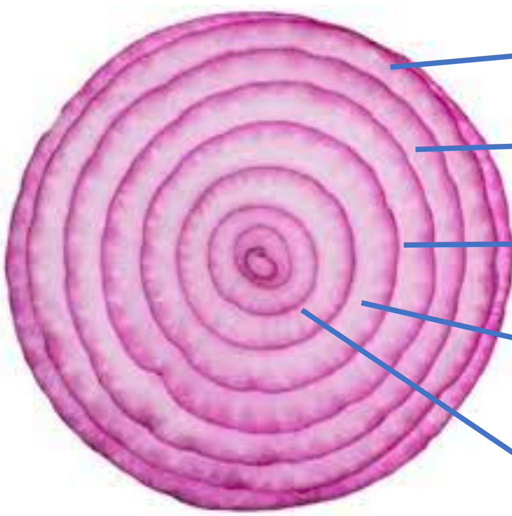
# LibAUC: A Deep Learning Library for X-risk Optimization

Tianbao Yang
University of Iowa

# Outline

- Overview & Background
- Algorithmic Foundation
- Use Cases and Impact

**AI is like an Onion**



**D**omains (e.g., medicine)

**A**pplications (e.g., drug discovery)

**R**epresentations (data, models)

**F**ormulations (objectives)

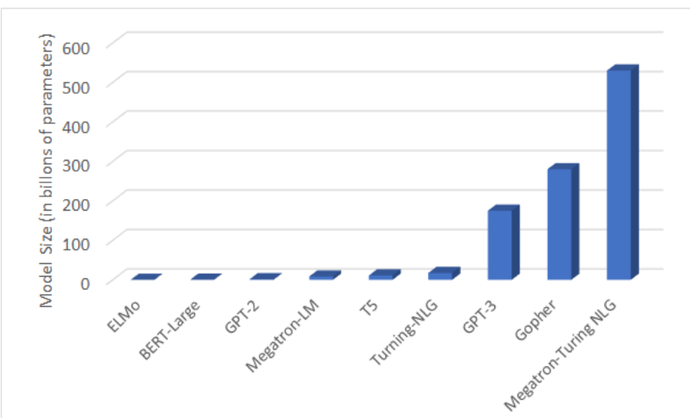
**O**ptimization (algorithms)

## Advancing Optimization to Make ML/AI Faster and Better

**Training Faster**

**Testing better**

# Why Training Matters


Example: GPT-3

175 Billion Parameters

45 TB text data

355 GPU Years

$4.6M

https://lambdalabs.com/blog/demystifying-gpt-3/

**Carbon footprint for 'training GPT-3' same as driving to our natural satellite and back**

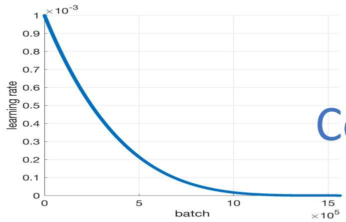
# Optimization for Machine Learning

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

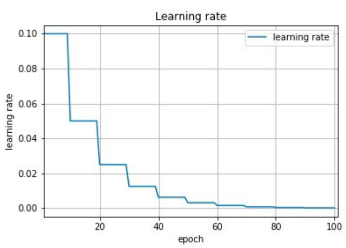
Empirical Risk Minimization (ERM)

# SGD: Stochastic Gradient Descent

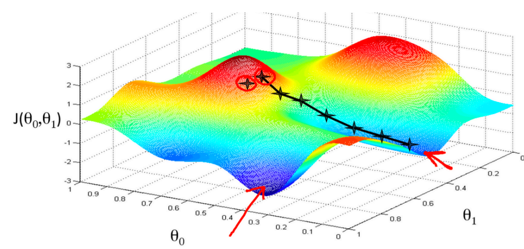
$$\mathbf{w}_{t+1} = \mathbf{w}_t - \boxed{\eta_t} \nabla \ell(\mathbf{w}_t, \mathbf{z}_t)$$




Conventional: Polynomially Decreasing




Modern: Stagewise




Modern: Adaptive

# In the Era of Deep Learning (2012 -)

| | | |
|---|---|---|
| Imagenet classification with deep convolutional neural networks<br>A Krizhevsky, I Sutskever, GE Hinton<br>Advances in neural information processing systems 25, 1097-1105 | 99188 | 2012 |
| On the importance of initialization and momentum in deep learning<br>I Sutskever, J Martens, G Dahl, G Hinton<br>International conference on machine learning, 1139-1147 | 4069 | 2013 |
| Adam: A method for stochastic optimization<br>D Kingma, J Ba<br>International Conference on Learning Representations | 92479 | 2015 |

Stochastic Heavy-ball Method (SHB)

Stochastic Nesterov's Accelerated Gradient (SNAG)

Adam

Momentum term

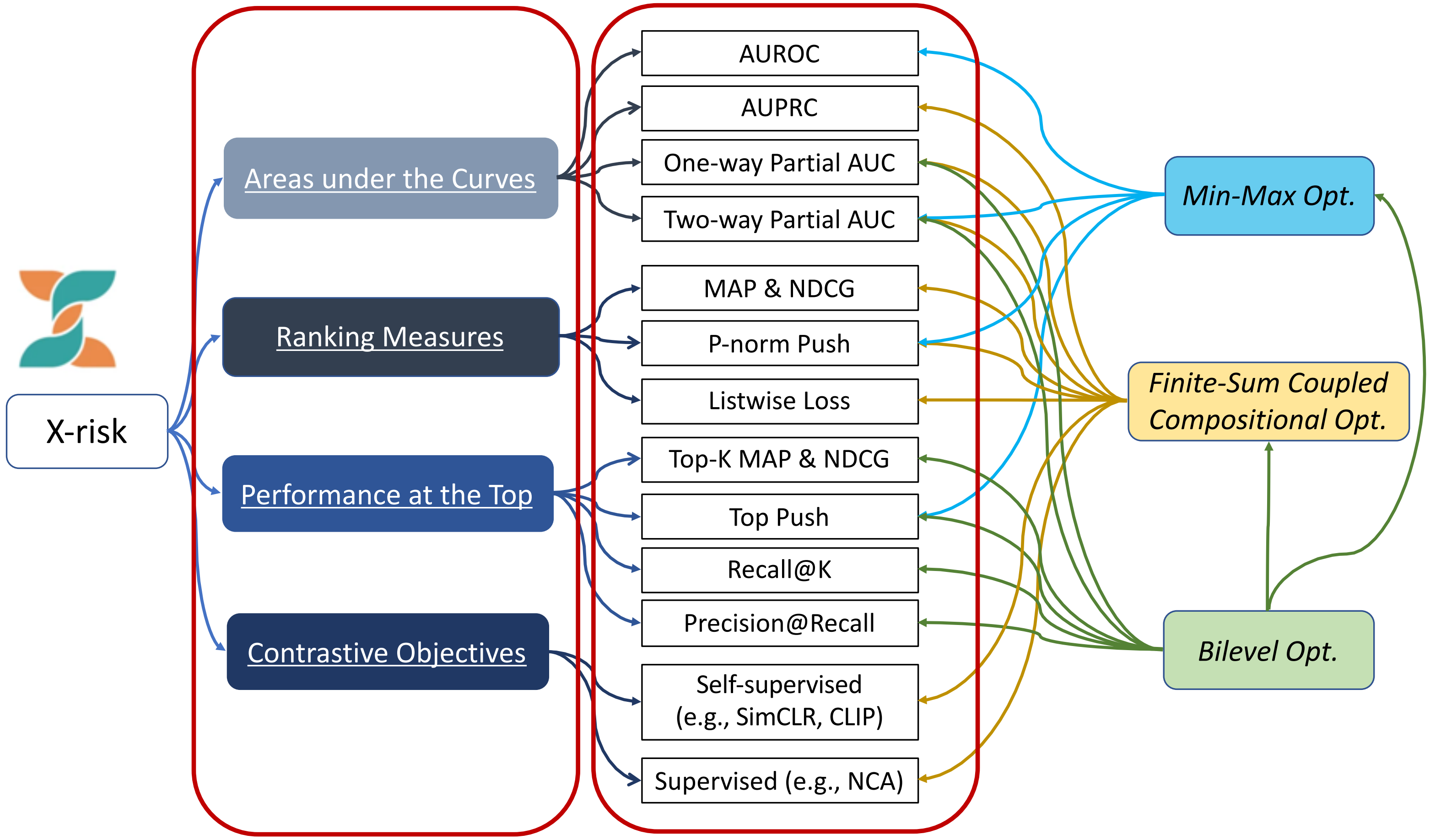
$$\mathbf{w}_{t+1} = \mathbf{w}_t - \boxed{\eta_t} \nabla \ell(\mathbf{w}_t, \mathbf{z}_t) + \boxed{\delta_t}$$

Adaptive or Stagewise

# Beyond ERM: Deep X-risk Optimization

# What is X-risk?

**Compositional** measures that involve **C**omparison between each data and a set of data



# Why are SGD/ADAM NOT Enough?

A set of Samples

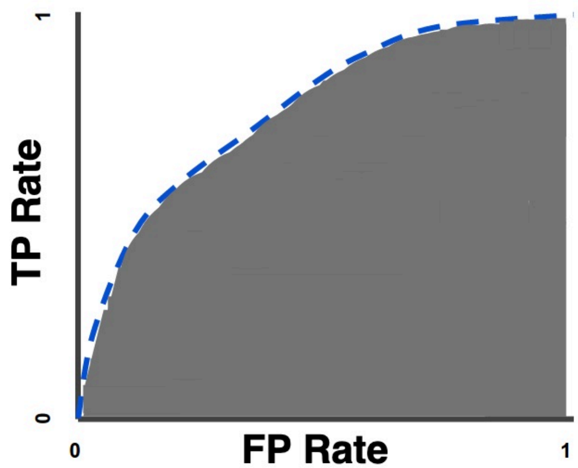
**Compositional** $$F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f(g(\mathbf{w}, \mathbf{z}_i, \mathcal{S}))$$

**C**hallenge: Unbiased Stochastic Gradient is Not Available

# Outline

- **Algorithmic Foundation**
  - Deep AUROC Maximization (Min-max Opt.)
  - Deep AUPRC/AP Maximization (Compositional Opt.)
  - Deep Top-K NDCG Maximization (Bilevel Opt.)
- Use Cases
  - Medical Image Classification
  - Drug Discovery
  - Recommender System

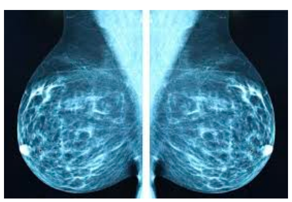
# Deep AUROC Maximization



# Medical Image Diagnosis

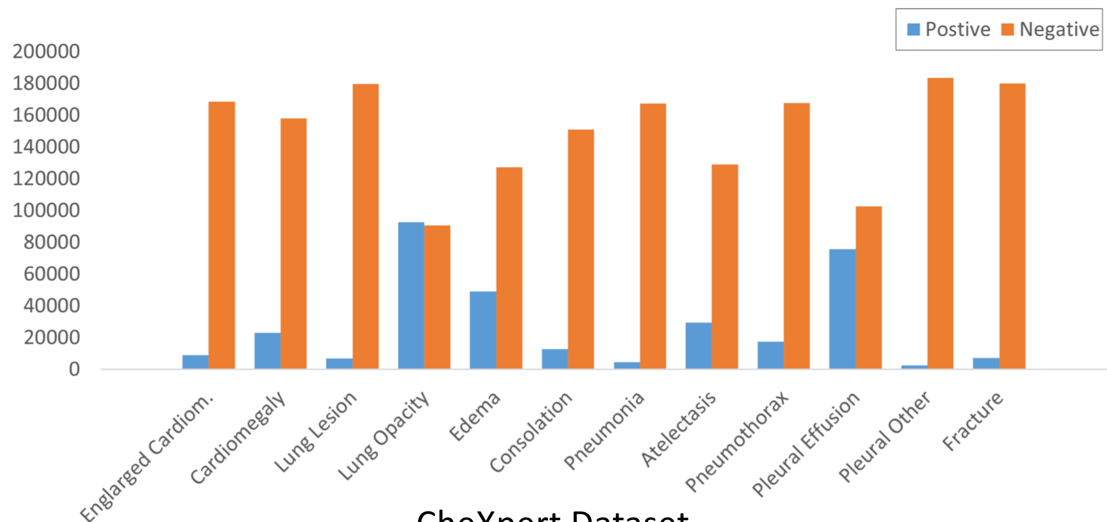
Esteva et al. 2017

Irvin et al. 2019



Wu et al. 2020

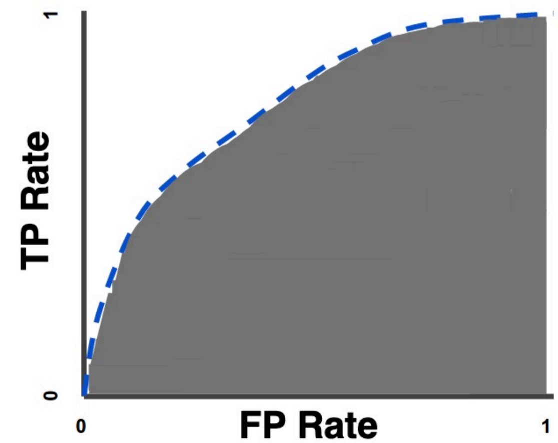
**C**hallenge: Imbalanced Data




CheXpert Dataset

**E**valuation **M**etric: **AUC (ROC)**

# Non-parametric Estimator




$$\widehat{\mathrm{AUC}}(h) = \frac{1}{n_+}\frac{1}{n_-}\sum_{\mathbf{x}_i \in \mathcal{S}_+}\sum_{\mathbf{x}_j \in \mathcal{S}_-} \mathbb{I}(h(\mathbf{x}_i) > h(\mathbf{x}_j))$$

1/0

Model

Positive

Negative

# Formulation: Pairwise Surrogate Loss

**Larger the difference, Larger the Loss**

$$\min_{\mathbf{w}} F(\mathbf{w}) = \frac{1}{n_+}\frac{1}{n_-}\sum_{\mathbf{x}_i \in \mathcal{S}_+}\sum_{\mathbf{x}_j \in \mathcal{S}_-} \ell(h(\mathbf{x}_j) - h(\mathbf{x}_i))$$

Limitations

- Need to Construct Pairs
- Not Suitable for Online Optimization
- Not Suitable for Distributed Optimization

# Deep AUC Maximization (DAM)

**Limitations** of Literature on AUROC Maximization

(1) Linear/Kernelized Models (Convex Analysis) or

(2) Not Scalable to Big Data

**Our Contributions:**

**(1) N**ew Formulation based on Min-Max Opt.

**(2) F**irst Algorithms and Theories for **N**on-Convex **M**in-**M**ax

**(3) O**ptimal Theory and **P**ractical Algorithm

**(4) F**ederated Learning Algorithms

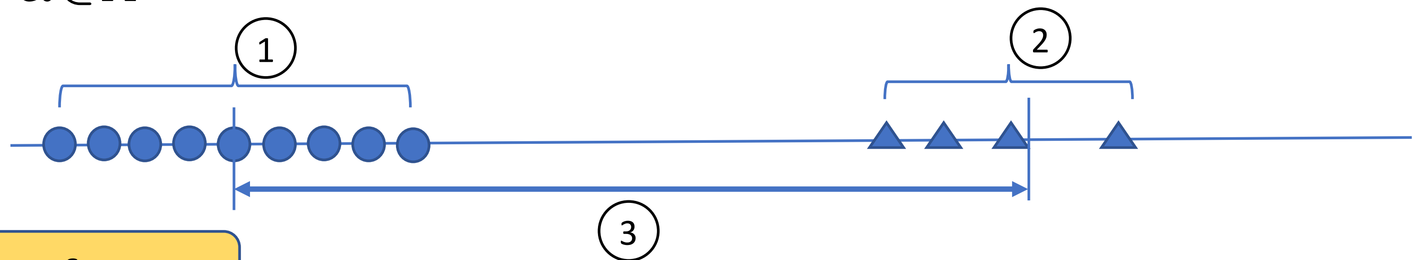


(NeurIPS'19, ICLR'20, ICML'20, ICCV'21, ICML'21, OMS'21, ICLR'22)

# Our Formulation: Min-Max Margin Objective

(ICCV 2021)

$$\min_{\mathbf{w},a,b} \max_{\alpha\in\Omega} F(\mathbf{w}, a, b, \alpha) = \mathbb{E}_{\mathbf{z}}[F(\mathbf{w}, a, b, \alpha; \mathbf{z})]$$



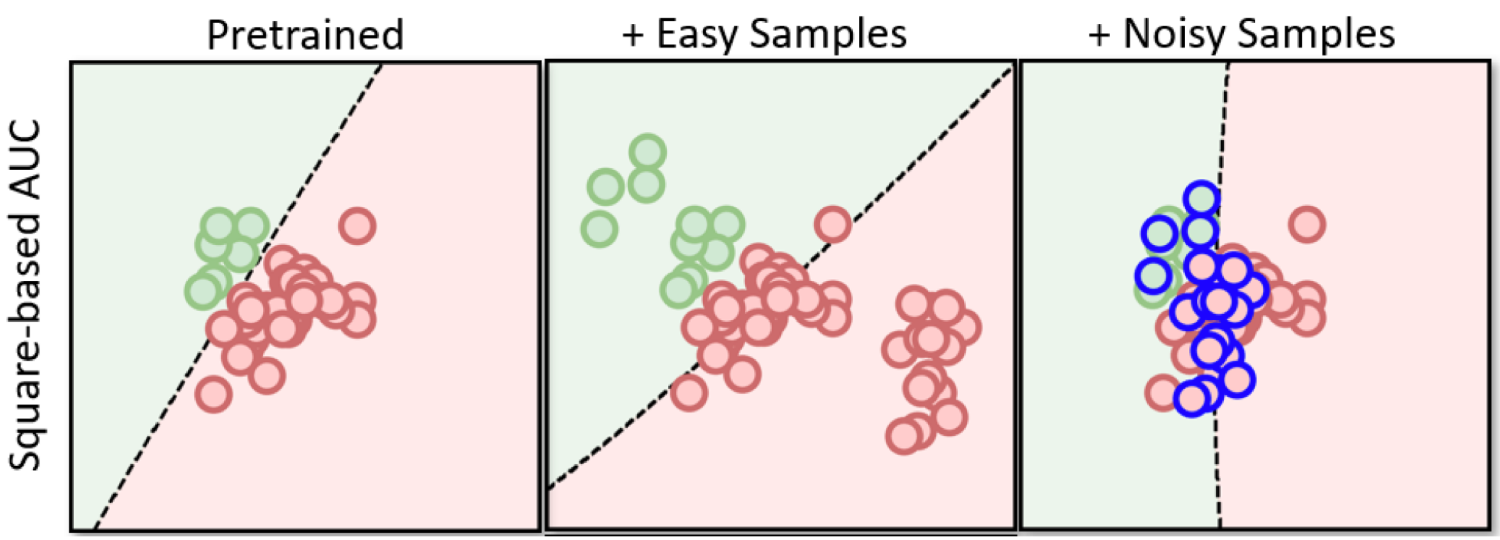

**Benefits**

- No Need to Construct Pairs
- Suitable for Online Optimization
- Suitable for Distributed Optimization
- Equivalent to pairwise square loss $\alpha \in \mathbb{R}$
- Could be more robust by modifying $\Omega$

## Limitations of Square Loss

- Adverse Effect on Easy Data
- Sensitive to Noisy Data

$$\ell(h(\mathbf{x}_j) - h(\mathbf{x}_i)) = (h(\mathbf{x}_j) + c - h(\mathbf{x}_i))^2$$



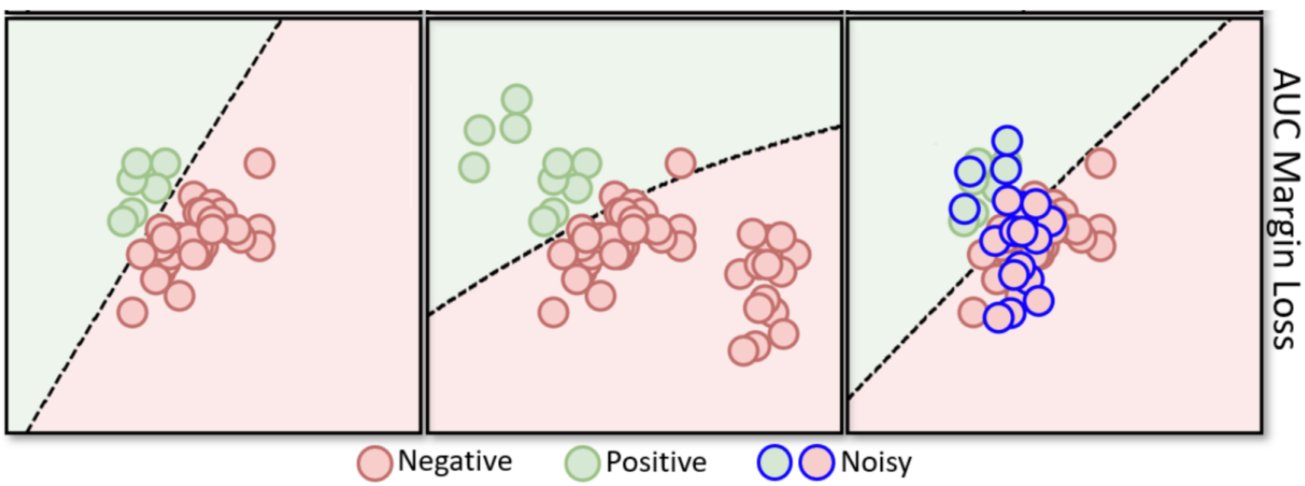
# Our Formulation: Min-Max Margin Objective

(ICCV 2021)

Non-Convex Strongly Concave Min-Max Optimization

$$\min_{\mathbf{w},a,b} \max_{\boxed{\alpha \geq 0}} F(\mathbf{w}, a, b, \alpha) := \mathbb{E}_{\mathbf{z}}\left[F(\mathbf{w}, a, b, \alpha; \mathbf{z})\right],$$

Idea: $(a(\mathbf{w}) - b(\mathbf{w}) - c)^2 \Longrightarrow \max(0, a(\mathbf{w}) - b(\mathbf{w}) - c)^2$



# Algorithm (PESG)

$$\min_{\mathbf{w}} \max_{\alpha \in \Omega} F(\mathbf{w}, \alpha) = \mathbb{E}_{\mathbf{z}}[F(\mathbf{w}, \alpha; \mathbf{z})]$$

Make Non-Convex Function Convex

For $k$=1, … $K$

Step 1: Construct $F_k(\mathbf{w}, \alpha) = F(\mathbf{w}, \alpha) + \frac{\gamma}{2}\|\mathbf{w} - \mathbf{w}_0^k\|^2$

Step 2: Initialize $\boxed{\alpha_0^k}$

Step 3: Solve $(\mathbf{w}_k, \alpha_k) = \mathcal{A}(F_k, \mathbf{w}_0^k, \alpha_0^k, \eta_k, T_k)$

Any Suitable Stochastic Alg.

# Theories

| | Complexity | Goal |
| --- | --- | --- |
| OMS (2018) | $O\left(\frac{1}{\epsilon^4} + \frac{n}{\epsilon^2}\right)$ | $\|\nabla F(\mathbf{w})\| \leq \epsilon$ |
| NeurIPS (2020) | $O\left(\frac{1}{\epsilon^4}\right)$ | $\|\nabla F(\mathbf{w})\| \leq \epsilon$ |
| ICLR (2019)<br>arXiv (2020) | $O\left(\frac{1}{\epsilon}\right)$ | $F(\mathbf{w}) - F_* \leq \epsilon$ |

(ICLR 2020)

## **Purple** and **Blue** are ours

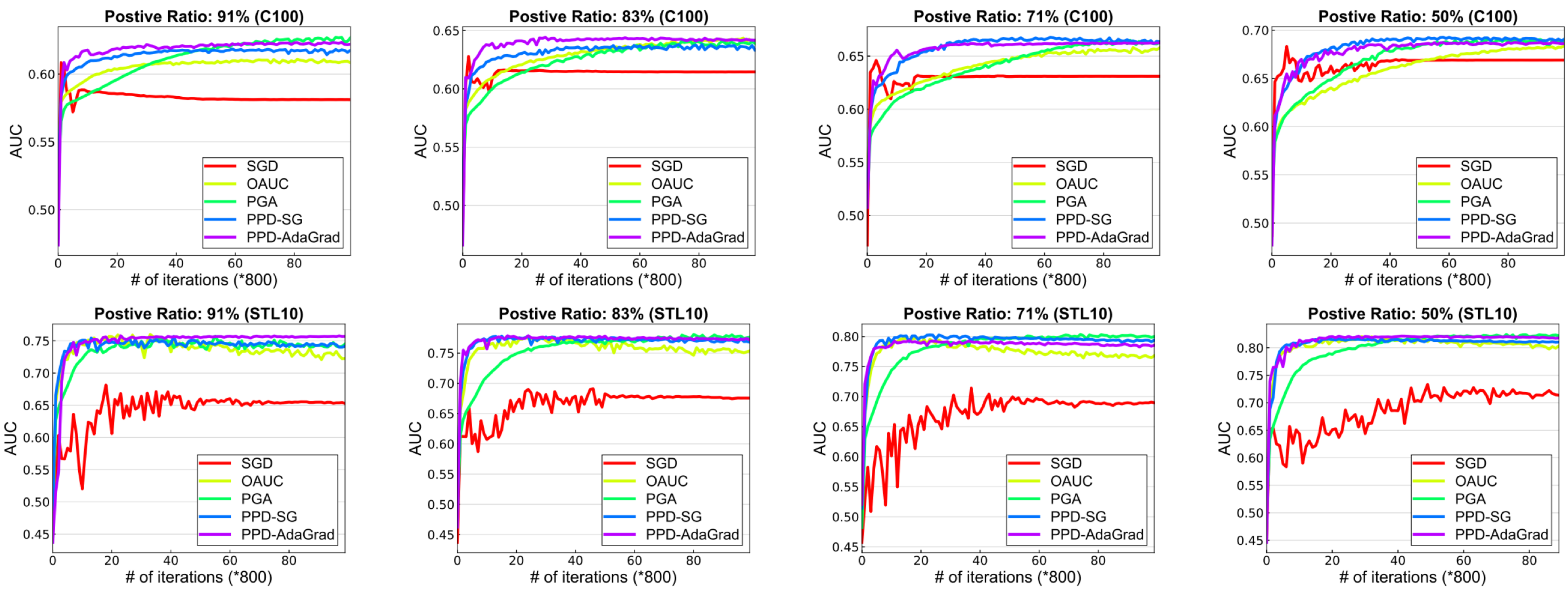
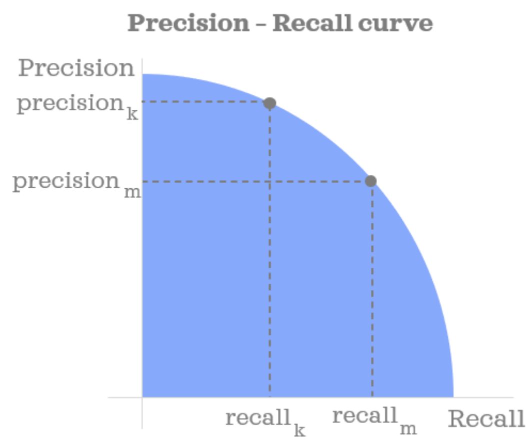




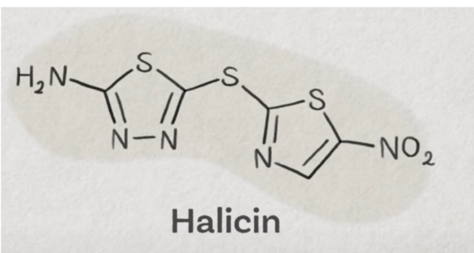

**Image Classification** **Convolutional Neural Networks**

# Deep AUPRC/AP Maximization



MIT AICures Challenge

**Fighting Secondary Effects of Covid**




Stokes et al. 2020. Cell.

## **E**valuation **M**etric: **AUPRC**

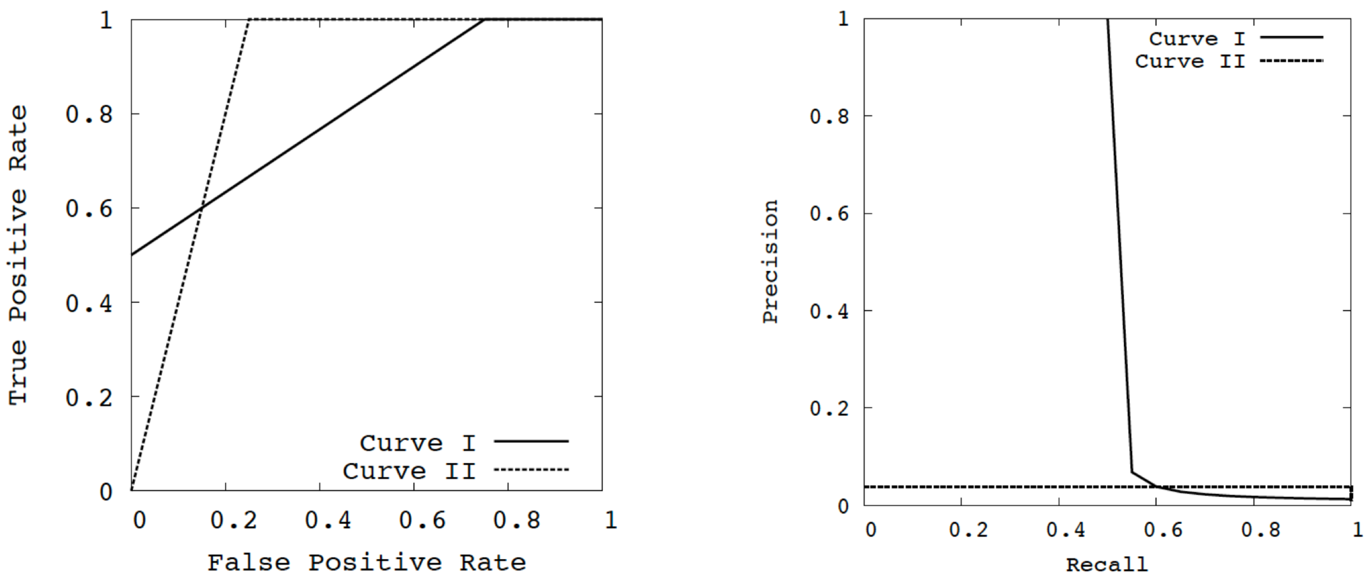
(a) Test PRC-AUC

| 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 |

# Why AUROC Max. is NOT Enough?




**Challenge:** Highly Imbalanced Data

# Non-Parametric Estimator: Average Precision

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

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

# Deep AUPRC Maximization

**Limitations** of Literature on AUPRC Maximization
(1) Small Data or
(2) Heuristic (No Convergence)

**Our Contributions:**
**(1)** **N**ew Formulation based on Compositional Opt.
**(2)** **F**irst Algorithms with Convergence Theory
**(3)** **P**ractical Algorithms and **I**mproved Theory

(NeurIPS’21, AISTATS’22, ICML’22)

# Our Formulation

(NeurIPS 2021)

Precision
$$\frac{\boxed{\sum_{\mathbf{x}_j\in\mathcal{S}_+}\ell(h_{\mathbf{w}}(\mathbf{x}_j)-h_{\mathbf{w}}(\mathbf{x}_i))}}{\boxed{\sum_{\mathbf{x}_j\in\mathcal{S}}\ell(h_{\mathbf{w}}(\mathbf{x}_j)-h_{\mathbf{w}}(\mathbf{x}_i))}} \quad \begin{matrix}\rightarrow [g_i(\mathbf{w})]_1\\ \rightarrow [g_i(\mathbf{w})]_2\end{matrix}$$

**Limitations of Existing Methods**

- Not Convergent or Not-scalable
- Require Large batch size

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

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

**Finite-sum Coupled Compositional Optimization**

# Key Idea of SOAP

$$u_i^t = (1-\beta)u_i^{t-1} + \beta \hat{g}_i(\mathbf{w}_t) \qquad \mathbf{x}_i \in \mathcal{B}_+$$

**Sampled Positive**

Full Gradient $\nabla f(g_i(\mathbf{w}_t))$ at $t^{th}$ iteration

Naïve Mini-batch $\nabla f(\hat{g}_i(\mathbf{w}_t))$ Vs. Variance-reduced $\nabla f(u_i^t)$

**Unbiased**

**Biased but variance-reduced**

# Theories

Goal $\|\nabla F(\mathbf{w})\| \leq \epsilon$

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 $O\left(\frac{1}{\epsilon^4}\right)$

**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 | 0.3236 (± 0.0078) | 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) |
| | SmothAP | 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 | **0.3485 (± 0.0083)** | **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 | **0.0493 (±0.0261)** | **0.3352** (±0.0008) | **0.0236** (±0.0038) |

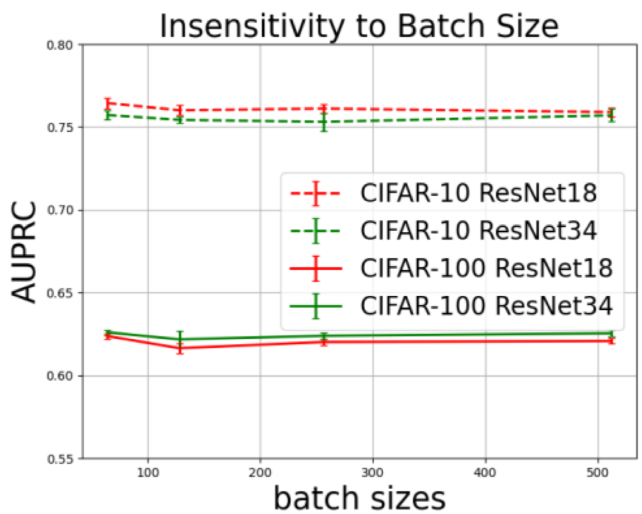
**0.2% Positive** **33% Improvement**

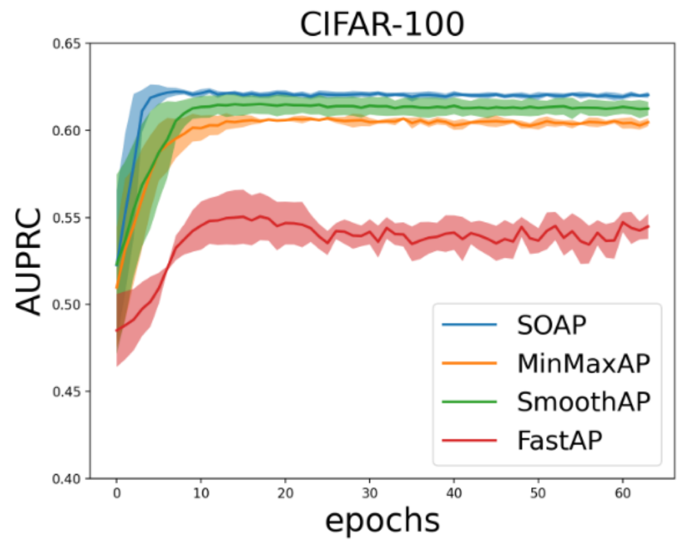
| Data | MIT AICURES | |
|---|---|---|
| Networks | GINE | MPNN |
| CE | 0.5037 (± 0.0718) | 0.6282 (± 0.0634) |
| CB-CE | 0.5655 (± 0.0453) | 0.6308 (± 0.0263) |
| Focal | 0.5143 (± 0.1062) | 0.5875 (± 0.0774) |
| LDAM | 0.5236 (± 0.0551) | 0.6489 (± 0.0556) |
| AUC-M | 0.5149 (± 0.0748) | 0.5542 (± 0.0474) |
| SmothAP | 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**



**Molecular Properties Prediction** **Graph Neural Networks**


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


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

# Deep top-K NDCG Maximization

**M**ost **R**elevant Items on the **T**op

Search Engines

Recommender Systems

Social Media

# NDCG

Relevance Score

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

Ideal Score

Ranking position

# Top-K NDCG

$$\frac{1}{Z_q^K} \sum_{i=1}^{n} \underbrace{\boxed{\mathbb{I}(i\text{-th item in top-K positions})}}_{\text{Top-K selector}} \frac{2^{y_i} - 1}{\underbrace{\log_2(1 + r(i))}_{f(g)}}$$

**Challenges**

- Finding top-K items require O(nlog n)
- Top-K selector is non-differentiable

# Deep top-K NDCG Maximization

**Limitations** of Literature on NDCG Maximization
(1) Small Data or
(2) Not Applicable to Deep Learning

**Our Contributions:** (ICML'22)
**(1) N**ew Formulation based on Bilevel Optimization
**(2) F**irst Algorithms with Convergence Theory
**(3) P**ractical Algorithms

# Transforming Top-K Selector

(ICML 2022)

Prediction score

The (K+1)-th largest score

$$\mathbb{I}(h_{\mathbf{w}}(\mathbf{x}_i; q) > \lambda_q(\mathbf{w}))$$

$$\lambda_q(\mathbf{w}) = \arg\min_{\lambda} \frac{K+\varepsilon}{n}\lambda + \frac{1}{n}\sum_{i=1}^{n}(h_{\mathbf{w}}(\mathbf{x}_i; q) - \lambda)_+$$

# New Formulation

(ICML 2022)

Bilevel Optimization

$$\min \frac{1}{|\mathcal{S}|} \sum_{(q,\mathbf{x}_i^q)\in\mathcal{S}} \sigma(h_{\mathbf{w}}(\mathbf{x}_i^q; q) - \lambda_q(\mathbf{w})) f(g_{q,i}(\mathbf{w}))$$

$$s.t., \lambda_q(\mathbf{w}) = \arg\min_{\lambda} L_q(\lambda; \mathbf{w}), \forall q \in \mathcal{Q}$$

Challenges

- Large number of query-item pairs
- Large number of queries/items

# Algorithms (SONG/K-SONG)

For $t$=1, ... $T$

Step 1: Update $\lambda_q^t$ by one-step SGD

Step 2: Update $u_{q,i}^{(t+1)} = \beta_0 \hat{g}_{q,i}(\mathbf{w}_t) + (1-\beta_0) u_{q,i}^{(t)}$

Step 3: Update $\mathbf{w}$ by a momentum/Adam-style update

# Theories

Goal $$\|\nabla F(\mathbf{w})\| \leq \epsilon$$

ICML'22 $$O\left(\frac{1}{\epsilon^4}\right)$$

Learning to rank

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.

| METHOD | MSLR WEB30K | | | YAHOO! LTR DATASET | | |
|---|---|---|---|---|---|---|
| | NDCG@10 | NDCG@30 | NDCG@60 | NDCG@10 | NDCG@30 | NDCG@60 |
| RANKNET | 0.5227±0.0012 | 0.5837±0.0006 | 0.6481±0.0007 | 0.7668±0.0007 | 0.8319±0.0008 | 0.8491±0.0008 |
| LISTNET | 0.5337±0.0022 | 0.5910±0.0019 | 0.6535±0.0014 | 0.7805±0.0010 | 0.8441±0.0006 | 0.8613±0.0005 |
| LISTMLE | 0.5210±0.0017 | 0.5800±0.0015 | 0.6450±0.0012 | 0.7796±0.0007 | 0.8436±0.0006 | 0.8606±0.0006 |
| LAMBDARANK | 0.5324±0.0037 | 0.5885±0.0032 | 0.6529±0.0026 | 0.7794±0.0009 | 0.8442±0.0008 | 0.8619±0.0007 |
| APPROXNDCG | 0.5339±0.0008 | 0.5906±0.0005 | 0.6530±0.0003 | 0.7688±0.0004 | 0.8367±0.0004 | 0.8556±0.0004 |
| NEURALNDCG | 0.5329±0.0027 | 0.5881±0.0013 | 0.6510±0.0012 | 0.7812±0.0002 | 0.8443±0.0002 | 0.8622±0.0003 |
| SONG | 0.5382±0.0007 | 0.5953±0.0006 | **0.6573**±0.0005 | 0.7842±0.0004 | **0.8477**±0.0003 | **0.8644**±0.0003 |
| K-SONG | **0.5397**±0.0009 | **0.5955**±0.0004 | 0.6571±0.0003 | **0.7859**±0.0003 | 0.8464±0.0002 | 0.8642±0.0003 |

Movie Recommendation

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.

| METHOD | MOVIELENS20M | | | NETFLIX PRIZE DATASET | | |
|---|---|---|---|---|---|---|
| | NDCG@10 | NDCG@20 | NDCG@50 | NDCG@10 | NDCG@20 | NDCG@50 |
| RANKNET | 0.0109±0.0011 | 0.0190±0.0010 | 0.0450±0.0016 | 0.0090±0.0007 | 0.0146±0.0008 | 0.0261±0.0010 |
| LISTNET | 0.0182±0.0004 | 0.0305±0.0002 | 0.0587±0.0004 | 0.0115±0.0018 | 0.0191±0.0013 | 0.0347±0.0014 |
| LISTMLE | 0.0117±0.0005 | 0.0210±0.0011 | 0.0493±0.0010 | 0.0081±0.0005 | 0.0134±0.0009 | 0.0253±0.0005 |
| LAMBDARANK | 0.0178±0.0010 | 0.0310±0.0008 | 0.0595±0.0006 | 0.0103±0.0003 | 0.0175±0.0003 | 0.0332±0.0004 |
| APPROXNDCG | 0.0202±0.0004 | 0.0338±0.0004 | 0.0629±0.0004 | 0.0121±0.0015 | 0.0198±0.0005 | 0.0360±0.0006 |
| NEURALNDCG | 0.0194±0.0013 | 0.0322±0.0011 | 0.0609±0.0012 | 0.0113±0.0011 | 0.0186±0.0008 | 0.0342±0.0007 |
| SONG | 0.0232±0.0003 | 0.0369±0.0004 | 0.0646±0.0003 | 0.0141±0.0004 | 0.0222±0.0005 | **0.0384**±0.0003 |
| K-SONG | **0.0248**±0.0003 | **0.0381**±0.0003 | **0.0662**±0.0004 | **0.0154**±0.0003 | **0.0234**±0.0006 | 0.0377±0.0005 |

# Outline

- Algorithmic Foundation
    - Deep AUROC Maximization (Min-max Opt.)
    - Deep AUPRC/AP Maximization (Compositional Opt.)
    - Deep Top-K NDCG Maximization (Bilevel Opt.)
- **Use Cases and Impact**
    - **Medical Image Classification**
    - Drug Discovery
    - Recommender System

# Stanford CheXpert Competition

**1st Place**

Andrew Ng's Group


150+ Teams Worldwide

## Leaderboard

Will your model perform as well as radiologists in detecting different pathologies in chest X-rays?

| Rank | Date | Model | AUC | Num Rads Below Curve |
|---|---|---|---|---|
| 1 | Aug 31, 2020 | DeepAUC-v1 *ensemble* https://arxiv.org/abs/2012.03173 | **0.930** | 2.8 |
| 2 | Sep 01, 2019 | Hierarchical-Learning-V1 (ensemble) *Vingroup Big Data Institute* https://arxiv.org/abs/1911.06475 | 0.930 | 2.6 |
| 3 | Oct 16, 2019 | Conditional-Training-LSR *ensemble* | 0.929 | 2.6 |
| 4 | Dec 04, 2019 | Hierarchical-Learning-V4 (ensemble) *Vingroup Big Data Institute* https://arxiv.org/abs/1911.06475 | 0.929 | 2.6 |

(ICCV 2021)

| Disease | Image Domain | #pos/#all | # Training | Improvements | Competition Results |
|---|---|---|---|---|---|
| Lung-related | Chest X-ray | 20.21% | 224,316 | 2% | 1/150+ |
| Melanoma | Skin Lesion | 7.1% | 46,131 | 1% | 33/3314 |
| Breast Cancer | Mammogram | 13% | 55,000 | 1.5% | NA |
| Tumor | Microscopic | 1% | 148,960 | 5% | NA |

**Convolutional Neural Networks**

# Outline

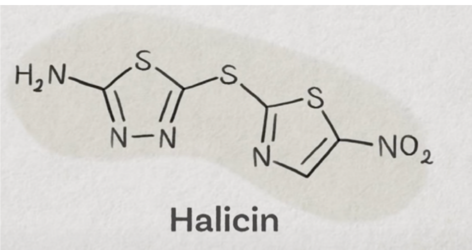
- Algorithmic Foundation
  - Deep AUROC Maximization (Min-max Opt.)
  - Deep AUPRC/AP Maximization (Compositional Opt.)
  - Deep Top-K NDCG Maximization (Bilevel Opt.)
- **Use Cases and Impact**
  - Medical Image Classification
  - **Drug Discovery**
  - Recommender System

# MIT AICures Challenge

**Evaluation Metric: AUPRC**

**1st Place**

**Fighting Secondary Effects of Covid**




Stokes et al. 2020. Cell.

Collaborating with Prof. Shuiwang Ji's group at TAMU

| 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 |

# Comparison with w/o DAM

| 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

| | | | | |
|---|---|---|---|---|
| 1 | MoleculeKit | DIVE@TAMU | 7 | 0.928 |

w/o DAM

AUPRC

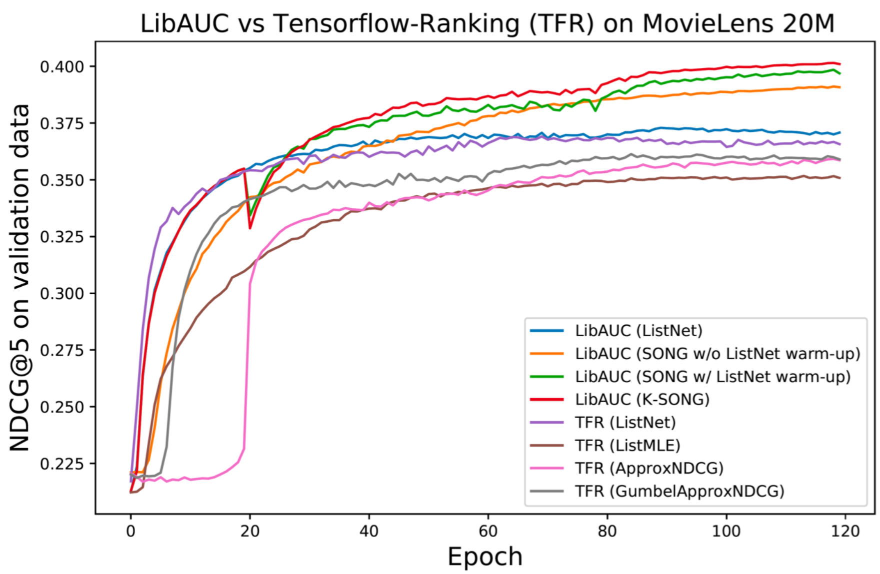
| | | | | |
|---|---|---|---|---|
| 3 | MoleculeKit | DIVE@TAMU | 7 | 0.677 |

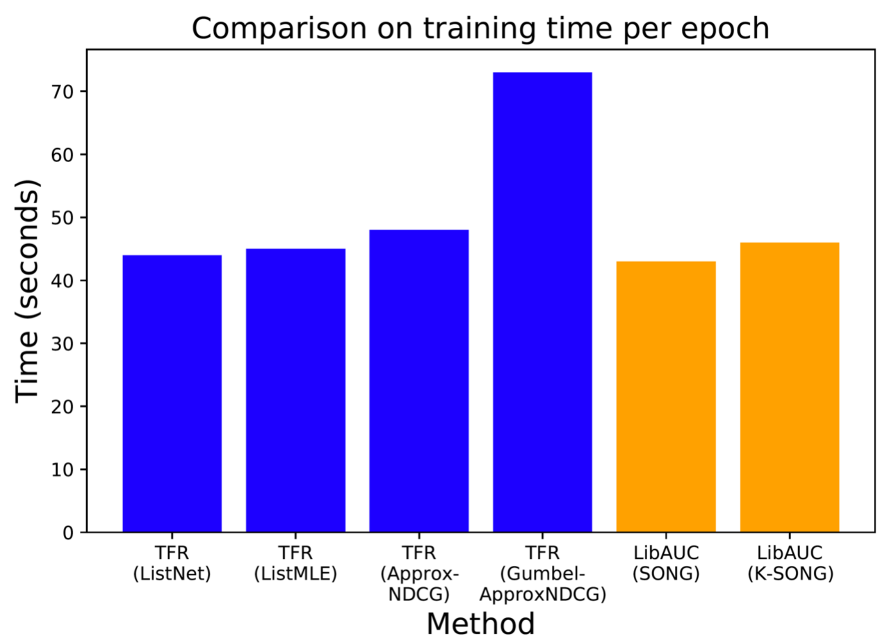
**5%** Improvement in **AUPRC, 3%** Improvement in **AUROC**

# Outline

- Algorithmic Foundation
  - Deep AUROC Maximization (Min-max Opt.)
  - Deep AUPRC/AP Maximization (Compositional Opt.)
  - Deep Top-K NDCG Maximization (Bilevel Opt.)
- **Use Cases and Impact**
  - Medical Image Classification
  - Drug Discovery
  - **Recommender System**

# Movielens: 20 Millions User-Movie Pairs



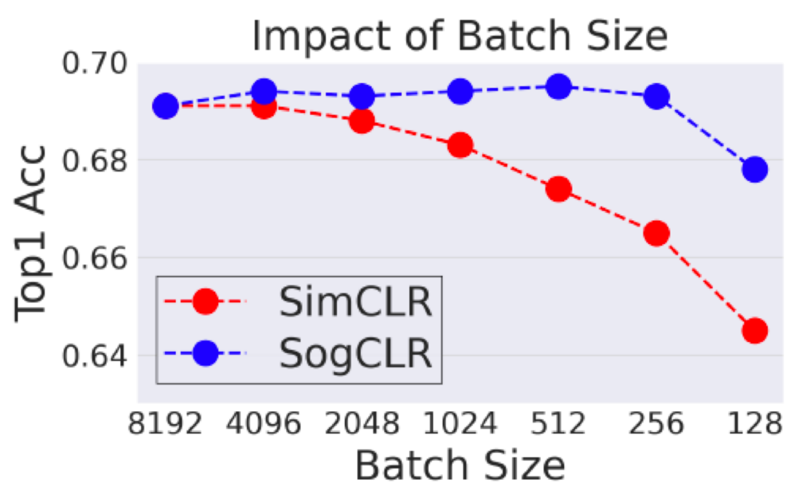



# Other Use Cases: Optimization for BIG Models

Self-supervised Contrastive Learning

(ICML'22, **Collaboration** with Google)

$$\frac{1}{n}\sum_{i=1}^{n} f(g_i(\mathbf{w}))$$

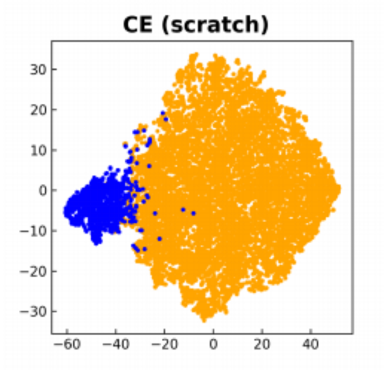



Small batch size Does not hurt Performance

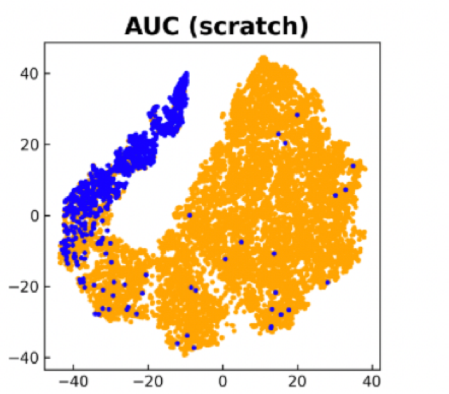
# Deep X Optimization ≠ Non-Convex Optimization

**Representation Learning**

Traditional Loss

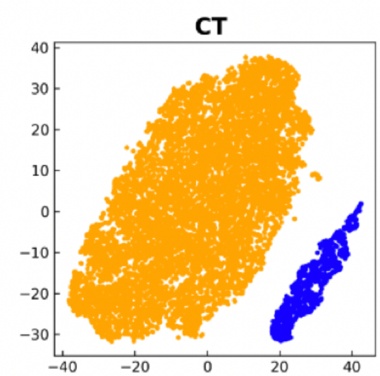



AUC Loss




- Pre-training
- Compositional Training

**End-to-End Training**
**(Compositional Training)**
(ICLR 2022 Spotlight)



# libauc.org

LibAUC

Notifications | Fork 17 | Star 117

LibAUC — Installation | Examples | Research | Talks | Team | Github

## A DEEP LEARNING LIBRARY FOR X-RISK OPTIMIZATION

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

Latest News | Install

[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 AICures, OGB Graph Property Prediction).

**3+**

Collaborations with multiple top industrial units.

**17+**

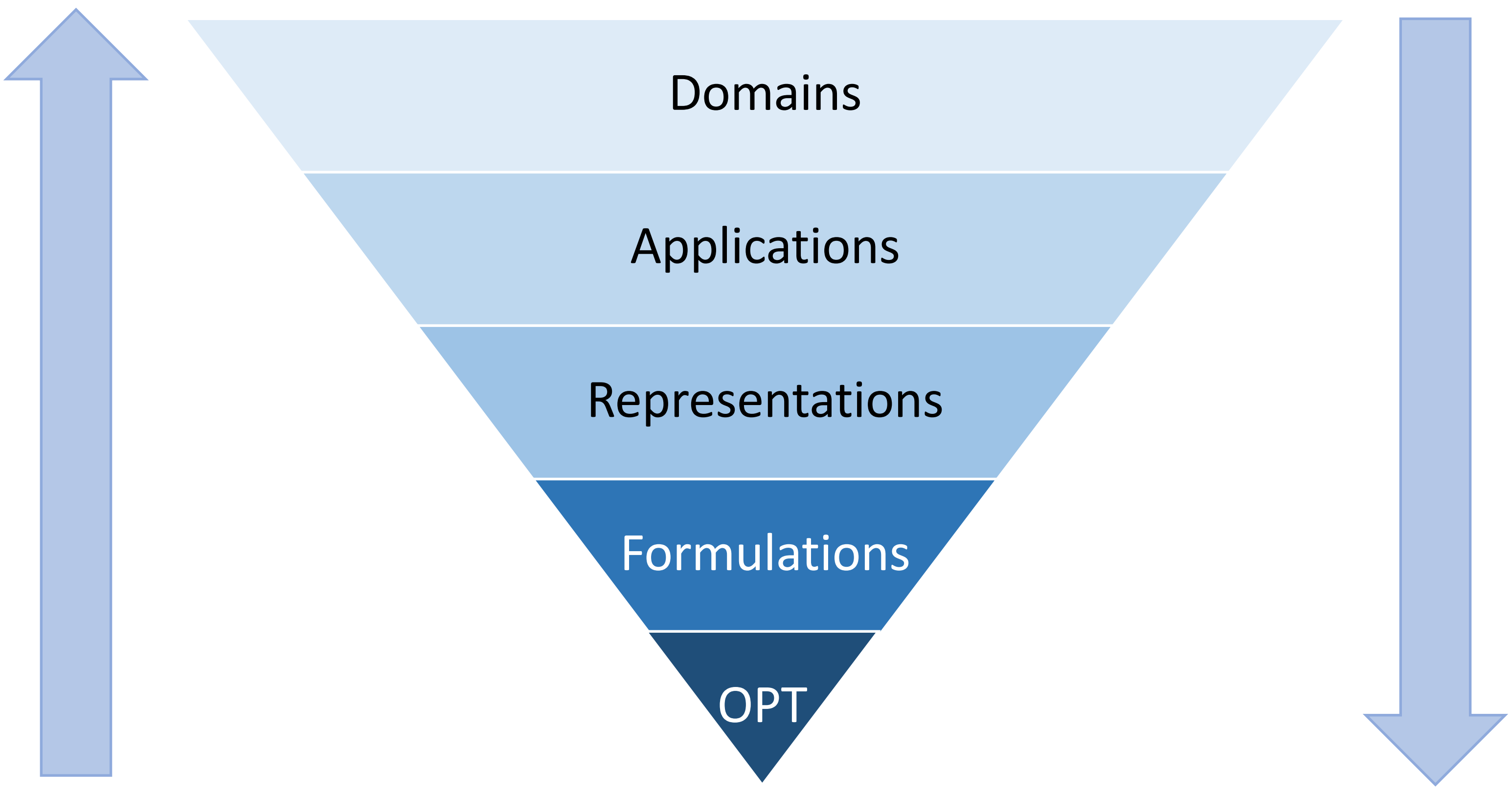
Scientific publications on top-tier AI Conferences (such as ICML, NeurIPS, ICLR).

**11000+**

Downloaded by more than 11K+ times from over 11 countries.

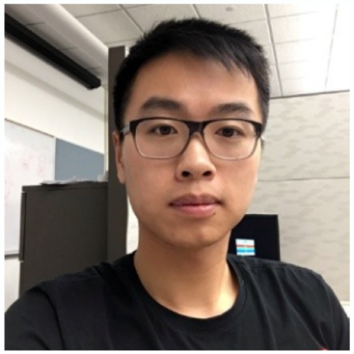
# What is Next

# Deep X-risk Optimization

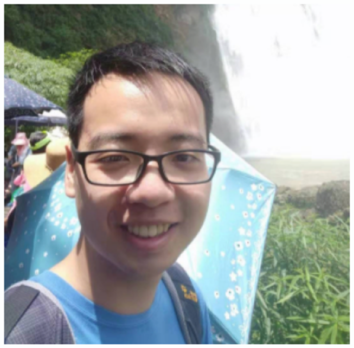


# Acknowledgements: Students

**Main Development**



**Zhuoning Yuan**
PhD Student
University of Iowa



**Zi-Hao Qiu**
PhD Student
Nanjing University

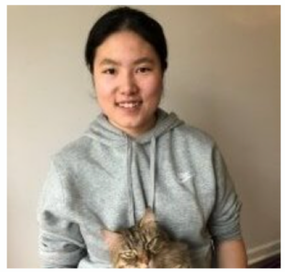
**Dixian Zhu**
PhD Student
University of Iowa

**Gang Li**
PhD Student
University of Iowa

# Acknowledgements: Students

**Other Contributors**

**Zhishuai Guo**
PhD Student
University of Iowa

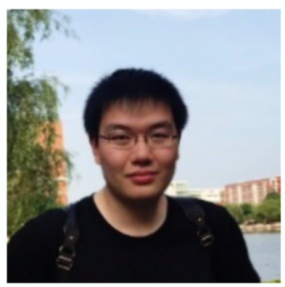


**Quanqi Hu**
PhD Student
University of Iowa

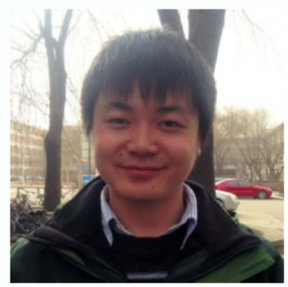

**Bokun Wang**
PhD Student
University of Iowa

**Qi Qi**
PhD Student
University of Iowa

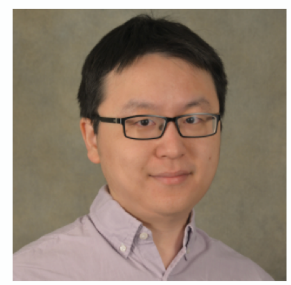
**Yongjian Zhong**
PhD Student
University of Iowa



**Mingrui Liu**
Assistant Professor
George Mason University



**Yan Yan**
Assistant Professor
Washington State University



**Yi Xu**
Associate Professor
Dalian University of Technology

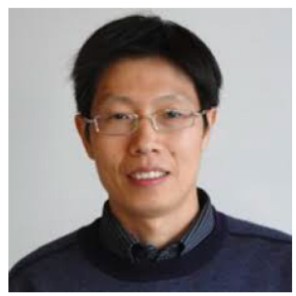
# Acknowledgements: Collaborators

Milan Sonka
(UIowa)

Nitesh Chawla
(ND)

Hassan Rafique
(UIndy)

Qihang Lin
(UIowa)



Yiming Ying
(UAlbany)

Shuiwang Ji
(TAMU)

# Acknowledgements

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

amazon Alibaba Group Tencent 腾讯 Google


Q
A