Finite-Sum Coupled Compositional Optimization

Theories and Practical Applications

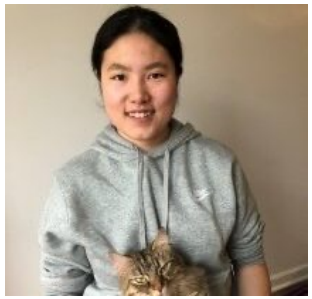
Tianbao Yang

CSE@TAMU

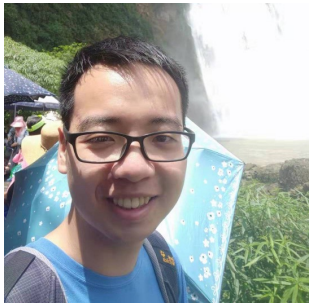
Acknowledgements

Bokun Wang

Zhuoning Yuan
(Netflix)



Quanqi Hu



Zihao Qiu
(Nanjing Univ.)

Dixian Zhu
(Stanford)

Qi Qi

Focus on Optimization

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i)$$

ERM

Focus on Optimization

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i)$$

~~ERM~~

Focus on Optimization

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}; \mathbf{x}_i)$$

~~ERM~~

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i))$$

$$g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i) = \frac{1}{|\mathcal{S}_i|}\sum_{\mathbf{z}_j \in \mathcal{S}_i} \ell(\mathbf{w}; \mathbf{x}_i, \mathbf{z}_j)$$

Finite-sum Coupled Compositional Optimization (FCCO)

Focus on Optimization

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i)$$

~~ERM~~

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i))$$

X-risk

$$g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i) = \frac{1}{|\mathcal{S}_i|}\sum_{\mathbf{z}_j \in \mathcal{S}_i} \ell(\mathbf{w}; \mathbf{x}_i, \mathbf{z}_j) \longrightarrow g_i(\mathbf{w})$$

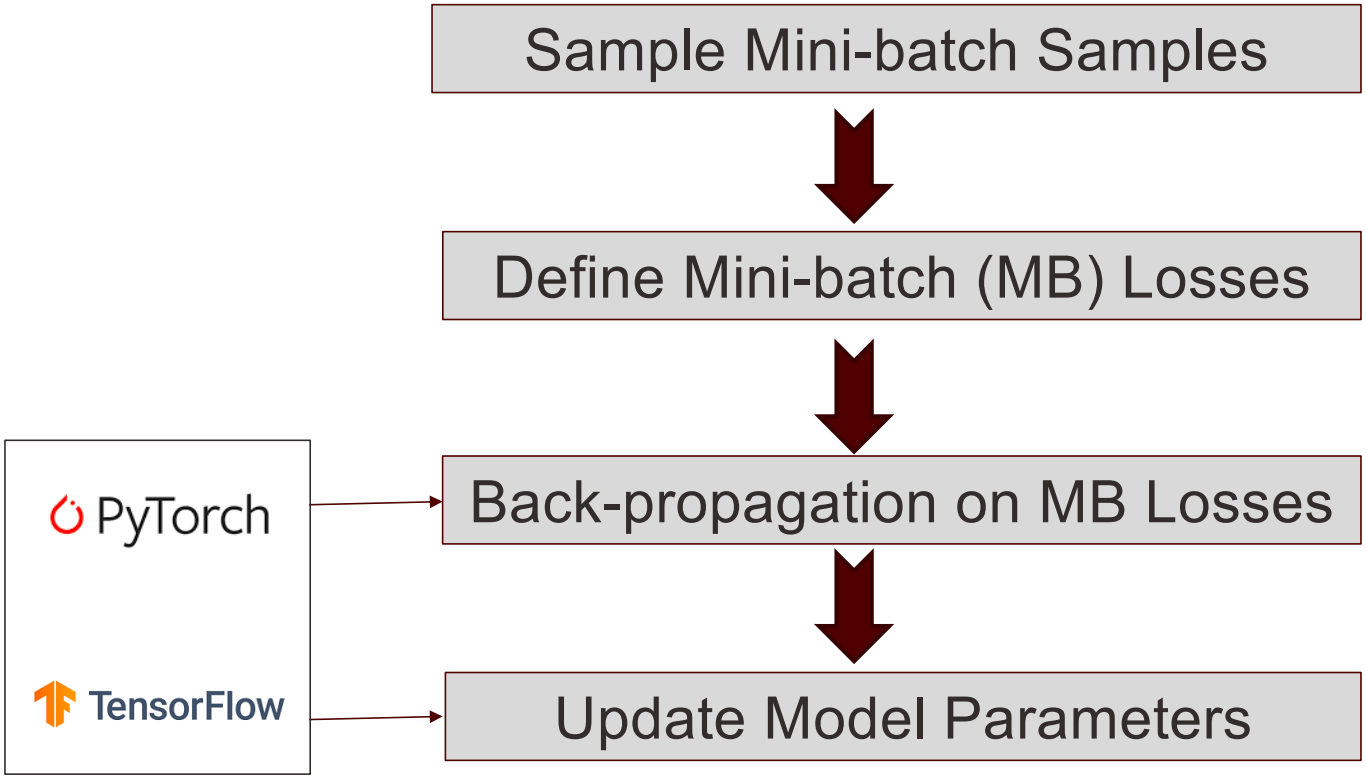
Finite-sum Coupled Compositional Optimization (FCCO)

Outline

- **Motivation & Examples**
- **Challenges & Baselines**
- **Algorithms & Theories**
- **Applications in ML&AI**

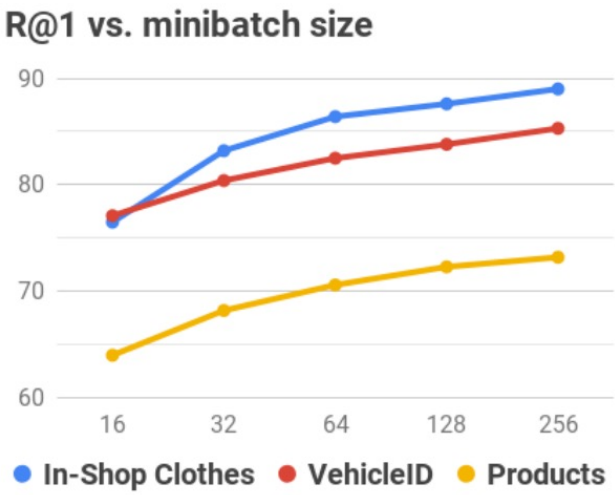
Motivation & Examples

Standard DL Pipeline



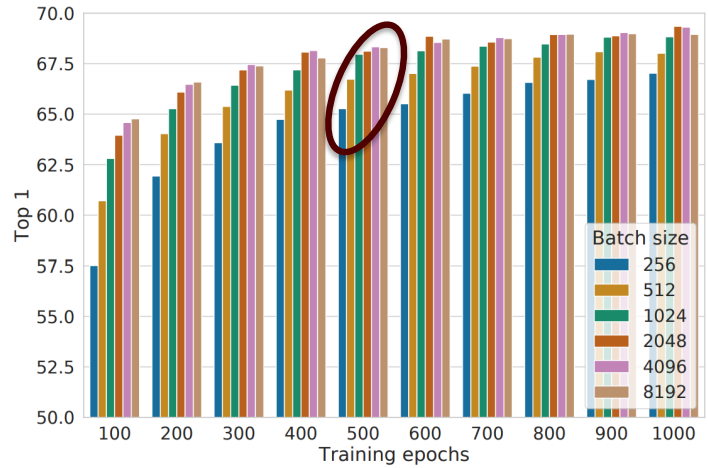
Non-Convergence Issue

Optimizing ranking measures



FastAP: Cakir et al. 2019

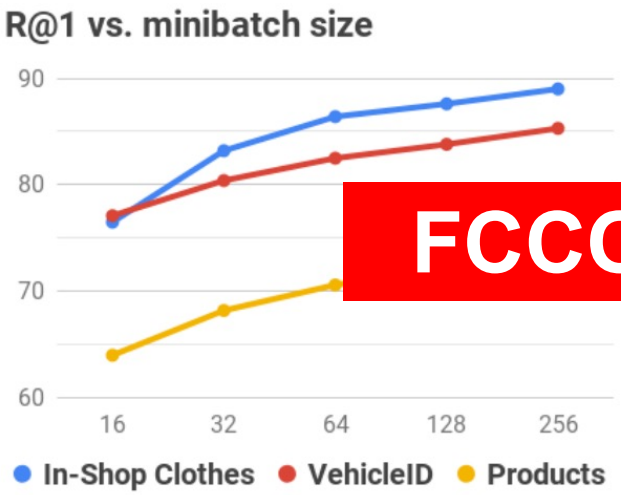
Self-supervised Learning



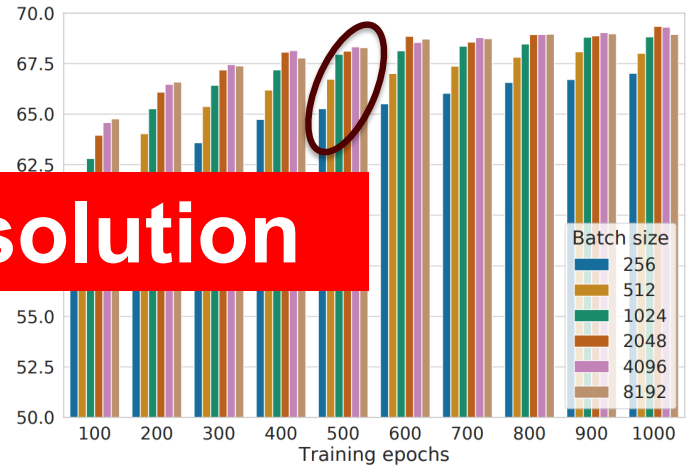
SimCLR: Chen et al. 2020.

Non-Convergence Issue

Optimizing ranking measures



Self-supervised Learning



FCCO is a solution

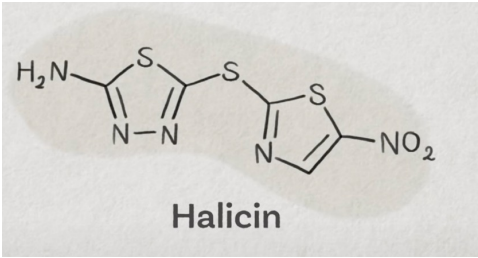
FastAP: Cakir et al. 2019

SimCLR: Chen et al. 2020.

Molecular Property Prediction

MIT AICures Challenge

Fighting Secondary Effects of Covid



Stokes et al. 2020. Cell.

Evaluation Metric: **AUPRC**

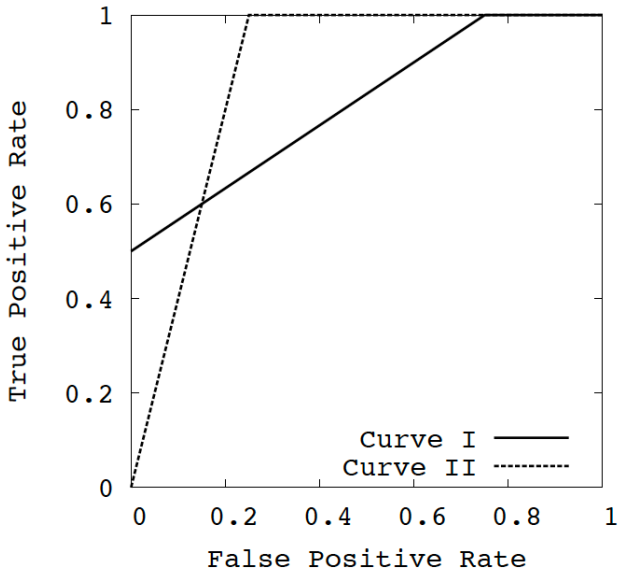
(a) Test PRC-AUC

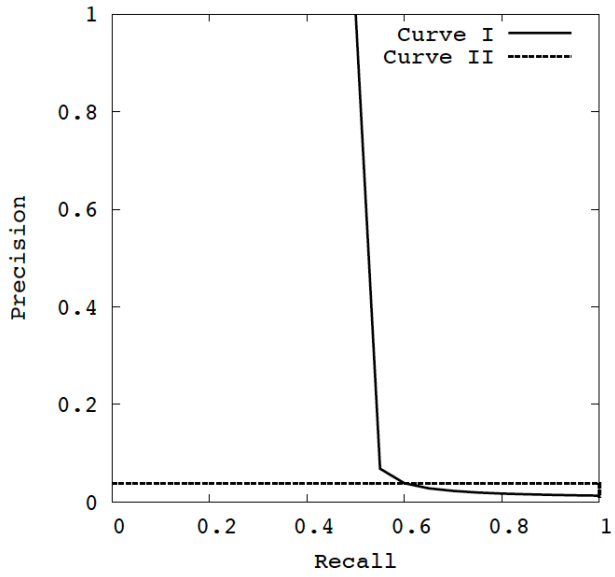
Rank	Model	Author	Submissions	Test PRC-AUC
1	MolecularG	AIDrug@PA	7	0.725
2	-	AGL Team	20	0.702
3	MoleculeKit	DIVE@TAMU	7	0.677
4	GB	BI	6	0.67
5	Chemprop ++	AICures@MIT	4	0.662
6	-	Mingjun Liu	3	0.657
7	Pre-trained OGB-GIN (ensemble)	Weihua Hu@Stanford	2	0.651
8	RF + fingerprint	Cyrus Maher@Vir Bio	1	0.649
9	Graph Self-supervised Learning	SJTU_NRC_Mila	3	0.622
10	-	Congjie He	10	0.611

(b) Test ROC-AUC

Rank	Model	Author	Submissions	Test ROC-AUC
1	MoleculeKit	DIVE@TAMU	7	0.928
2	Chemprop ++	AICures@MIT	4	0.877
3	-	Gianluca Bontempi	7	0.848
4	-	Apoorv Umang	1	0.84
5	Pre-trained OGB-GIN (ensemble)	Weihua Hu@Stanford	2	0.837
6	-	Kexin Huang	1	0.824
7	Chemprop	Rajat Gupta	7	0.818
8	MLP	IITM	7	0.807
9	Graph Self-supervised Learning	SJTU_NRC_Mila	3	0.8
10	-	Congjie He	10	0.8

AUPRC vs AUROC





AUPRC: Highly Imbalanced Data

Estimator: Average Precision (AP)

$$\mathrm{AP}(h) = \boxed{\frac{1}{n_+} \sum_{\mathbf{x}_i \in \mathcal{S}_+}} \mathrm{Precision}(\boxed{h(\mathbf{x}_i)})$$

Estimator: Average Precision (AP)

$$\mathrm{AP}(h) = \frac{1}{n_+} \sum_{\mathbf{x}_i \in \mathcal{S}_+} \boxed{\mathrm{Precision}(h(\mathbf{x}_i))}$$

Positive Examples

$$\mathrm{Precision}(h(\mathbf{x}_i)) = \frac{\sum_{\mathbf{x}_j \in \mathcal{S}_+} \mathbb{I}(h(\mathbf{x}_j) \geq h(\mathbf{x}_i))}{\sum_{\mathbf{x}_j \in \mathcal{S}} \mathbb{I}(h(\mathbf{x}_j) \geq h(\mathbf{x}_i))}$$

All Examples

AP surrogate loss is X-risk

Precision

$$\frac{\sum_{\mathbf{x}_j \in \mathcal{S}_+} \ell(h_{\mathbf{w}}(\mathbf{x}_j) - h_{\mathbf{w}}(\mathbf{x}_i))}{\sum_{\mathbf{x}_j \in \mathcal{S}} \ell(h_{\mathbf{w}}(\mathbf{x}_j) - h_{\mathbf{w}}(\mathbf{x}_i))}$$

$\longrightarrow [g_i(\mathbf{w})]_1$

$\longrightarrow [g_i(\mathbf{w})]_2$

Qi et al. NeurIPS'21

AP surrogate loss is X-risk

Precision

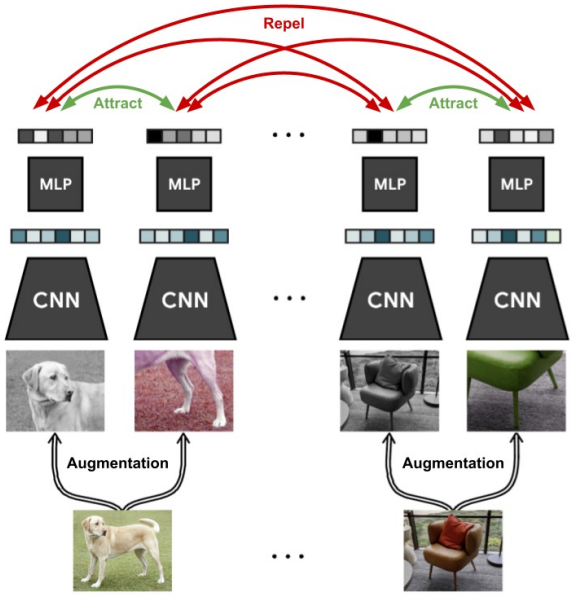
$$\frac{\sum_{\mathbf{x}_j \in \mathcal{S}_+} \ell(h_{\mathbf{w}}(\mathbf{x}_j) - h_{\mathbf{w}}(\mathbf{x}_i))}{\sum_{\mathbf{x}_j \in \mathcal{S}} \ell(h_{\mathbf{w}}(\mathbf{x}_j) - h_{\mathbf{w}}(\mathbf{x}_i))} \longrightarrow [g_i(\mathbf{w})]_1 \longrightarrow [g_i(\mathbf{w})]_2$$

$$f(g) = -\frac{[g]_1}{[g]_2}$$

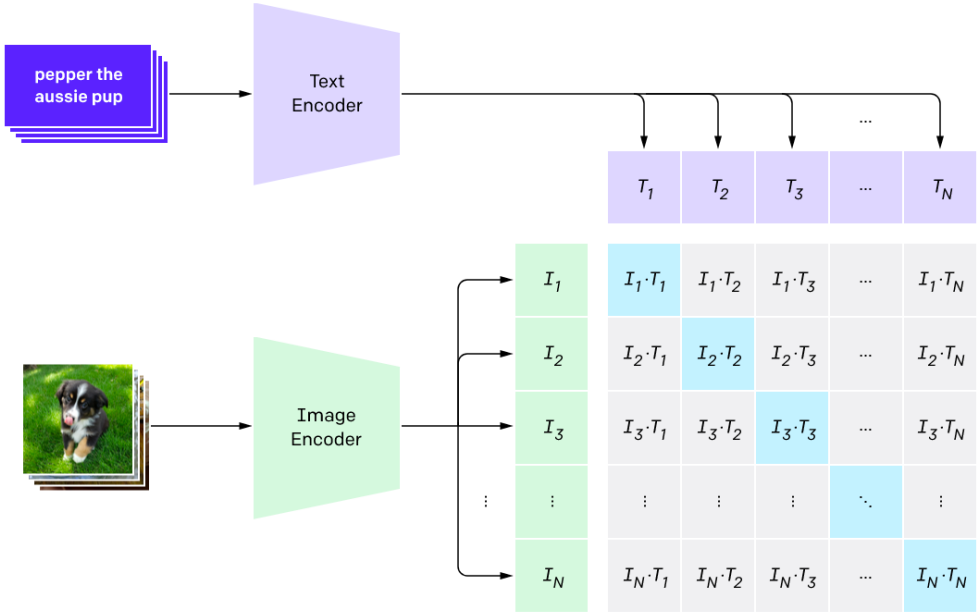
$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{n_+} \sum_{\mathbf{x}_i \in \mathcal{S}_+} f(g_i(\mathbf{w}))$$

Qi et al. NeurIPS'21

Contrastive Self-supervised Learning

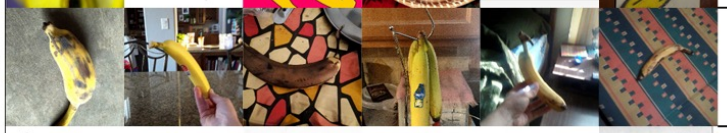
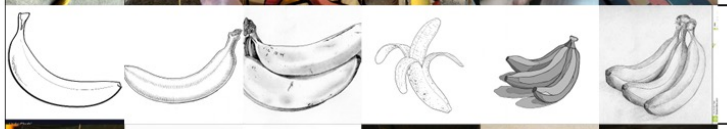


Google's SimCLR (Chen et al. '20)



OpenAI's CLIP (Radford et al. '21)

CLIP: Zero-shot Classification

	Dataset Examples	ImageNet ResNet101	Zero-Shot CLIP	Δ Score
ImageNet		**76.2**	**76.2**	0%
ImageNetV2		64.3	**70.1**	+5.8%
ImageNet-R		37.7	**88.9**	+51.2%
ObjectNet		32.6	**72.3**	+39.7%
ImageNet Sketch		25.2	**60.2**	+35.0%
ImageNet-A		2.7	**77.1**	+74.4%

Global Contrastive Loss is X-risk

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell_{\mathrm{GCL}}(\mathbf{x}_i)$$

$$\ell_{\mathrm{GCL}}(\mathbf{x}_i) = -\tau \log \left(\frac{\exp(h_{\mathbf{w}}(\mathbf{x}_i)^\top h_{\mathbf{w}}(\mathbf{x}_i^+)/\tau)}{\sum_{\mathbf{x}_j \sim \mathcal{S}_i^-} \exp(h_{\mathbf{w}}(\mathbf{x}_i)^\top h_{\mathbf{w}}(\mathbf{x}_j)/\tau)} \right)$$

Yuan et al. ICML'22

Global Contrastive Loss is X-risk

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell_{\mathrm{GCL}}(\mathbf{x}_i)$$

$$\ell_{\mathrm{GCL}}(\mathbf{x}_i) = -\tau \log \left(\frac{\exp(h_{\mathbf{w}}(\mathbf{x}_i)^\top h_{\mathbf{w}}(\mathbf{x}_i^+)/\tau)}{\boxed{\sum_{\mathbf{x}_j \sim \mathcal{S}_i^-} \exp(h_{\mathbf{w}}(\mathbf{x}_i)^\top h_{\mathbf{w}}(\mathbf{x}_j)/\tau)}} \right)$$

$$f(g_i(\mathbf{w})) = \tau \log(g_i(\mathbf{w}))$$

Yuan et al. ICML'22

Challenges & Baselines

Stochastic Gradient is Expensive

$$\nabla f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i))\nabla g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i) \Rightarrow O(|\mathcal{S}_i|)$$

Baseline: SGD

Stochastic Gradient is Expensive

$$\nabla f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i))\nabla g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i) \Rightarrow O(|\mathcal{S}_i|)$$

Baseline: SGD

	Iteration Complexity	Complexity Per-iteration	Total Complexity
Convex (C)	$O(\epsilon^{-2})$	$\max_i O(\lvert\mathcal{S}_i\rvert)$	$\max_i O(\lvert\mathcal{S}_i\rvert\epsilon^{-2})$
Non-Convex (NC)	$O(\epsilon^{-4})$	$\max_i O(\lvert\mathcal{S}_i\rvert)$	$\max_i O(\lvert\mathcal{S}_i\rvert\epsilon^{-4})$

Stochastic Gradient is Expensive

$$\nabla f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i))\nabla g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i) \Rightarrow O(|\mathcal{S}_i|)$$

Baseline: SGD **Not Practical**

	Iteration Complexity	Complexity Per-iteration	Total Complexity
Convex (C)	$O(\epsilon^{-2})$	$\max_i O(\lvert\mathcal{S}_i\rvert)$	$\max_i O(\lvert\mathcal{S}_i\rvert\epsilon^{-2})$
Non-Convex (NC)	$O(\epsilon^{-4})$	$\max_i O(\lvert\mathcal{S}_i\rvert)$	$\max_i O(\lvert\mathcal{S}_i\rvert\epsilon^{-4})$

Biased Stochastic Gradient

$$\nabla f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{B}_i))\nabla g(\mathbf{w}, \mathbf{x}_i, \mathcal{B}_i)$$

$$\mathbb{E} \neq$$

Mini-batch

$$\nabla f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i))\nabla g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i)$$

Baseline: BSGD (Hu et al.'19)

Biased Stochastic Gradient

$$\nabla f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{B}_i))\nabla g(\mathbf{w}, \mathbf{x}_i, \mathcal{B}_i)$$

$$\mathbb{E} \neq$$

Mini-batch

$$\nabla f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i))\nabla g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i)$$

Baseline: BSGD (Hu et al.'19)

	Iteration Complexity	Complexity Per-iteration	Total Complexity
Convex (C)	$O(\epsilon^{-2})$	$O(\epsilon^{-1})$	$O(\epsilon^{-3})$
Non-Convex (NC)	$O(\epsilon^{-4})$	$O(\epsilon^{-2})$	$O(\epsilon^{-6})$

Biased Stochastic Gradient

$$\nabla f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{B}_i))\nabla g(\mathbf{w}, \mathbf{x}_i, \mathcal{B}_i)$$

$$\mathbb{E} \neq$$

Mini-batch

Large Batch Size

$$\nabla f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i))\nabla g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i)$$

Baseline: BSGD (Hu et al.'19)

	Iteration Complexity	Complexity Per-iteration	Total Complexity
Convex (C)	$O(\epsilon^{-2})$	$O(\epsilon^{-1})$	$O(\epsilon^{-3})$
Non-Convex (NC)	$O(\epsilon^{-4})$	$O(\epsilon^{-2})$	$O(\epsilon^{-6})$

History

- **Stochastic Compositional Optimization** $\mathbb{E}_\zeta f_\zeta(\mathbb{E}_\xi[g_\xi(\mathbf{w})])$
  - Ermoliev'76, Wang et al'14 (SGCD)
  - Wang et al'17, Ghadimi et al'20 (NASA)
  - Zhang & Xiao'19, '21, Balasubramanian et al'21, Chen et al'21
  - Qi et al'21, Jiang et al.'22 (optimal rates)

No Coupling

- **Conditional Stochastic Optimization** $\mathbb{E}_\zeta f_\zeta(\mathbb{E}_{\xi|\zeta}[g_\xi(\mathbf{w};\zeta)])$
  - Hu et al'19

Huge Batch Size

Algorithms & Theories

Our Goals

1. **Simple** as SGD for ERM
2. **Constant** Batch Size Ensures Convergence
3. **Same-order** Complexity as SGD for ERM
4. **Parallel** Speed-up using Mini-Batch

Overview of Our Results

SOAP	(NeurIPS'21)	First Provable Algorithm
MOAP	(AISTATS'22)	Improving Rate of NC functions
SOX	(ICML'22)	Parallel Speed-up & Cvx functions
MSVR	(NeurIPS'22)	Improving Rate of NC & Cvx functions
SONX	(NeurIPS'23)	Non-smooth NC Functions

SOX: Steps

Estimator of g_i

$$u_i^{t+1} = \begin{cases} (1-\gamma_t)u_i^t + \gamma_t g(\mathbf{w}_t, \mathbf{x}_i, \mathcal{B}_i), & \mathbf{x}_i \in \mathcal{B}_1^t \\ u_i^t & \text{o.w.} \end{cases}$$

Gradient **E**stimator

$$G_t = \frac{1}{B_1} \sum_{\mathbf{x}_i \in \mathcal{B}_1^t} \nabla f(u_i^t) \nabla g(\mathbf{w}_t, \mathbf{x}_i, \mathcal{B}_i)$$

Model **U**pdate

$$\mathbf{v}_{t+1} = \beta_1 \mathbf{v}_t + (1-\beta_1) G_t$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{v}_{t+1}$$

Momentum or **A**dam

Wang & Yang. ICML'22

SOX: Analysis

$$\boxed{\frac{1}{n}\sum_{i=1}^{n}(u_i^t - g(\mathbf{w}_t; \mathbf{x}_i, \mathcal{S}_i))^2}$$

Estimator of g_i

$$u_i^{t+1} = \begin{cases} (1-\gamma_t)u_i^t + \gamma_t g(\mathbf{w}_t, \mathbf{x}_i, \mathcal{B}_i), & \mathbf{x}_i \in \mathcal{B}_1^t \\ u_i^t & \text{o.w.} \end{cases}$$

Wang & Yang. ICML'22

SOX: Analysis

$$\frac{1}{T}\sum_{t=1}^{T}\boxed{\frac{1}{n}\sum_{i=1}^{n}(u_i^t - g(\mathbf{w}_t;\mathbf{x}_i,\mathcal{S}_i))^2} \leq O\left(\frac{1}{\sqrt{T}}\right) \to 0$$

Estimator of g_i

$$u_i^{t+1} = \begin{cases} (1-\gamma_t)u_i^t + \gamma_t g(\mathbf{w}_t,\mathbf{x}_i,\mathcal{B}_i), & \mathbf{x}_i \in \mathcal{B}_1^t \\ u_i^t & \text{o.w.} \end{cases}$$

Wang & Yang. ICML'22

SOX: Analysis

$$\frac{1}{T}\sum_{t=1}^{T}\boxed{\frac{1}{n}\sum_{i=1}^{n}(u_i^t - g(\mathbf{w}_t;\mathbf{x}_i,\mathcal{S}_i))^2} \leq O\left(\frac{1}{\sqrt{T}}\right) \to 0$$

Stochastic Coordinate Descent (SCD)

Estimator of g_i

$$u_i^{t+1} = \begin{cases} (1-\gamma_t)u_i^t + \gamma_t g(\mathbf{w}_t, \mathbf{x}_i, \mathcal{B}_i), & \mathbf{x}_i \in \mathcal{B}_1^t \\ u_i^t & \text{o.w.} \end{cases}$$

Wang & Yang. ICML'22

SOX: Complexity

Assumption: smoothness & Lipschitz continuity

Iteration Complexity

$$\mathbb{E}\|\nabla F(\mathbf{w})\|^2 \leq \epsilon^2 \Rightarrow O\left(\frac{n}{B_1 B_2 \epsilon^4}\right)$$

Better Total Complexity than SGD and BSGD

$$n \leq \max_i |\mathcal{S}_i| \qquad n \leq \frac{1}{\epsilon^2}$$

Wang & Yang. ICML'22

SOX: Complexity

Assumption: smoothness & Lipschitz continuity

Iteration Complexity

SCD of u

$$\mathbb{E}\|\nabla F(\mathbf{w})\|^2 \leq \epsilon^2 \Rightarrow O\left(\frac{\boxed{n}}{B_1 B_2 \epsilon^4}\right)$$

Better Total Complexity than SGD and BSGD

$$n \leq \max_i |\mathcal{S}_i| \qquad n \leq \frac{1}{\epsilon^2}$$

Wang & Yang. ICML'22

SOX: Complexity

Assumption: smoothness & Lipschitz continuity

Iteration Complexity

$$\mathbb{E}\|\nabla F(\mathbf{w})\|^2 \leq \epsilon^2 \Rightarrow O\left(\frac{\boxed{n}}{B_1 B_2 \boxed{\epsilon^4}}\right)$$

SCD of u

Match SGD

Better Total Complexity than SGD and BSGD

$$n \leq \max_i |\mathcal{S}_i| \qquad n \leq \frac{1}{\epsilon^2}$$

Wang & Yang. ICML'22

SOX: Complexity

Assumption: smoothness & Lipschitz continuity

Iteration Complexity

$$\mathbb{E}\|\nabla F(\mathbf{w})\|^2 \leq \epsilon^2 \Rightarrow O\left(\frac{\boxed{n}}{\boxed{B_1 B_2}\boxed{\epsilon^4}}\right)$$

SCD of u

Match SGD

Parallel Speed-up

Better Total Complexity than SGD and BSGD

$$n \leq \max_i |\mathcal{S}_i| \qquad n \leq \frac{1}{\epsilon^2}$$

Wang & Yang. ICML'22

SOX: Complexity

Objective is **C**onvex

Iteration Complexity

$$\mathbb{E}[F(\mathbf{w}) - F(\mathbf{w}_*)] \leq \epsilon \Rightarrow O\left(\frac{n}{B_1 B_2 \epsilon^3}\right)$$

Not Match SGD

Using **D**ouble-loop **T**rick

Wang & Yang. ICML'22

MSVR: Steps

Error Correction

Estimator of g_i

$$u_i^{t+1} = (1-\gamma_t)u_i^t + \gamma_t g_i(\mathbf{w}_t, \mathcal{B}_i) + \beta_t(g_i(\mathbf{w}_t, \mathcal{B}_i) - g_i(\mathbf{w}_{t-1}, \mathcal{B}_i)), \ \mathbf{x}_i \in \mathcal{B}_1^t$$

$$\beta_t = \frac{n - B_1}{B_1(1-\gamma_t)} + 1 - \gamma_t$$

Gradient
Estimator

STORM (Cutkosky and Orabona'19)

Jiang et al. NeurIPS'22

MSVR: Complexity

Assumption: average smoothness & Lipschitz continuity

Non-Convex

$$O\left(\frac{n}{B_1\sqrt{B_2}\epsilon^3}\right)$$

Convex

$$O\left(\frac{n}{B_1\sqrt{B_2}\epsilon^2}\right)$$

Match SGD

Jiang et al. NeurIPS'22

MSVR: Complexity

Assumption: average smoothness & Lipschitz continuity

Non-Convex

$$O\left(\frac{n}{B_1\sqrt{B_2}\epsilon^3}\right)$$

Convex

$$O\left(\frac{n}{B_1\sqrt{B_2}\epsilon^2}\right)$$

Match SGD

Non-trivial (ongoing)

$$O\left(\frac{n}{B_1 B_2\epsilon^2}\right)$$

Jiang et al. NeurIPS'22

SONX: Non-smooth Non-convex

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i))$$

Non-**S**mooth **W**eakly-**C**onvex

Hu et al. NeurIPS'23

SONX: Non-smooth Non-convex

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i))$$

Non-**S**mooth **W**eakly-**C**onvex

$$\|\nabla f(u_i) - \nabla f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i))\|^2 \cancel{\leq L_f} \|u_i - (g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i)\|^2$$

Hu et al. NeurIPS'23

SONX: Non-smooth Non-convex

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i))$$

Non-**S**mooth **W**eakly-**C**onvex

$$\|\nabla f(u_i) - \nabla f(g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i))\|^2 \cancel{\leq} L_f \|u_i - (g(\mathbf{w}, \mathbf{x}_i, \mathcal{S}_i)\|^2$$

f is **non-decreasing**

Hu et al. NeurIPS'23

SONX: Steps

Estimator of g_i	**Same** as MSVR
Gradient **E**stimator	**Same** as SOX
Model **U**pdate	**Same** as SGD

Hu et al. NeurIPS'23

SONX: Analysis and Complexity

$$\langle \frac{1}{n} \sum_i \partial f(u_i^t) \partial g_i(\mathbf{w}_t), \widehat{\mathbf{w}}_t - \mathbf{w}_t \rangle$$

$$\widehat{\mathbf{w}}_t = \arg\min_{\mathbf{w}} F(\mathbf{w}) + \frac{\rho}{2} \|\mathbf{w} - \mathbf{w}_t\|_2^2$$

$$\mathbf{E}[\min_t \|\widehat{\mathbf{w}}_t - \mathbf{w}_t\|^2] \leq \epsilon^2$$

Hu et al. NeurIPS'23

SONX: Analysis and Complexity

$$\langle \frac{1}{n}\sum_i \partial f(u_i^t)\partial g_i(\mathbf{w}_t), \widehat{\mathbf{w}}_t - \mathbf{w}_t \rangle$$

$$\widehat{\mathbf{w}}_t = \arg\min_{\mathbf{w}} F(\mathbf{w}) + \frac{\rho}{2}\|\mathbf{w} - \mathbf{w}_t\|_2^2$$

Weakly-
Convex

$$\leq \frac{1}{n}\sum_i \boxed{[f(g_i(\mathbf{w}_t)) - f(u_i^t) - \partial f(u_i^t)(g_i(\mathbf{w}_t) - u_i^t)]}$$

Non-
Decreasing

$$-O(\|\widehat{\mathbf{w}}_t - \mathbf{w}_t\|^2) + O(\frac{1}{n}\sum_i \|u_i^t - g_i(\mathbf{w}_t)\|^2)$$

First-order Error

$$\mathbf{E}[\min_t \|\widehat{\mathbf{w}}_t - \mathbf{w}_t\|^2] \leq \epsilon^2$$

Hu et al. NeurIPS'23

SONX: Analysis and Complexity

$$\langle \frac{1}{n}\sum_i \partial f(u_i^t)\partial g_i(\mathbf{w}_t), \widehat{\mathbf{w}}_t - \mathbf{w}_t\rangle$$

$$\widehat{\mathbf{w}}_t = \arg\min_{\mathbf{w}} F(\mathbf{w}) + \frac{\rho}{2}\|\mathbf{w} - \mathbf{w}_t\|_2^2$$

Weakly-
Convex

$$\leq \frac{1}{n}\sum_i \boxed{[f(g_i(\mathbf{w}_t)) - f(u_i^t) - \partial f(u_i^t)(g_i(\mathbf{w}_t) - u_i^t)]}$$

Non-
Decreasing

$$-O(\|\widehat{\mathbf{w}}_t - \mathbf{w}_t\|^2) + O(\frac{1}{n}\sum_i \|u_i^t - g_i(\mathbf{w}_t)\|^2)$$

First-order Error

$$\mathbf{E}[\min_t \|\widehat{\mathbf{w}}_t - \mathbf{w}_t\|^2] \leq \epsilon^2 \quad \Rightarrow \quad O\left(\frac{n}{B_1\sqrt{B_2}\epsilon^6}\right)$$

Hu et al. NeurIPS'23

Applications in ML/AI

Applications

- **AUPRC/AP Maximization**
- **Contrastive Self-supervised Learning**
- **Other Applications**

AP Max. for Molecular Property Prediction

3.5% Positive **2 ~3% Improvement**

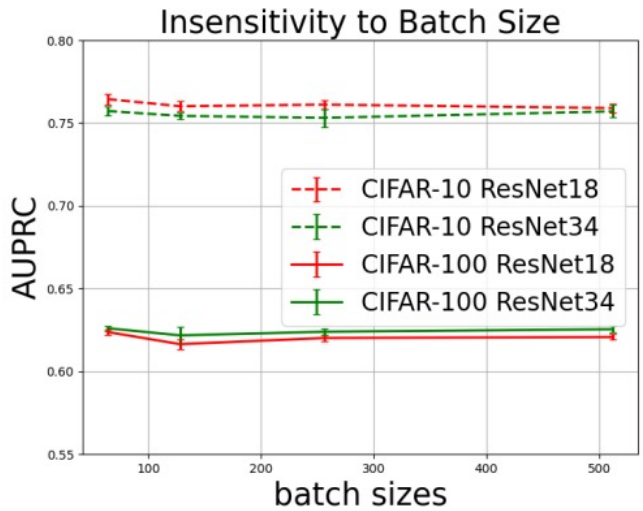
Dataset	Method	GINE	MPNN	ML-MPNN
HIV	CE	0.2774 (± 0.0101)	0.3197 (± 0.0050)	0.2988 (± 0.0076)
	CB-CE	0.3082 (± 0.0101)	0.3056 (± 0.0018)	0.3291 (± 0.0189)
	Focal	0.3179 (± 0.0068)	0.3136 (± 0.0197)	0.3279 (± 0.0173)
	LDAM	0.2904 (± 0.0008)	0.2994 (± 0.0128)	0.3044 (± 0.0116)
	AUC-M	0.2998 (± 0.0010)	0.2786 (± 0.0456)	0.3305 (± 0.0165)
	SmoothAP	0.2686 (± 0.0007)	0.3276 (± 0.0063)	0.3235 (± 0.0092)
	FastAP	0.0169 (± 0.0031)	0.0826 (± 0.0112)	0.0202 (± 0.0002)
	MinMax	0.2874 (± 0.0073)	0.3119 (± 0.0075)	0.3098 (± 0.0167)
	SOAP	**0.3385 (± 0.0024)**	**0.3401 (± 0.0045)**	**0.3547 (± 0.0077)**
MUV	CE	0.0017 (±0.0001)	0.0021 (±0.0002)	0.0025 (±0.0004)
	CB-CE	0.0055 (±0.0011)	0.0483 (±0.0083)	0.0121 (±0.0016)
	Focal	0.0041 (±0.0007)	0.0281 (±0.0141)	0.0122 (±0.0001)
	LDAM	0.0044 (±0.0022)	0.0118 (±0.0098)	0.0059 (±0.0021)
	AUC-M	0.0026 (±0.0001)	0.0040 (±0.0012)	0.0028 (±0.0012)
	SmoothAP	0.0073 (±0.0012)	0.0068 (±0.0038)	0.0029 (±0.0005)
	FastAP	0.0016 (±0.0000)	0.0023 (±0.0021)	0.0022 (±0.0012)
	MinMax	0.0028 (±0.0008)	0.0027 (±0.0005)	0.0043 (±0.0015)
	SOAP	**0.0254 (±0.0261)**	**0.3352** (±0.0008)	**0.0236** (±0.0038)

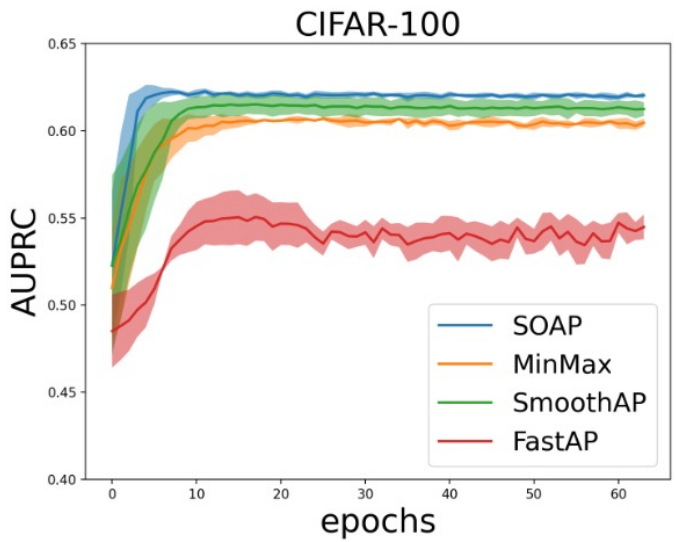
0.2% Positive **33% Improvement**

Data	MIT AICURES	
Networks	GINE	MPNN
CE	0.5037 (± 0.0718)	0.6282 (± 0.0634)
CB-CE	0.5655 (± 0.0453)	0.6308 (± 0.0263)
Focal	0.5143 (± 0.1062)	0.5875 (± 0.0774)
LDAM	0.5236 (± 0.0551)	0.6489 (± 0.0556)
AUC-M	0.5149 (± 0.0748)	0.5542 (± 0.0474)
SmoothAP	0.2899 (± 0.0220)	0.4081 (± 0.0352)
FastAP	0.4777 (± 0.0896)	0.4518 (± 0.1495)
MinMax	0.5292 (± 0.0330)	0.5774 (± 0.0468)
SOAP	**0.6639 (± 0.0515)**	**0.6547 (± 0.0616)**

2.2% Positive **3% Improvement**

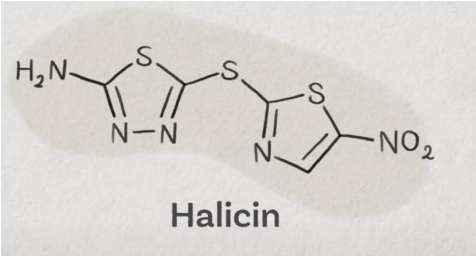
Qi et al. NeurIPS'22


Insensitivity to Batch Size
AUPRC
batch sizes
0.80
0.75
0.70
0.65
0.60
0.55
100
200
300
400
500
CIFAR-10 ResNet18
CIFAR-10 ResNet34
CIFAR-100 ResNet18
CIFAR-100 ResNet34


CIFAR-100
AUPRC
epochs
0.65
0.60
0.55
0.50
0.45
0.40
0
10
20
30
40
50
60
SOAP
MinMax
SmoothAP
FastAP

1st Place at MIT AICures Challenge

Fighting Secondary Effects of Covid



Stokes et al. 2020. Cell.

Collaborating with Prof. Shuiwang Ji's group

Rank	Model	Author	Submissions	10-fold CV ROC-AUC	10-fold CV PRC-AUC	Test ROC-AUC	Test PRC-AUC
1		DIVE@TAMU	11			0.957	0.729
2	MolecularG	AIDrug@PA	9			0.7	0.725
3		AGL Team	20			0.675	0.702
4		phucdoitoan@Fujitsu	14	0.898 +/- 0.113	0.508 +/- 0.253	0.867	0.694
5	GB	BI	6			0.698	0.67
6	Chemprop ++	AICures@MIT	4			0.877	0.662
7		Mingjun Liu	3			0.72	0.657
8	Pre-trained OGB-GIN (ensemble)	Weihua Hu@Stanford	2	0.905 +/- 0.133	0.494 +/- 0.333	0.837	0.651
9	RF + fingerprint	Cyrus Maher@Vir Bio	1	0.896 +/- 0.074	0.481 +/- 0.338	0.799	0.649
10	Graph Self-supervised Learning	SJTU_NRC_Mila	3	0.825 +/- 0.210	0.530 +/- 0.342	0.800	0.622

1st Place at MIT AICures Challenge

Rank	Model	Author	Submissions	10-fold CV ROC-AUC	10-fold CV PRC-AUC	Test ROC-AUC	Test PRC-AUC
1		DIVE@TAMU	11			0.957	0.729

AUROC

AUPRC

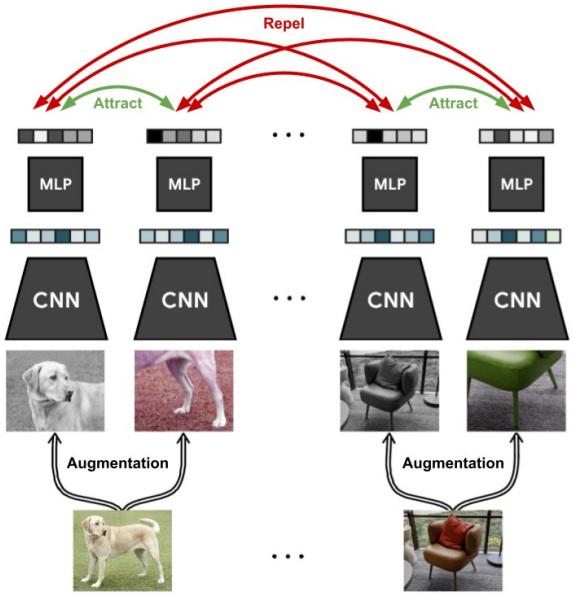
w/o LibAUC

1	MoleculeKit	DIVE@TAMU	7	0.928
3	MoleculeKit	DIVE@TAMU	7	0.677

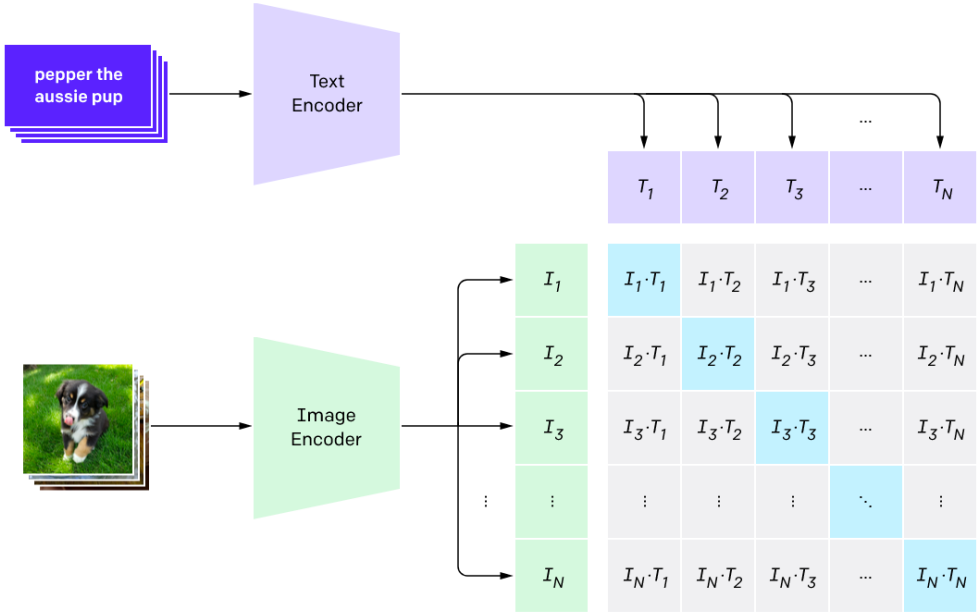
5% Improvement in **AUPRC**, **3%** Improvement in **AUROC**

Wang et al. (Bioinformatics'22)

Contrastive Self-supervised Learning



Google's SimCLR (Chen et al. '20)



OpenAI's CLIP (Radford et al. '21)

Global Contrastive Loss (GCL)

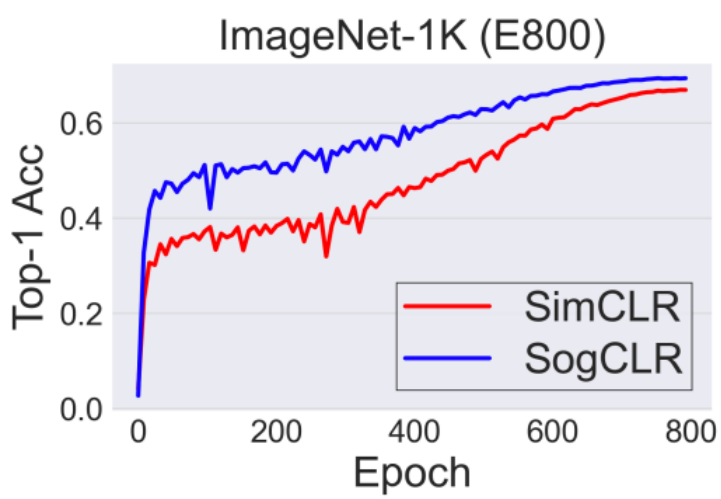
Temperature

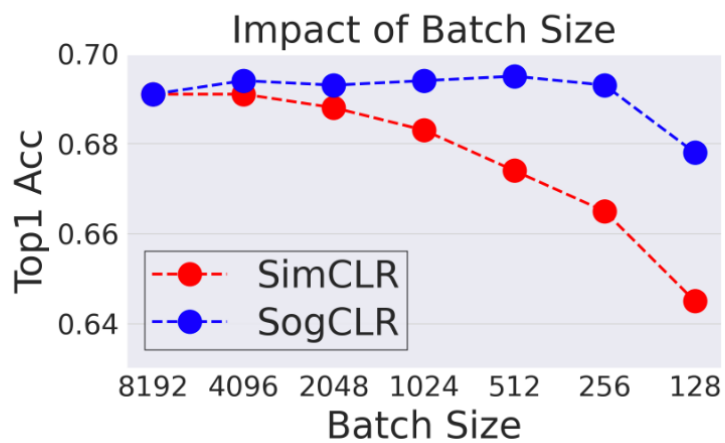
$$\ell_{\mathrm{GCL}}(\mathbf{x}_i) = \tau \log\left(\sum_{\mathbf{x}_j \sim \mathcal{S}_i^-} \exp(\underbrace{(h_{\mathbf{w}}(\mathbf{x}_i)^\top h_{\mathbf{w}}(\mathbf{x}_j) - h_{\mathbf{w}}(\mathbf{x}_i)^\top h_{\mathbf{w}}(\mathbf{x}_i^+))}_{\ell(\mathbf{w}, \mathbf{x}_i, \mathbf{x}_j)}/\tau)\right)$$

All Negative Data

SogCLR = SOX + $u(\mathbf{x}_i, \mathbf{x}_i^+) = u(\mathbf{x}_i)$

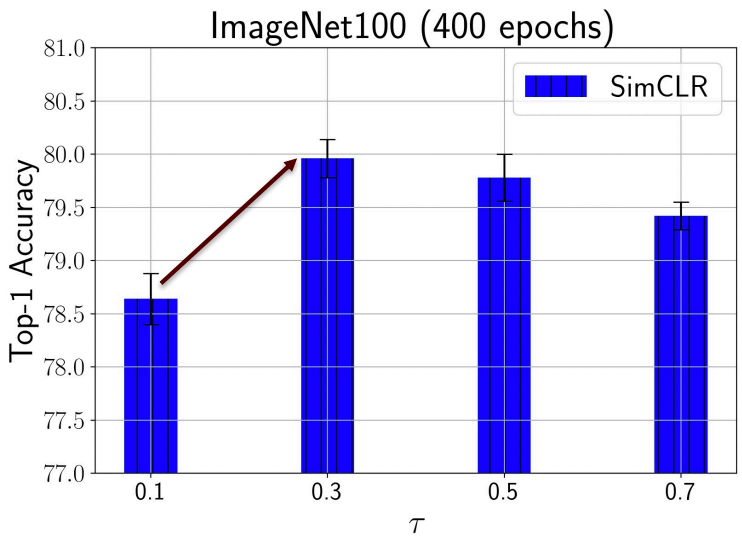
SogCLR vs SimCLR



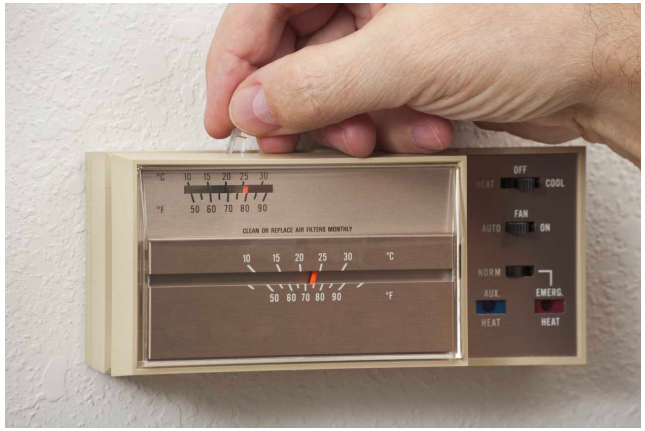


Yuan et al. ICML'22

Optimizing Temperature



Sensitive to Temperature



DRO for Optimizing Temperatures

Weights

$$\ell_{\mathrm{RGCL}}(\mathbf{x}_i) := \max_{\mathbf{p}\in\Delta_m} \sum\nolimits_{\mathbf{x}_j\in\mathcal{S}_i^-} \mathbf{p}_j \ell(\mathbf{w}, \mathbf{x}_i, \mathbf{x}_j) - \tau_0 \mathrm{KL}(\mathbf{p}, \mathbf{1}/m)$$
$$\text{s.t.} \quad \mathrm{KL}(\mathbf{p}, \mathbf{1}/m) \leq \rho,$$

Robust Global Contrastive Loss

Qiu et al. ICML'23

Optimization Algorithm: iSogCLR

$$\min_{\mathbf{w},\tau\geq\tau_0} F(\mathbf{w},\tau) := \frac{1}{n}\sum_{i=1}^{n}\left\{\tau_i \log \mathbb{E}_{\mathbf{x}_j\sim\mathcal{S}_i^-} \exp\left(\frac{\ell(\mathbf{w};\mathbf{x}_i,\mathbf{x}_j)}{\tau_i}\right) + \tau_i\rho\right\}$$

Let us Optimize It

Individualized

iSogCLR = SogCLR + SCD update of temperatures

Qiu et al. ICML'23

Automatic Temperature Individualization (ATI)

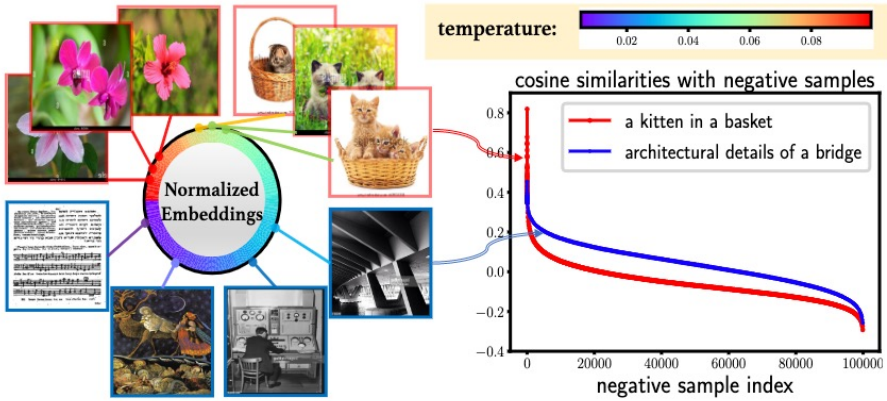


Figure 1. Right: Samples with *frequent* semantics (e.g., a kitten in a basket) have many more similar samples than that with *rare* semantics (e.g., architectural details of a bridge). Left: An illustration of temperature individualization by our algorithm named iSogCLR, making "hot" images with frequent semantics use a higher temperature to keep semantic structures, and making "cold" images with rare semantics use a lower temperature for inducing more separable features. The circular heatmap is plotted using learned temperatures of 100 random images from the CC3M dataset.

an image with a large τ

red kittens in a basket isolated on a white background

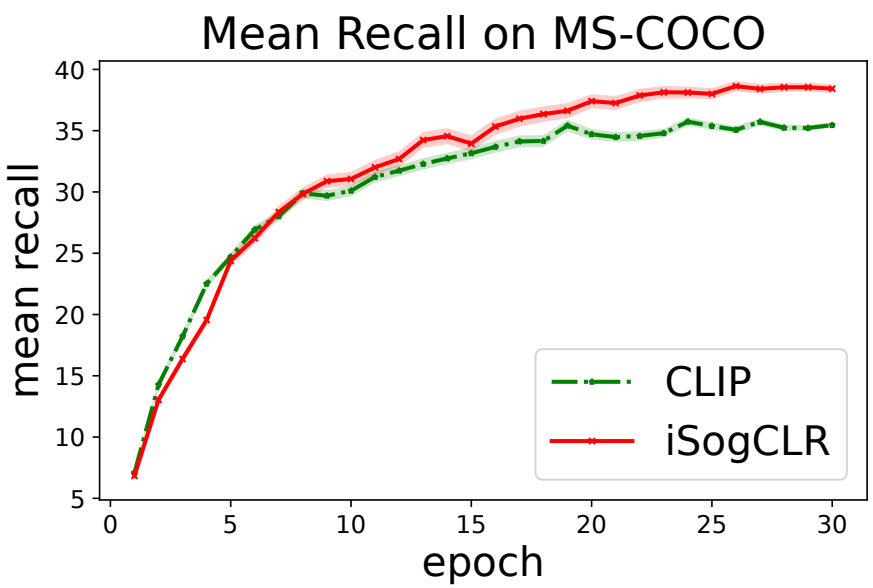
"Hot" Images
High Temperature

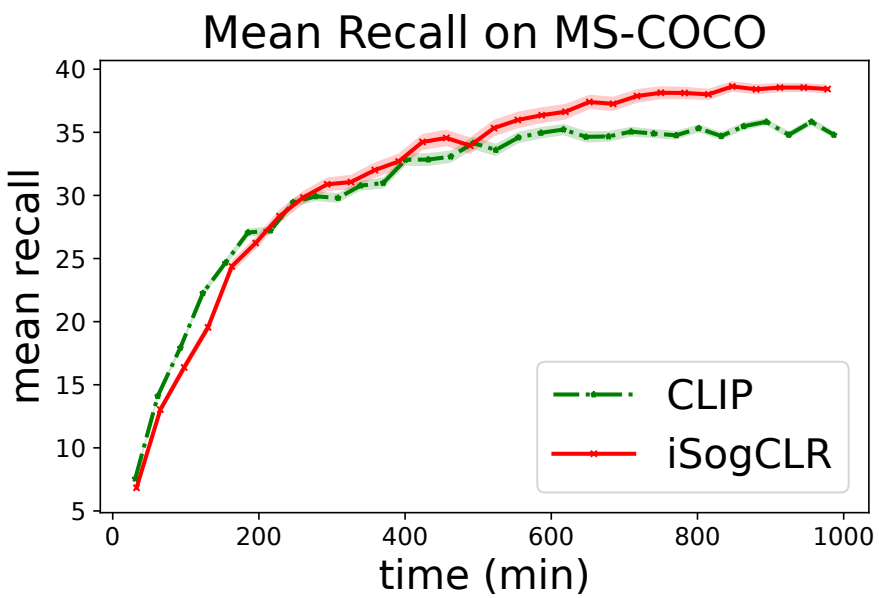
an image with a small τ

architectural details of a bridge

"Cold" Images
Low Temperature

Experiments: vs OpenAI CLIP

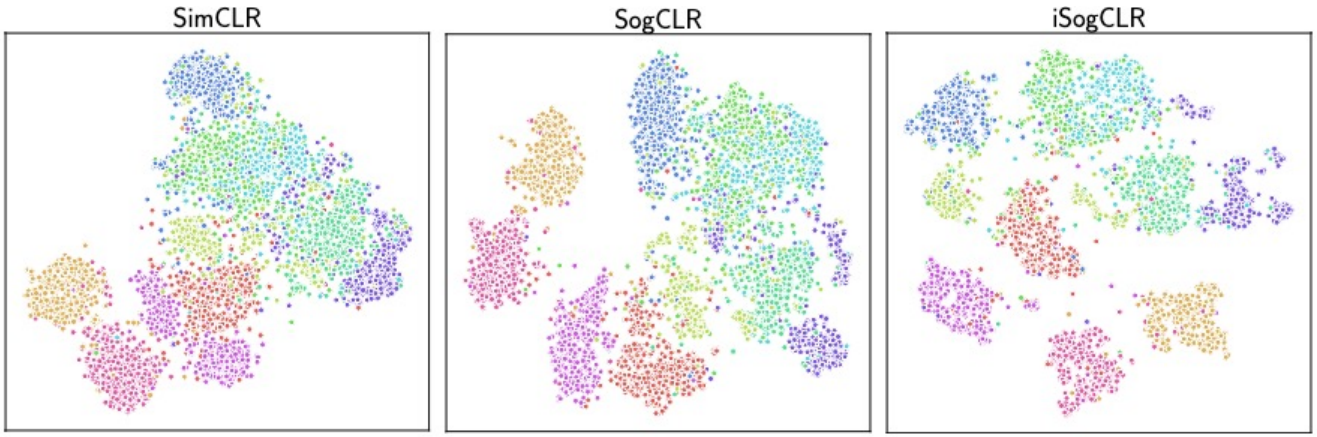




Faster/Better Convergence

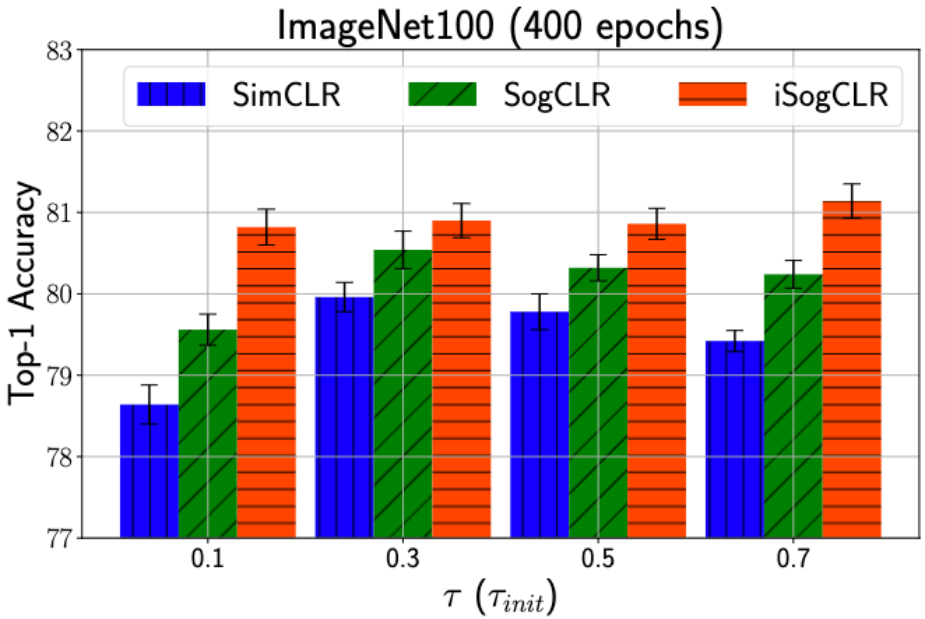
Experiments: vs Google SimCLR

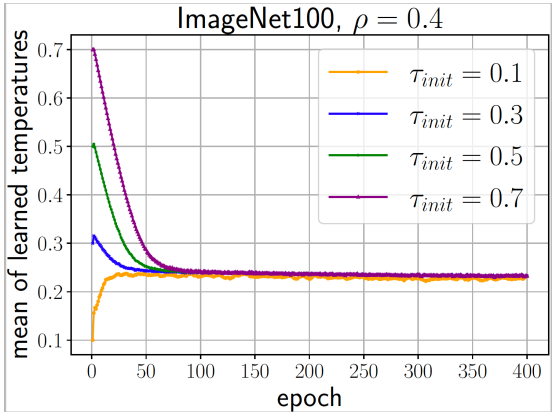
CIFAR-10

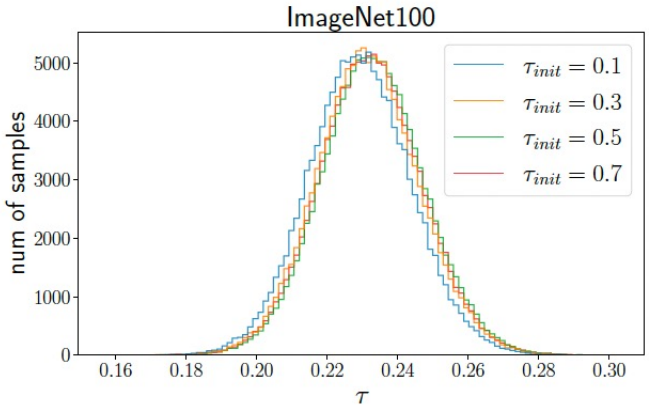


Better Representations

Experiments: vs Google SimCLR







Not Sensitive to Initial Temperature

Experiments: vs Others

Image

Table 1. Linear evaluation results with 400 pretraining epochs on six unimodal image datasets. We report the average top-1 accuracies (%) and standard deviation over 3 runs with different random seeds. Full results are provided in Table 3 and 4 in Appendix C.3.

Method	CIFAR10	CIFAR100	ImageNet100	CIFAR10-LT	CIFAR100-LT	iNaturalist
SimCLR	88.74±0.18	62.34±0.09	79.96±0.20	77.09±0.13	49.33±0.12	91.52±0.17
Barlow Twins	87.39±0.14	62.28±0.13	79.16±0.13	75.94±0.08	48.39±0.14	91.89±0.21
FlatCLR	88.61±0.10	63.27±0.07	80.24±0.16	77.96±0.12	52.61±0.06	92.54±0.09
Spectral CL	88.77±0.09	63.06±0.18	80.48±0.08	76.38±0.21	51.86±0.16	92.13±0.16
SogCLR	88.93±0.11	63.14±0.12	80.54±0.14	77.70±0.07	52.35±0.08	92.60±0.08
VICReg	88.96±0.16	62.44±0.13	80.16±0.22	75.05±0.09	48.43±0.13	93.03±0.14
SimCo	88.86±0.12	62.67±0.06	79.73±0.17	77.71±0.13	51.06±0.09	92.10±0.12
iSogCLR	**89.24**±0.15	**63.82**±0.14	**81.14**±0.19	**78.37**±0.16	**53.06**±0.12	**93.08**±0.19

Image-text

Table 2. Results on two bimodal downstream tasks. For image-text retrieval on Flickr30K and MSCOCO, we compute IR@1 and TR@1 for the Recall@1 on image-retrieval (IR) and text-retrieval (TR). For classification tasks, we compute top-1 accuracy (%). We report the average of scores and standard deviation over 3 runs with different random seeds. Full results are in Table 5, 6, and 7 in Appendix C.3.

Method	Flickr30K Retrieval		MSCOCO Retrieval		Zero-shot Classification top-1 Acc		
	IR@1	TR@1	IR@1	TR@1	CIFAR10	CIFAR100	ImageNet1K
CLIP	40.98±0.22	50.90±0.17	21.32±0.12	26.98±0.21	60.63±0.19	30.70±0.11	36.27±0.17
CyCLIP	42.46±0.13	51.70±0.23	21.58±0.19	26.18±0.24	57.19±0.20	33.11±0.14	36.75±0.21
SogCLR	41.18±0.18	50.90±0.20	21.59±0.13	28.60±0.22	60.16±0.24	**33.33**±0.12	36.78±0.19
iSogCLR	**42.82**±0.12	**54.00**±0.26	**22.11**±0.18	**31.62**±0.13	**61.86**±0.15	33.30±0.18	**38.51**±0.23

Better Predictive Performance

Deep X-risk Optimization

Library Facts

Features

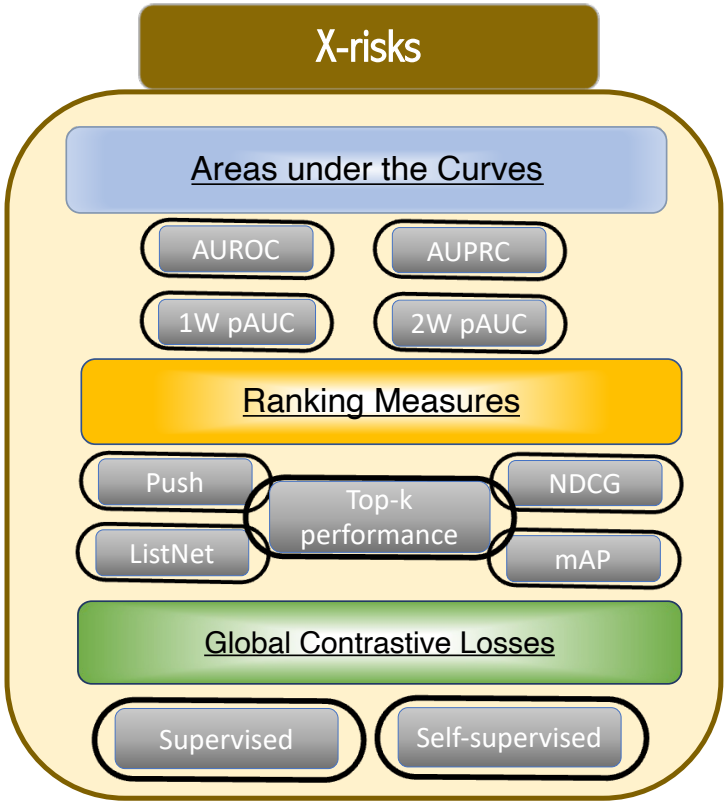
- **A**ny batch size
- **B**ig data
- **C**onvergence guaranteed
- **D**eployment is easy

When Using this Library

- Learning with Imbalanced Data
- Learning to Rank
- Contrastive Learning

Formula

$$\frac{1}{n}\sum_{i=1}^{n} f_i(g_i(\mathbf{w}; \mathbf{z}_i, \mathcal{S}_i))$$



Medicine

Search Engines

Recommender Systems

GenAI

LibAUC

A Deep Learning Library for X-Risk Optimization

An open-source library that translates theories to real-world applications

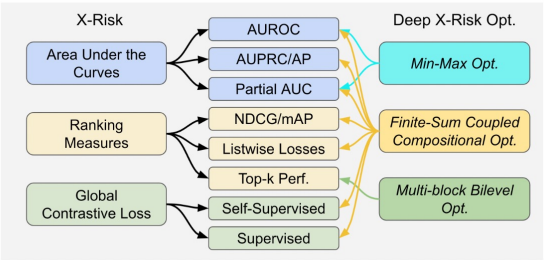
Latest News | Install

[2023-06] LibAUC 1.3.0 is released!

Why LibAUC?

LibAUC is a novel deep learning library to offer an easier way to directly optimize commonly used performance measures and losses with user-friendly APIs. LibAUC has broad applications in AI for tackling both classic and emerging challenges, such as *Classification of Imbalanced Data (CID)*, *Learning to Rank (LTR)*, and *Contrastive Learning of Representation (CLR)*.

LibAUC provides a unified framework to abstract the optimization of a family of risk functions called **X-Risk**, including surrogate losses for *AUROC*, *AUPRC/AP*, and *partial AUROC* that are suitable for CID, surrogate losses for *NDCG*, *top-K NDCG*, and *listwise losses* that are used in LTR, and *global contrastive losses* for CLR. For more details, please check our LibAUC paper.



3+

Challenges winning solution (e.g., Stanford CheXpert, MIT AICures, OGB Graph Property Prediction).

4+

Collaborations and Deployments at multiple industrial units, e.g., Google, Uber, Tencent, etc.

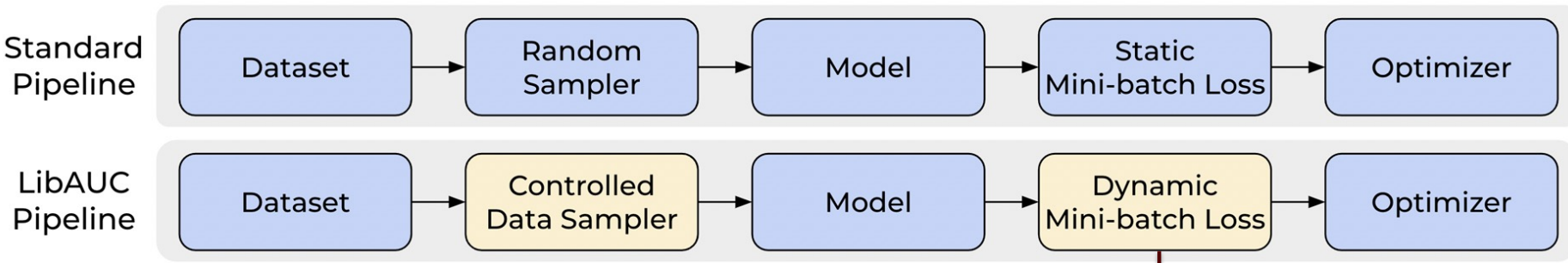
17+

Scientific publications on top-tier AI Conferences (such as ICML, NeurIPS, ICLR).

35000+

Downloaded by more than 35K+ times by researchers around the world.

Simple Training Pipeline



$$\frac{1}{B_1}\sum_{\mathbf{x}_i\in\mathcal{B}_1^t}\nabla f(u_i^t)g(\mathbf{w}_t,\mathbf{x}_i,\mathcal{B}_i)$$

detach from Computation Graph

Summary

- **<u>FCCO: General Framework</u>**
  - Extensive theories (C/NC, S/NS)
- **<u>X-risks: Broad Applications</u>**
  - AUC, AP, NDCG, GCL, etc.
- **<u>LibAUC Library: Practical Algorithms</u>**
  - Many people are using
  - Please give us feedback

References

Stochastic optimization of areas under precision-recall curves with provable convergence
Q Qi, Y Luo, Z Xu, S Ji, T Yang — NeurIPS'21

Finite-sum coupled compositional stochastic optimization: Theory and applications — ICML'22
B Wang, T Yang

Multi-block-single-probe variance reduced estimator for coupled compositional optimization — NeurIPS'22
W Jiang, G Li, Y Wang, L Zhang, T Yang

Non-Smooth Weakly-Convex Finite-sum Coupled Compositional Optimization — NeurIPS'23
Q Hu, D Zhu, T Yang

Libauc: A deep learning library for x-risk optimization — KDD'23
Z Yuan, D Zhu, ZH Qiu, G Li, X Wang, T Yang

Algorithmic foundation of deep x-risk optimization — arXiv'22
T Yang