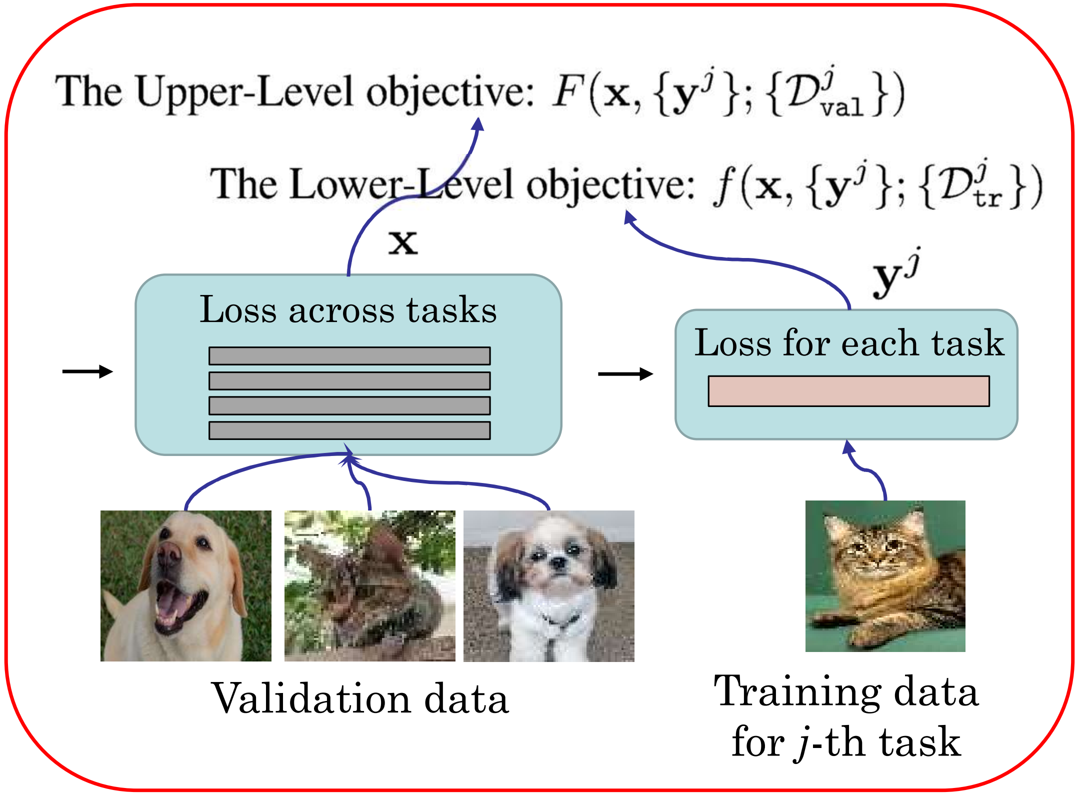
- **Meta learning** (e.g., few-shot classification)

  intends to design models that can learn new skills or adapt to new environments rapidly with a few training examples

- A classifier is a "**learner**" model, trained for operating a given task
- We train it over **a variety of learning tasks** to obtain the best performance **on a distribution of tasks**, including potentially unseen tasks

We denote $\mathcal{D} = \{\mathcal{D}^j\}_{j=1}^N$, and $\mathcal{D}^j = (\mathcal{D}^j_{\mathtt{tr}}, \mathcal{D}^j_{\mathtt{val}})$.

**Bi-level Formulation :**



# Existing first-order bi-level schemes

**Lower-Level Singleton (LLS) assumption**

- Rather than considering the original BLPs, they actually solve a simplification:

$$\boxed{\min_{\mathbf{x}\in\mathcal{X}} F(\mathbf{x},\mathbf{y}),\ \ s.t.\ \mathbf{y}\ \text{``}{=}\text{''}\ \arg\min_{\mathbf{y}} f(\mathbf{x},\mathbf{y})}$$

- **LL subproblem:** A sequence $\mathbf{y}_k$ parameterized by $\mathbf{x}$ is generated, e.g.,

$$\mathbf{y}_{k+1}(\mathbf{x}) = \mathbf{y}_k(\mathbf{x}) - s_l \nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}_k(\mathbf{x})),\ \ k = 0, \cdots, K-1,$$

where $s_l > 0$ is an appropriately chosen step size.

- **UL subproblem:** Incoporate $\mathbf{y}_K(\mathbf{x})$ into $F$ and update $\mathbf{x}$ by $\min_{\mathbf{x}\in\mathcal{X}}\ F(\mathbf{x}, \mathbf{y}_K(\mathbf{x}))$.

**Domke, 2012; Machaurin et al. 2015; Franceschi et al. 2017, 2018; Lorraine et al. 2018; MacKay et al. 2019; Shaban et al. 2019.**

# An interesting counter-example

$$\min_{\mathbf{x}\in[-100,100]} \frac{1}{2}(\mathbf{x}-[\mathbf{y}]_2)^2+\frac{1}{2}([\mathbf{y}]_1-1)^2,$$
$$s.t.\ \mathbf{y}\in\arg\min_{\mathbf{y}\in\mathbb{R}^2} \frac{1}{2}[\mathbf{y}]_1^2-\mathbf{x}[\mathbf{y}]_1.$$

- $[\cdot]_i$ denotes the $i$-th element of the vector.
- $\mathbf{x}\in[-100,100]$ and $\mathbf{y}\in\mathbb{R}^2$.
- The optimal solution is $\mathbf{x}^*=1, \mathbf{y}^*=(1,1)$.

---

- Initialize $\mathbf{y}_0=(0,0)$ and vary step size $s_l^k\in(0,1)$
- $[\mathbf{y}_K]_1=(1-\prod_{k=0}^{K-1}(1-s_l^k))\mathbf{x}$ and $[\mathbf{y}_K]_2=0$
- We have $\mathbf{x}_K^*=\arg\min_{\mathbf{x}\in[-100,100]} F(\mathbf{x},\mathbf{y}_K(\mathbf{x}))$

$$=\frac{(1-\prod_{k=0}^{K-1}(1-s_l^k))}{1+(1-\prod_{k=0}^{K-1}(1-s_l^k))^2}.$$

**Schemes with LLS assumption**

- As $\lim_{K\to\infty}\prod_{k=0}^{K-1}(1-s_l^k)\in[0,1]$
- Then $\lim_{K\to\infty}\frac{(1-\prod_{k=0}^{K-1}(1-s_l^k))}{1+(1-\prod_{k=0}^{K-1}(1-s_l^k))^2}\in[0,\frac{1}{2}]$.
- Thus $\lim_{K\to\infty}\mathbf{x}_K^*\in[0,\frac{1}{2}]$.

$\mathbf{x}_K^*$ cannot converge to $\mathbf{x}^*=1$

# Bi-level Descent Aggregation (BDA)

- **Optimistic Bi-level Algorithmic Framework**
  - $\min_{\mathbf{x}\in\mathcal{X}} \varphi(\mathbf{x}),\ \text{with}\ \varphi(\mathbf{x}) := \inf_{\mathbf{y}\in\mathcal{S}(\mathbf{x})} F(\mathbf{x},\mathbf{y})$
  - $\varphi$: the value function of simple bi-level problem

$$\min_{\mathbf{y}} F(\mathbf{x},\mathbf{y}),\ \ s.t.\ \mathbf{y}\in\mathcal{S}(\mathbf{x}),\ \ (\text{with fixed } \mathbf{x}).$$

Inspired by this observation, we may update $\mathbf{y}$ as

$$\mathbf{y}_{k+1}(\mathbf{x}) = \mathcal{T}_k(\mathbf{x},\mathbf{y}_k(\mathbf{x})),\ \ k=0,\cdots,K-1,$$

where $\mathcal{T}_k(\mathbf{x},\cdot)$ stands for a certain simple bi-level solution strategy with a fixed UL variable $\mathbf{x}$.

# Bi-level Descent Aggregation (BDA)

- **Flexible Iteration Modules**
  - For a given $\mathbf{x}$, the descent directions of the UL and LL objectives are

$$\mathbf{d}_k^F(\mathbf{x}) := s_u \nabla_{\mathbf{y}} F(\mathbf{x}, \mathbf{y}_k), \ \mathbf{d}_k^f(\mathbf{x}) := s_l \nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}_k),$$

    where $s_u, s_l > 0$ are their step size parameters.
  - With aggregation parameter $\alpha_k \in (0, 1]$, we formulate $\mathcal{T}_k$ as

$$\mathbf{y}_{k+1}(\mathbf{x}) = \mathcal{T}_k(\mathbf{x}, \mathbf{y}_k(\mathbf{x})) = \mathbf{y}_k - (\alpha_k \mathbf{d}_k^F(\mathbf{x}) + (1 - \alpha_k)\mathbf{d}_k^f(\mathbf{x})).$$

  - Replacing $\varphi(\mathbf{x})$ by $F(\mathbf{x}, \mathbf{y}_K(\mathbf{x}))$, we have $\min_{\mathbf{x} \in \mathbf{X}} \varphi_K(\mathbf{x}) := F(\mathbf{x}, \mathbf{y}_K(\mathbf{x}))$ where $\mathbf{y}_K(\mathbf{x})$ is the output after $K$ iterations.

**BDA is flexible enough to incorporate a variety of numerical schemes!**

# Theoretical investigations

- **A General Proof Recipe**

(1) **LL solution set property:** For any $\epsilon > 0$, there exists $k(\epsilon) > 0$ such that whenever $K > k(\epsilon)$,

$$\sup_{\mathbf{x}\in\mathcal{X}} \mathtt{dist}(\mathbf{y}_K(\mathbf{x}), \mathcal{S}(\mathbf{x})) \leq \epsilon.$$

(2) **UL objective convergence property:** $\varphi(\mathbf{x})$ is LSC on $\mathcal{X}$,

$$\lim_{K\to\infty} \varphi_K(\mathbf{x}) \to \varphi(\mathbf{x}), \ \forall \mathbf{x} \in \mathcal{X}.$$

**LSC: Lower/Upper Semi-Continuous.**

# Theoretical investigations

- **A General Proof Recipe**

**Theorem 1:** Suppose both the above LL solution set and UL objective convergence properties hold, then for $\mathbf{x}_K \in \arg\min_{\mathbf{x}\in\mathcal{X}} \varphi_K(\mathbf{x})$, we have

(1) Any limit point $\bar{\mathbf{x}}$ of the sequence $\{\mathbf{x}_K\}$ satisfies that $\bar{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathcal{X}} \varphi(\mathbf{x})$;

(2) $\inf_{\mathbf{x}\in\mathcal{X}} \varphi_K(\mathbf{x}) \to \inf_{\mathbf{x}\in\mathcal{X}} \varphi(\mathbf{x})$ as $K \to \infty$.

**Remark:** If $\mathbf{x}_K$ is local minimum of $\varphi_K(\mathbf{x})$ with uniform neighbourhood modulus, then any limit point $\bar{\mathbf{x}}$ of the sequence $\{\mathbf{x}_K\}$ is a local minimum of $\varphi$.

# Theoretical investigations

- **Convergence Properties of BDA**

**Theorem 2.** Suppose $F(\mathbf{x},\cdot)$ is $L_0$-Lipschitz continuous, $L_F$-smooth, and $\sigma$-strongly convex, $f(\mathbf{x},\cdot)$ is $L_f$-smooth and convex for any $\mathbf{x} \in \mathcal{X}$. Let $s_l \in (0, 1/L_f]$, $s_u \in (0, 2/(L_F+\sigma)]$, $\alpha_k = \min\{2\gamma/k(1-\beta), 1\}$, $k \geq 1$ with $\gamma \in (0, 1]$ and $\beta = \sqrt{1 - 2s_u\sigma L_F/(\sigma + L_F)}$. Assume further that $\mathcal{S}(\mathbf{x})$ is continuous on $\mathcal{X}$. Then we have the same convergence results as that in **Theorem 1.**

**Remark**: When the LL objective takes a composite form, e.g., $h = f + g$ with smooth $f$ and nonsmooth $g$, we can adopt the proximal operator based iteration module to construct $\mathcal{T}_k$ and **Theorem 2** still holds.

# Theoretical investigations

- **Improving Existing Theories in the LLS Scenario**

**Theorem 3.** Suppose $\mathcal{S}(\mathbf{x})$ is singleton for any $\mathbf{x} \in \mathcal{X}$. $f(\mathbf{x}, \mathbf{y})$ is level-bounded w.r.t. $\mathbf{y}$ and locally uniform w.r.t. $\mathbf{x}$; $\{\mathbf{y}_K(\mathbf{x})\}$ is uniformly bounded on $\mathcal{X}$, and $\{f(\mathbf{x}, \mathbf{y}_K(\mathbf{x}))\}$ converges uniformly to $f^*(\mathbf{x})$ on $\mathcal{X}$ as $K \to \infty$. Then concerning $\{\mathbf{x}_K\}_{t\in\mathbb{N}}$ and $\{\varphi_K(\mathbf{x})\}$, we have the same convergence results as that in **Theorem 1.**

- We take the gradient-based bi-level scheme to illustrate our improvement, i.e., $\mathbf{y}_{k+1} = \mathbf{y}_k - s_l \nabla_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}_k),\ k = 0, \cdots, K-1$.
- We can immediately verify our required assumption on $\{f(\mathbf{x}, \mathbf{y}_K(\mathbf{x}))\}$ in the absence of strong convexity for $f$.

# Existing bi-level FOMs vs BDA

| Alg. | | LLS | w/o LLS |
|---|---|---|---|
| Existing FOMs | **UL** | $F(\mathbf{x},\mathbf{y})$ is **JC**, $F(\mathbf{x},\cdot)$ is **LC** | Not Available |
| | **LL** | $f(\mathbf{x},\mathbf{y})$ is **JC**, $\mathbf{y}_K(\mathbf{x}) \xrightarrow{u} \mathbf{y}^*(\mathbf{x})$ | |
| | Main Results: $\varphi_K(\mathbf{x}_K) \to \inf_{\mathbf{x}\in\mathbf{X}} \varphi(\mathbf{x})$, $\mathbf{x}_K \xrightarrow{s} \mathbf{x}^*$ | | |
| BDA | **UL** | $F(\mathbf{x},\mathbf{y})$ is **JC**, $F(\mathbf{x},\cdot)$ is **LC** | $F(\mathbf{x},\mathbf{y})$ is **JC**, $F(\mathbf{x},\cdot)$ is **LC**, $\mathrm{L}_F$ Smooth and **SC** |
| | **LL** | $f(\mathbf{x},\mathbf{y})$ is **JC**, $f(\mathbf{x},\cdot)$ is **LB**, $f(\mathbf{x},\mathbf{y}_K(\mathbf{x})) \xrightarrow{u} f^*(\mathbf{x})$ | $f(\mathbf{x},\mathbf{y})$ is **JC**, $f(\mathbf{x},\cdot)$ is $\mathrm{L}_f$ Smooth, $\mathcal{S}(\mathbf{x})$ is Continuous. |
| | Main Results: $\varphi_K(\mathbf{x}_K) \to \inf_{\mathbf{x}\in\mathbf{X}} \varphi(\mathbf{x})$, $\mathbf{x}_K \xrightarrow{s} \mathbf{x}^*$ | | |

- Denote $\xrightarrow{s}$ and $\xrightarrow{u}$ as subsequentially convergent and uniformly convergent respectively. **JC**: Jointly Continuous. **LC**: Lipschitz Continuous. **SC**: Strongly Convex. **LB**: Level-Bounded.
- Existing FOMs: Domke 2012; Maclaurin et al. 2015; Franceschi et al. 2017,2018; Shaban et al. 2019, etc.

# Numerical verifications

- **Compare with gradient-based methods**

- Exact solution:
  $\mathbf{x}^* = 1,\ \mathbf{y}^* = (1,1).$
- RHG solution:
  $\lim_{K\to\infty} \mathbf{x}_K^* \in (0, \frac{1}{2}),$
  $[\mathbf{y}_K]_1 = (1 - \prod_{k=0}^{K-1}(1 - s_l^k))\mathbf{x},$
  $[\mathbf{y}_K]_2 = 0.$
- Our BDA solution:
  $\mathbf{x}_K^* \to 1,\ \mathbf{y}_K^* \to (1,1).$

- Initialization
  $\mathbf{x}_0 = 0,$
  $\mathbf{y}_0 = (0,0),$
- No. of LL Iter.
  $K = 16.$

- Initialization
  $\mathbf{x}_0 = 0,$
  $\mathbf{y}_0 = (2,2).$

**Fig 1.** RHG vs BDA

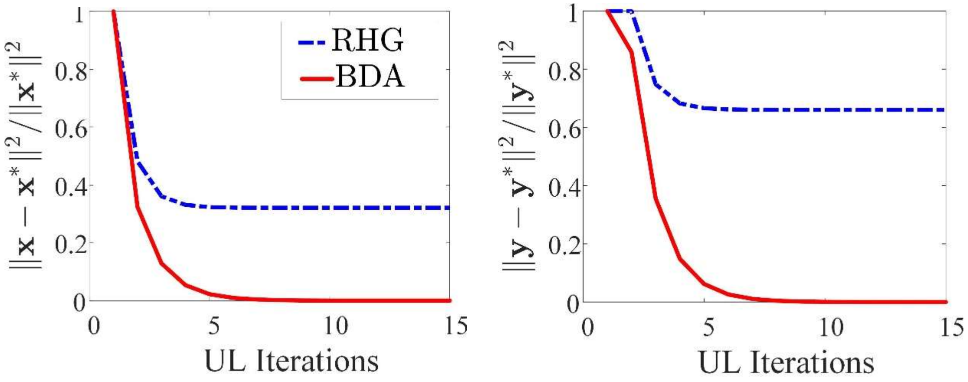
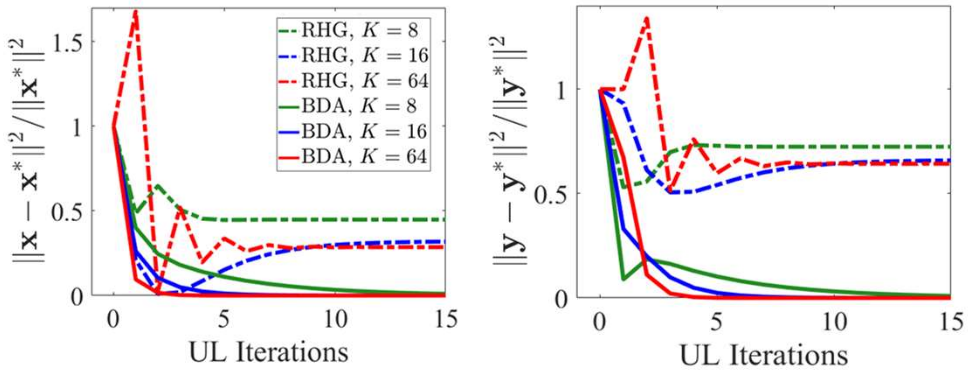



**Fig 2.** BLP with different K.



# Machine learning applications

- **Hyper-parameter optimization** (Data hyper-cleaning)
  - The UL objective $F$ measures the cross-entropy errors with regularization on validation set:

    $F(\mathbf{x},\mathbf{y}) = \sum_{(\mathbf{u}_i,\mathbf{v}_i)\in\mathcal{D}_{\mathtt{val}}} \ell(\mathbf{y}(\mathbf{x});\mathbf{u}_i,\mathbf{v}_i) + \nu\|\mathbf{y}(\mathbf{x})\|^2.$
  - The LL objective $f$ is defined as the weighted cross-entropy loss:

    $f(\mathbf{x},\mathbf{y}) = \sum_{(\mathbf{u}_i,\mathbf{v}_i)\in\mathcal{D}_{\mathtt{tr}}} [\sigma(\mathbf{x})]_i \ell(\mathbf{y};\mathbf{u}_i,\mathbf{v}_i).$

---

- **Dataset:** MNIST (LeCun et al., 1998)
- **SOTA methods:**

  **RHG** (Franceschi et al. 2017, 2018)

  **T-RHG** (Shaban et al. 2019)
- For T-RHG, we set the number of truncated BPs as 25

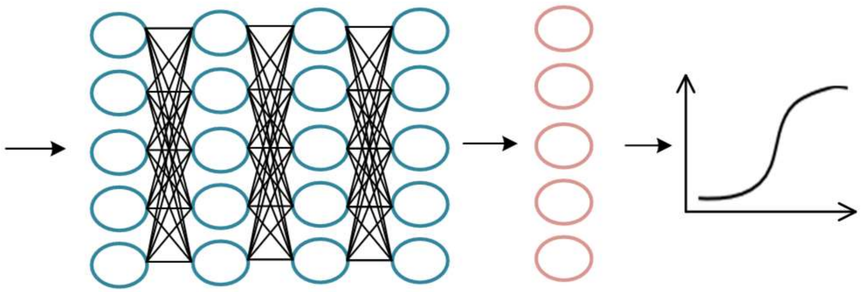
**Table 1.** Accuracy of data hyper-cleaning.

| Method | No. of LL Iterations ($K$) | | | |
|---|---|---|---|---|
| | 50 | 100 | 200 | 800 |
| RHG | 88.96 | 89.73 | 90.13 | 90.15 |
| T-RHG | 87.90 | 88.28 | 88.50 | 89.99 |
| BDA | **89.12** | **90.12** | **90.57** | **90.86** |

# Machine learning applications

- **Meta Learning** (Few-shot classification)

The UL objective: $F(\mathbf{x}, \{\mathbf{y}^j\}) = \sum_j \ell(\mathbf{x}, \mathbf{y}^j; \mathcal{D}^j_{\mathtt{val}})$,
The LL objective: $f(\mathbf{x}, \{\mathbf{y}^j\}) = \sum_j \ell(\mathbf{x}, \mathbf{y}^j; \mathcal{D}^j_{\mathtt{tr}})$.



- **Dataset:** Ominglot (Lake et al. 2015)
- **Setup:**

4 layers CNN (64 filters, with the size 3*3) followed by fully connected layer (Franceschi et al. 2018)

- **SOTA methods:**

**RHG** (Franceschi et al. 2017, 2018), **T-RHG** (Shaban et al. 2019), **MAML** (Finn et al. 2017), **Meta-SGD** (Li et al. 2018), **Reptile** (Nichol et al. 2018)

**Table 2.** Accuracy of few-shot learning.

| Method | 5 way | | 20 way | |
|---|---|---|---|---|
| | 1 shot | 5 shot | 1 shot | 5 shot |
| MAML | 98.70 | **99.91** | 95.80 | 98.90 |
| Meta-SGD | 97.97 | 98.96 | 93.98 | 98.40 |
| Reptile | 97.68 | 99.48 | 89.43 | 97.12 |
| RHG | 98.60 | 99.50 | 95.50 | 98.40 |
| T-RHG | 98.74 | 99.52 | 95.82 | 98.95 |
| BDA | **99.04** | 99.62 | **96.50** | **99.10** |

# Take home message

- **A counter-example** explicitly indicates the importance of the Lower-Level Singleton (LLS) condition for existing bi-level FOMs.
- By formulating BLPs from **the view point of optimistic bi-level**, BDA provides a generic bi-level algorithmic framework
- We strictly prove the convergence of BDA for general BLPs **without the LLS condition**.
- As a nontrivial byproduct, we revisit and **improve the convergence justification** of existing gradient-based schemes for BLPs in the LLS scenario.

# Thanks for your attention

A Generic First-Order Algorithmic Framework for
Bi-Level Programming Beyond Lower-Level Singleton

(Paper ID: 1129)