X-risk Optimization: A New Paradigm for Deep Learning

Tianbao Yang
Texas A&M University
Outline

• Overview & Background

• Three Use Cases
My Research Focus

Advancing Optimization to Make ML/AI Faster and Better

- **Domains** (e.g., medicine)
- **Applications** (e.g., drug discovery)
- **Representations** (data, models)
- **Formulations** (objectives)
- **Optimization** (algorithms)

AI is like an Onion

- Training Faster
- Testing better
Optimization for Machine Learning

\[ \min_w F(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w, z_i) \]

Empirical Risk Minimization (ERM)
SGD: Stochastic Gradient Descent

\[ \mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla \ell(\mathbf{w}_t, \mathbf{z}_t) \]

Conventional: Polynomially Decreasing

Modern: Stagewise

Modern: Adaptive
Momentum and Adaptive Methods

**Imagenet classification with deep convolutional neural networks**
A Krizhevsky, I Sutskever, GE Hinton
Advances in neural information processing systems 25, 1097-1105

**Stochastic Heavy-ball Method (SHB)**

**On the importance of initialization and momentum in deep learning**
I Sutskever, J Martens, G Dahl, G Hinton
International conference on machine learning, 1139-1147

**Stochastic Nesterov’s Accelerated Gradient (SNAG)**

**Adam: A method for stochastic optimization**
D Kingma, J Ba
International Conference on Learning Representations

\[ \mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla \ell(\mathbf{w}_t, \mathbf{z}_t) + \delta_t \]

Momentum term

Adaptive or Stagewise
A Standard Learning Paradigm

1. Sample Mini-batch Samples
2. Define Mini-batch (MB) Losses
3. Back-propagation on MB Losses
4. Update Model Parameters
As provided in Figure 4a, R@1 monotonically improves with larger batch size on all three datasets. This observation resonates with the fact that large batches reduce the variance of the stochastic gradients, which has been shown to be beneficial [32]. On the other hand, from the learn-
Some Undesirable Consequences

Patel et al. Recall@k Surrogate Loss with Large Batches and Similarity Mixup. In CVPR, 2022.

```

Batch size. The effect of the varying batch size is shown in Figure 4 (right). It demonstrates that large batch size leads to better results. A significant performance boost is
```

5.2. Contrastive learning benefits (more) from larger batch sizes and longer training

Figure 9 shows the impact of batch size when models are trained for different numbers of epochs. We find that, when the number of training epochs is small (e.g. 100 epochs), larger batch sizes have a significant advantage over the smaller ones. With more training steps/epochs, the gaps
Conventionally Small Batch is Fine

\[
\min_w F(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w, z_i)
\]

"The stochastic gradient descent (SGD) method and its variants are algorithms of choice for many Deep Learning tasks. These methods operate in a small-batch regime wherein a fraction of the training data, say 32–512 data points, is sampled to compute an approximation to the gradient. It has been observed in practice that when using a larger batch there is a degradation in the quality of the model, as"

A Standard Learning Paradigm

Sample Mini-batch Samples

Define Mini-batch (MB) Losses

Back-propagation on MB Losses

Update Model Parameters

Q: What is Wrong about this Learning Paradigm?

A: ERM is **NOT** enough
Beyond ERM: Deep X-risk Optimization
A family of **Compositional** measures in which the loss function of each data point is defined in a way that **Contrasts** the data point with a **Large number of items**.

\[
F(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(g(w, z_i, S_i))
\]
Challenges of Optimizing X-risk

\[ F(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(g(w, z_i, S_i)) \]

**Full Gradient**  
for each data

\[ \nabla f_i(g(w, z_i, S_i)) \nabla g(w, z_i, S_i) \]

**Mini-batch Gradient**

\[ \nabla f_i(g(w, z_i, B_i)) \nabla g(w, z_i, B_i) \]

Biased  
Mini-batch
Outline

• Three Use Cases
  • AUPRC/AP Maximization
  • Top-K NDCG Maximization
  • Self-supervised Learning
Deep AUPRC/AP Maximization
# Evaluation Metric: AUPRC

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

## Test ROC-AUC

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

---

**MIT AICures Challenge**

**Fighting Secondary Effects of Covid**

Why AUROC Max. is NOT Enough?

**Challenge:** Highly Imbalanced Data
Non-Parametric Estimator: Average Precision

\[ AP(h) = \frac{1}{n_+} \sum_{x_i \in S_+} \text{Precision}(h(x_i)) \]

\[ \text{Precision}(h(x_i)) = \frac{\sum_{x_j \in S_+} \mathbb{I}(h(x_j) \geq h(x_i))}{\sum_{x_j \in S} \mathbb{I}(h(x_j) \geq h(x_i))} \]

Positive Examples

All Examples
Deep AUPRC Maximization

Limitations of Literature on AUPRC Maximization
(1) Not applicable to deep learning (e.g., SVM-AP, Yue et al.)
(2) No Convergence, require large batch (e.g., FastAP, Cakir et al.)

Our Contributions:
(1) New Formulation based on Compositional Opt.
(2) First Algorithms with Convergence Theory
(3) Practical Algorithms and Improved Theory

(NeurIPS’21, AISTATS’22, ICML’22)
Our Formulation

(NeurIPS 2021)

\[
\begin{align*}
\min_{w} F(w) &= \frac{1}{n_+} \sum_{x_i \in S_+} \left[ f(g_i(w)) \right] \\
\sum_{x_j \in S_+} \ell(h_w(x_j) - h_w(x_i)) &= [g_i(w)]_1 \\
\sum_{x_j \in S} \ell(h_w(x_j) - h_w(x_i)) &= [g_i(w)]_2
\end{align*}
\]

Limitations of Existing Methods

• Not Convergent (e.g., SGD/Adam)
• Not-scalable (e.g., NASA, Ghadimi et al.)
• Require Large batch size (e.g., BSGD, Hu et al.)

Finite-sum Coupled Compositional Optimization
Key Idea of SOAP

\[ \nabla f(g_i(w_t)) \]

Full Gradient at \( t^{th} \) iteration

Naïve Mini-batch \( \nabla f(\hat{g}_i(w_t)) \)

Unbiased

Vs.

Variance-reduced \( \nabla f(u^t_i) \)

Biased but variance-reduced

\[ u^t_i = (1 - \beta)u^{t-1}_i + \beta \hat{g}_i(w_t) \]

Sampled Positive

\( x_i \in B_+ \)
Theories

Goal

\[ \| \nabla F(w) \| \leq \epsilon \]

---

NeurIPS’21

First Algorithm with Convergence Guarantee

SGD-style Update

\[ O \left( \frac{1}{\epsilon^5} \right) \]

---

ICML’22, AISTATS’22

Improved Convergence

Momentum or Adam-style Update

\[ O \left( \frac{1}{\epsilon^4} \right) \]
**MIT AICures Challenge**

### 1st Place

**Fighting Secondary Effects of Covid**


Collaborating with Prof. Shuiwang Ji’s group at TAMU

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

**Evaluation Metric:** AUPRC
## Comparison with w/o DAM

<table>
<thead>
<tr>
<th>Rank</th>
<th>Model</th>
<th>Author</th>
<th>Submissions</th>
<th>10-fold CV ROC-AUC</th>
<th>10-fold CV PRC-AUC</th>
<th>Test ROC-AUC</th>
<th>Test PRC-AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DIVE@TAMU</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td>0.957</td>
<td>0.729</td>
</tr>
</tbody>
</table>

**w/o DAM**

<table>
<thead>
<tr>
<th></th>
<th>Model</th>
<th>Author</th>
<th>Submissions</th>
<th>10-fold CV ROC-AUC</th>
<th>10-fold CV PRC-AUC</th>
<th>Test ROC-AUC</th>
<th>Test PRC-AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>MoleculeKit</td>
<td>DIVE@TAMU</td>
<td>7</td>
<td></td>
<td></td>
<td>0.928</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>MoleculeKit</td>
<td>DIVE@TAMU</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td>0.677</td>
</tr>
</tbody>
</table>

5% Improvement in AUPRC, 3% Improvement in AUROC
Deep top-K NDCG Maximization
Most Relevant Items on the Top

Relevance
3 3 2 1 0

Position
1 2 3 4 5

Ideal Order of Items

Search Engines

Recommender Systems

Social Media
NDCG

$$NDCG_q = \frac{1}{Z_q} \sum_{i=1}^{n} \frac{2y_i - 1}{\log_2(1 + r(i))}$$

Relevance Score

Ideal DCG

Ranking position

Challenge I

$$r(i) = \sum_{x_j \in S_q} \mathbb{I}(h_w(x_j; q) \geq h_w(x_i; q))$$

Millions of Movies on Netflix
NDCG Surrogate is X-risk

\[ \text{NDCG}_q = \frac{1}{Z_q} \sum_{i=1}^{n} \frac{2y_i - 1}{\log_2(1 + r(i))} \]

\[ f(g(w; x_i, S_q)) \]

\[ g(w; x_i, S_q) = \sum_{x_j \in S_q} \ell(h_w(x_j; q) - h_w(x_i; q)) \]
Top-K NDCG

\[
\frac{1}{Z^K_q} \sum_{i=1}^{n} \mathbb{I}(i\text{-th item in top-K positions}) \left( \frac{2y_i - 1}{\log_2(1 + r(i))} \right)
\]

**Challenges**

- Finding top-K items require \(O(n\log n)\)
- Top-K selector is non-differentiable
Deep top-K NDCG Maximization

**Limitations** of Literature on Top-K NDCG Maximization

(1) Small Data or No Convergence (e.g., ApproxNDCG, Qin et al.)
(2) Not Applicable to Deep Learning (e.g., SVM-NDCG, Chakrabarti et al.)

**Our Contributions:** (ICML’22)

(1) **New** Formulation based on Bilevel Optimization
(2) **First** Algorithms with Convergence Theory
(3) **Practical** Algorithms
Transforming Top-$K$ Selector

(ICML 2022)

Prediction score

The $(K+1)$-th largest score

$$\mathbb{I}(h_w(x_i; q) > \lambda_q(w))$$

$$\lambda_q(w) = \arg\min_{\lambda} \frac{K + \varepsilon}{n} \lambda + \frac{1}{n} \sum_{i=1}^{n} (h_w(x_i; q) - \lambda)_+$$
New Formulation

Multi-block Bilevel Optimization

\[
\min \frac{1}{S} \sum_{(q, x_i^q) \in S} \sigma(h_w(x_i^q; q) - \lambda_q(w)) f(g_{q,i}(w)) \\
\text{s.t.} \quad \lambda_q(w) = \arg\min_{\lambda} L_q(\lambda, w, S_q), \forall q \in Q
\]

\[f(g_i(w))\]
\[ \nabla \sigma(h_w(x^q_i; q) - \lambda_q(w))(\nabla h_w(x^q_i; q) - \nabla \lambda_q(w)) \]

- Depends on $S_q$
- Implicit Gradient
Tackle Challenges (K-SONG)

(ICML 2022)

\[ \lambda_q(w_t) \]

Depends on \( S_q \)

\[ \lambda_{q}^{t+1} = \lambda_q^t - \eta_0 \nabla_{\lambda} L_q(\lambda_q^t, w_t, B_2^t) \]

SGD on Lower Level

Only for Sampled Query

\[ \nabla \lambda_q(w_t) \]

Implicit Gradient

Smoothing Lower Level

Estimating Hessian Inverse

Using Mini-batches
Theories

Goal

\[ \| \nabla F(w) \| \leq \epsilon \]

\[
O \left( \frac{1}{\epsilon^4} \right)
\]
Table 2: The test NDCG on two Learning to Rank datasets. We report the average NDCG@$k$ ($k \in [10, 30, 60]$) and standard deviation (within brackets) over 5 runs with different random seeds.

<table>
<thead>
<tr>
<th>Method</th>
<th>MSLR WEB30K</th>
<th>Yahoo! LTR Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NDCG@10</td>
<td>NDCG@30</td>
</tr>
<tr>
<td>RANKNET</td>
<td>0.5227±0.0012</td>
<td>0.5837±0.0006</td>
</tr>
<tr>
<td>LISTNET</td>
<td>0.5337±0.0022</td>
<td>0.5910±0.0019</td>
</tr>
<tr>
<td>LISTMLE</td>
<td>0.5210±0.0017</td>
<td>0.5800±0.0015</td>
</tr>
<tr>
<td>LAMBDA RANK</td>
<td>0.5324±0.0037</td>
<td>0.5885±0.0032</td>
</tr>
<tr>
<td>APPROXNDCC</td>
<td>0.5339±0.0008</td>
<td>0.5906±0.0005</td>
</tr>
<tr>
<td>NEURALNDCC</td>
<td>0.5329±0.0027</td>
<td>0.5881±0.0013</td>
</tr>
<tr>
<td>SONG</td>
<td>0.5382±0.0007</td>
<td>0.5953±0.0006</td>
</tr>
<tr>
<td>K-SONG</td>
<td><strong>0.5397±0.0009</strong></td>
<td><strong>0.5955±0.0004</strong></td>
</tr>
</tbody>
</table>

Table 4: The test NDCG on two movie recommendation datasets. We report the average NDCG@$k$ ($k \in [10, 20, 50]$) and standard deviation (within brackets) over 5 runs with different random seeds.

<table>
<thead>
<tr>
<th>Method</th>
<th>MOVIELENS20M</th>
<th>NETFLIX PRIZE DATASET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NDCG@10</td>
<td>NDCG@20</td>
</tr>
<tr>
<td>RANKNET</td>
<td>0.0109±0.0011</td>
<td>0.0190±0.0010</td>
</tr>
<tr>
<td>LISTNET</td>
<td>0.0182±0.0004</td>
<td>0.0305±0.0002</td>
</tr>
<tr>
<td>LISTMLE</td>
<td>0.0117±0.0005</td>
<td>0.0210±0.0011</td>
</tr>
<tr>
<td>LAMBDA RANK</td>
<td>0.0178±0.0010</td>
<td>0.0310±0.0008</td>
</tr>
<tr>
<td>APPROXNDCC</td>
<td>0.0202±0.0004</td>
<td>0.0338±0.0004</td>
</tr>
<tr>
<td>NEURALNDCC</td>
<td>0.0194±0.0013</td>
<td>0.0322±0.0011</td>
</tr>
<tr>
<td>SONG</td>
<td>0.0232±0.0003</td>
<td>0.0369±0.0004</td>
</tr>
<tr>
<td>K-SONG</td>
<td><strong>0.0248±0.0003</strong></td>
<td><strong>0.0381±0.0003</strong></td>
</tr>
</tbody>
</table>
Movielens: 20 Millions User-Movie Pairs

Comparison on training time per epoch

ListNet as X-risk
Self-supervised Learning
Self-supervised learning
SimCLR: Simple Contrastive Learning

A Simple Framework for Contrastive Learning of Visual ... - arXiv
by T Chen · 2020 · Cited by 3849 — Abstract: This paper presents SimCLR: a simple framework for contrastive learning of visual representations. We simplify recently proposed ...
Mini-batch Contrastive Loss

\[ L_B(w; x_i, A, A') = -\ln \frac{\exp(E(A(x_i))^\top E(A'(x_i))/\tau)}{\sum_{z_j \in B_i} (\exp(E(A(x_i))^\top E(z_j))/\tau)} \]
Issue of SimCLR

Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch. Chen et al. 2020

Huge Difference between large batch and small batch
Our Contributions:

(1) Explanation of Large Batch of SimCLR

(2) New Method SogCLR without Large Batch Size
How do we understand the issue of SimCLR?

Global Contrastive Loss is the Key

$$L(w; x_i, A, A') = -\ln \frac{\exp(E(A(x_i))^\top E(A'(x_i))/\tau)}{\sum_{z \in S_i} (\exp(E(A(x_i))^\top E(z)/\tau)},$$

All Images Except $x_i$

Global Contrastive Objective is X-risk

$$F(w) = \mathbb{E}_{x_i \sim \mathcal{D}, A, A' \sim \mathcal{P}} (E(A(x_i))^\top E(A'(x_i))) + \frac{\tau}{n} \sum_{x_i \in \mathcal{D}} \mathbb{E}_{A} \ln \left( \frac{1}{|S_i|} g(w; x_i, A, S_i) \right),$$

$$f(g(w; x_i, A, S_i))$$
SimCLR Suffers from Small Batch Size

\[
\frac{1}{n} \sum_{x_i \in D} \mathbb{E}_A f(g(w; x_i, A, S_i))
\]

\[
\nabla f(g(w; x_i, A, S_i)) \nabla g(w; x_i, A, S_i)
\]

SimCLR uses the Standard learning Paradigm

\[
\nabla f(g(w; x_i, A, B_i)) \nabla g(w; x_i, A, B_i)
\]

\[
\mathbb{E}[\|\nabla F(w)\|] \leq O\left(\frac{1}{\sqrt{B}}\right)
\]

Mini-batch
Better way to Optimize GCL: SogCLR

Estimating inner $g$

$$\nabla f(g(w; x_i, A, S_i)) \nabla g(w; x_i, A, S_i)$$

Maintain and update $u(x_i, A)$?  
Too Much Memory

$u(x_i)$
SogCLR

Update $u$

$$\mathbf{u}_{i,t} = (1 - \gamma)\mathbf{u}_{i,t-1} \quad \text{Mini-batch}$$

$$+ \frac{1}{\gamma} \frac{1}{2|\mathcal{B}_i|} \left( g(\mathbf{w}_t; \mathbf{x}_i, \mathcal{A}, \mathcal{B}_i) + g(\mathbf{w}_t; \mathbf{x}_i, \mathcal{A}', \mathcal{B}_i) \right),$$

Compute Gradient Estimator

$$\mathbf{m}_t = -\frac{1}{B} \sum_{\mathbf{x}_i \in \mathcal{B}} \nabla \left( \mathcal{E}(\mathcal{A}(\mathbf{x}_i))^\top \mathcal{E}(\mathcal{A}'(\mathbf{x}_i)) \right)$$

$$+ \nabla \mathcal{f}(\mathbf{u}_{i,t-1}) \frac{1}{2|\mathcal{B}_i|} \left( \nabla g(\mathbf{w}_t; \mathbf{x}_i, \mathcal{A}, \mathcal{B}_i) + \nabla g(\mathbf{w}_t; \mathbf{x}_i, \mathcal{A}', \mathcal{B}_i) \right).$$

This is the Key

Update $w$

$$\mathbf{v}_t = (1 - \beta)\mathbf{v}_{t-1} + \beta \mathbf{m}_t$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{v}_t \text{ (or use Adam-style update)}$$
Theory of SogCLR

Theorem 1

$$\mathbb{E}[\|\nabla F(w_{t'})\|^2] \leq O\left(\frac{1}{\sqrt{BT}} + \frac{\sqrt{n}}{B\sqrt{T}} + \epsilon^2\right)$$

Theorem 2

$$L_2(w; x_i, A, A') = -\ln \frac{\exp(E(A(x_i))^\top E(A'(x_i))/\tau)}{\mathbb{E}_{Ag}(w; x_i, A, S_i)}.$$
Experiments

Table 6: Comparison of small-batch training approaches.

<table>
<thead>
<tr>
<th>Method</th>
<th>Batch Size/Epochs</th>
<th>100</th>
<th>200</th>
<th>400</th>
<th>800</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimCLR</td>
<td>256</td>
<td>69.7</td>
<td>73.6</td>
<td>76.1</td>
<td>77.4</td>
</tr>
<tr>
<td>FlatNCE</td>
<td>256</td>
<td>71.5</td>
<td>75.5</td>
<td>76.7</td>
<td>77.8</td>
</tr>
<tr>
<td>SiMo</td>
<td>256</td>
<td>71.5</td>
<td>75.0</td>
<td>76.8</td>
<td>78.2</td>
</tr>
<tr>
<td>SogCLR</td>
<td>256</td>
<td><strong>71.9</strong></td>
<td><strong>76.3</strong></td>
<td><strong>78.7</strong></td>
<td><strong>79.4</strong></td>
</tr>
</tbody>
</table>

Table 1: Comparison of different InfoNCE-loss based contrastive learning methods and their top-1 linear evaluation accuracy by using 800 epochs, a batch size of 256, and ResNet-50 on ImageNet-1K. Momentum encoder is introduced by MoCo [20]. We expect the performance of SogCLR can be further improved by incorporating other techniques, e.g., InfoMin augmentation.

<table>
<thead>
<tr>
<th>Method</th>
<th>Batch Size</th>
<th>Memory Bank</th>
<th>Momentum Encoder</th>
<th>Other Tricks</th>
<th>Convergence</th>
<th>Top1 Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimCLR [4]</td>
<td>Large-batch</td>
<td>No</td>
<td>No</td>
<td>Strong Aug.</td>
<td>No</td>
<td>66.5</td>
</tr>
<tr>
<td>NNCLR [15]</td>
<td>Large-batch</td>
<td>No</td>
<td>No</td>
<td>Nearest Neighbors</td>
<td>No</td>
<td>68.7</td>
</tr>
<tr>
<td>SiMo [44]</td>
<td>Small-batch</td>
<td>No</td>
<td>Yes</td>
<td>Margin Trick</td>
<td>No</td>
<td>72.1</td>
</tr>
<tr>
<td>MoCoV2 [6]</td>
<td>Small-batch</td>
<td>Yes</td>
<td>Yes</td>
<td>Strong Aug.</td>
<td>No</td>
<td>71.1</td>
</tr>
<tr>
<td>InfoMin [36]</td>
<td>Small-batch</td>
<td>Yes</td>
<td>Yes</td>
<td>InfoMin Aug.</td>
<td>No</td>
<td>73.0</td>
</tr>
<tr>
<td>SogCLR (Ours)</td>
<td>Small-batch</td>
<td>No</td>
<td>No</td>
<td>GC Optimization</td>
<td>Yes</td>
<td>72.5</td>
</tr>
</tbody>
</table>
Summary: X-risk as a New Learning Paradigm

- **Any Batch Size**
- **Broad Applications**
- **Convergence Guarantee**
- **Easy Implementation**

1. **Sample Mini-batch Samples**
2. **Define Dynamic Mini-batch (MB) Losses**
3. **Back-propagation on Dynamic MB Losses**
4. **Update Model Parameters**
More X-risks

Areas under the Curves
- AUROC
- AUPRC
- One-way Partial AUC
- Two-way Partial AUC

Ranking Measures
- MAP & NDCG
- P-norm Push
- Listwise Loss
- Top-K MAP & NDCG
- Top Push
- Recall@K
- Precision@Recall

Performance at the Top
- Top-K MAP & NDCG
- Top Push
- Recall@K
- Precision@Recall

Contrastive Objectives
- Self-supervised (e.g., SimCLR, CLIP)
- Supervised (e.g., NCA)

X-risk

Min-Max Opt.

Finite-Sum Coupled Compositional Opt.

Bilevel Opt.
A DEEP LEARNING LIBRARY FOR X-RISK OPTIMIZATION
An open-source library that translates theories to real-world applications

[2022-06] 7 papers about optimization for ML/AI accepted to ICML 2022!

KEY FEATURES & CAPABILITIES

**Easy Installation**
Easy to install and insert LibAUC code into existing training pipeline with Deep Learning frameworks like PyTorch.

**Broad Applications**
Users can learn any neural network structures (e.g., linear, MLP, CNN, GNN, transformer, etc) that support their data types.

**Efficient Algorithms**
Stochastic algorithms with provable theoretical convergence that support learning with millions of data points.

**Hands-on Tutorials**
Hands-on tutorials are provided for optimizing a variety of measures and objectives belonging to the family of X-risks.
Impact of LibAUC Library

QUICK FACTS
The achievements we made so far.

3+
Challenges winning solution (e.g., Stanford CheXpert, MIT AICure, 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).

13000+
Downloaded by more than 13K+ times from over 11 countries.
Acknowledgements: Students

Main Development

Zhuoning Yuan
PhD Student
University of Iowa

Zi-Hao Qiu
PhD Student
Nanjing University

Dixian Zhu
PhD Student
University of Iowa

Gang Li
PhD Student
University of Iowa
Acknowledgements: Students

Other Contributors

Zhishuai Guo
PhD Student
University of Iowa

Quanqi Hu
PhD Student
University of Iowa

Bokun Wang
PhD Student
University of Iowa

Qi Qi
PhD Student
University of Iowa

Yongjian Zhong
PhD Student
University of Iowa

Mingrui Liu
Assistant Professor
George Mason University

Yan Yan
Assistant Professor
Washington State University

Yi Xu
Associate Professor
Dalian University of Technology
Acknowledgements: Collaborators

- Milan Sonka (Ulowa)
- Nitesh Chawla (ND)
- Hassan Rafique (UIndy)
- Qihang Lin (Ulowa)
- Yiming Ying (UAlbany)
- Shuiwang Ji (TAMU)
Acknowledgements

Big Data, Career, III, RI, Engineering, Smart Health, Fair AI