Learning Structure for Computer Systems Management
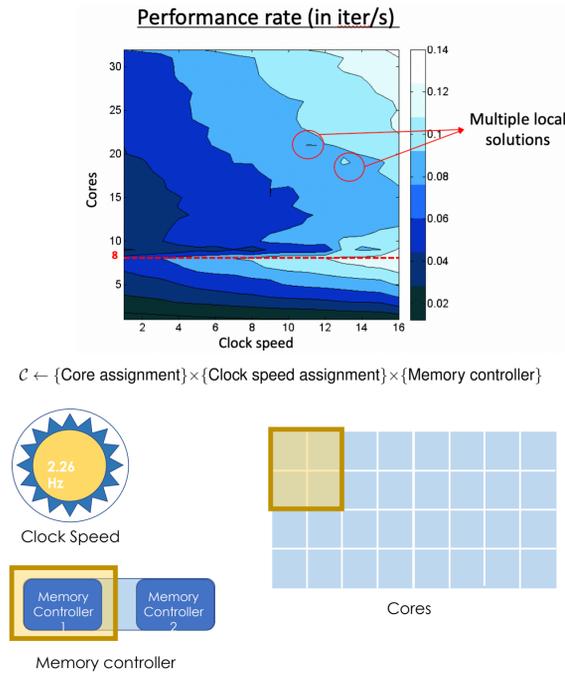
Yi Ding, University of Chicago

I. Introduction

1. Complex Computer Systems
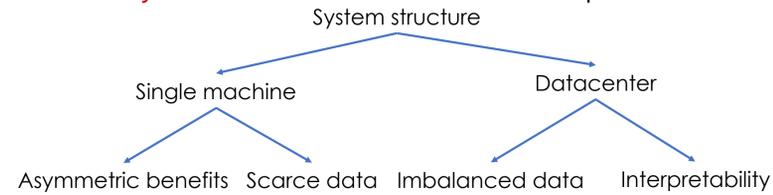

Performance rate (in iter/s)

Multiple local solutions

$\mathcal{C} \leftarrow \{\text{Core assignment}\} \times \{\text{Clock speed assignment}\} \times \{\text{Memory controller}\}$

2.26 Hz
Clock Speed

Memory Controller 1 | Memory Controller 2
Memory controller

Cores

2. Machine Learning for Systems

Robustness → Generalizability → Interpretability

3. My PhD Research

Key ideas:
- Understand system structure for robust and interpretable results

System structure
- Single machine
  - Asymmetric benefits
  - Scarce data
- Datacenter
  - Imbalanced data
  - Interpretability

- Incorporate causal analysis to traditional ML methods

1. M. Gao, Y. Ding, B. Aragam. A Polynomial-time Algorithm for Learning Nonparametric Causal Graphs. NeurIPS, 2020.
2. Y. Ding, P. Toulis. Dynamical Systems Theory for Causal Inference with Application to Synthetic Control Methods. AISTATS, 2020.
3. Y. Ding, N. Mishra, H. Hoffmann. Generative and Multi–phase Learning for Computer Systems Optimization. ISCA, 2019.
4. Y. Ding, R. Kondor, J. Eskreis-Winkler. Multiresolution Kernel Approximation for Gaussian Process Regression. NeurIPS, 2017. (Spotlight)
5. Y. Ding, C. Liu, P. Zhao, S. C. Hoi. Large Scale Kernel Methods for Online AUC Maximization. ICDM, 2017. (Long Oral)
6. Y. Ding, P. Zhao, S. C. Hoi, Y. S. Ong. An Adaptive Gradient Method for Online AUC Maximization. AAAI, 2015. (Oral)

II. Learning for Systems Optimization with Scarce Data and System Structure (ISCA'19)

How to find the optimal system configurations?

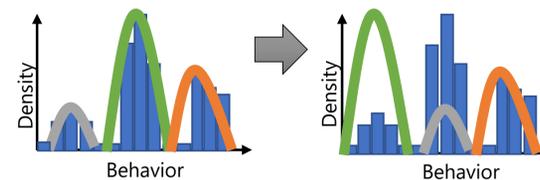### 1. Motivational Example: SRAD on big.LITTLE system
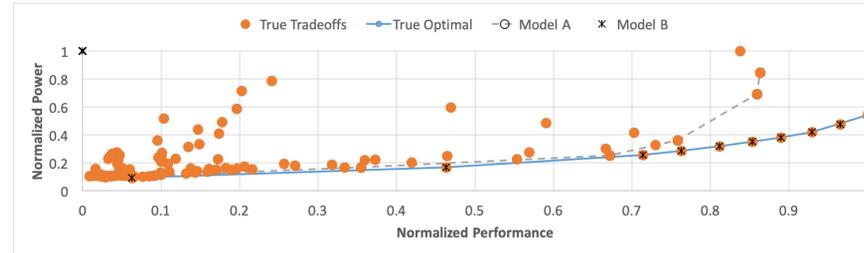
	Model A	Model B
Optimal points	Just far enough	True data
Non-optimal points	True data	Very far
Goodness of fit	99%	0
Energy over optimal	22% ❌	0 ✅

Key Insight:
High accuracy ≠ good system results

2. Improving Accuracy w/ Generative Model

Learn GMMs   **Swap** Max and Min
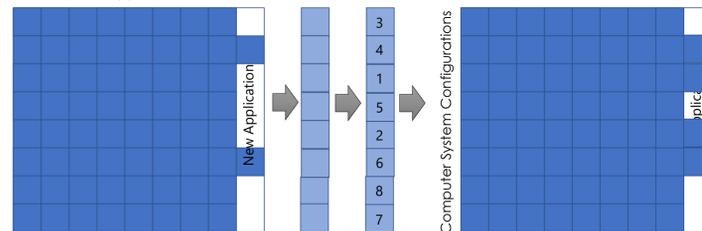

3. Improving Energy Savings w/ Multi-phase Sampling

Input: Configuration-Application data matrix, Sampling budget N

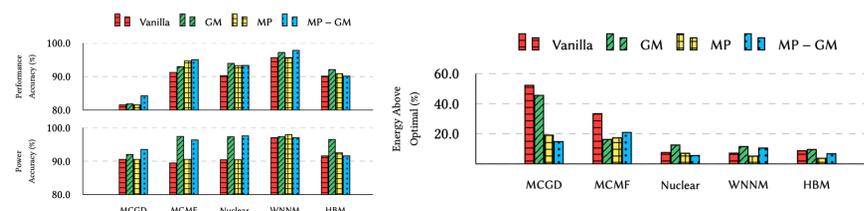Matrix Completion with Sample Size N/2 → Estimated Behavior for New Application → Select N/2 Best Configs → Matrix Completion with N/2 original samples and N/2 estimated best configs
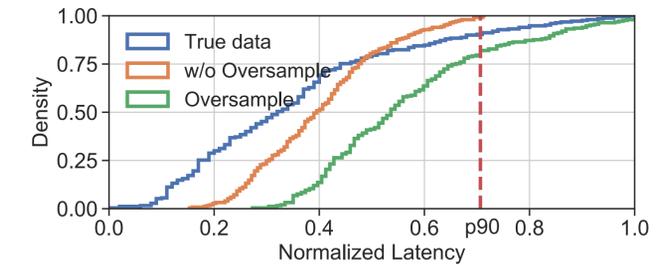
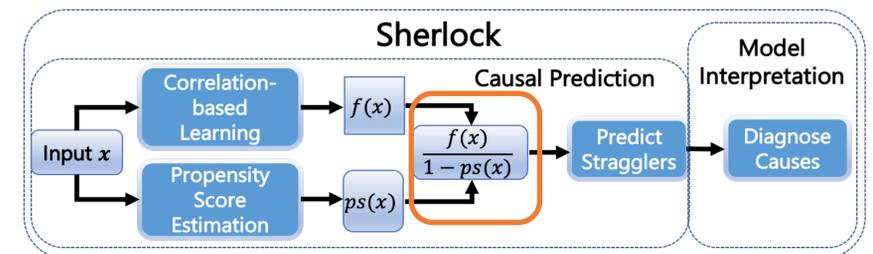### 4. Experimental Results

Vanilla | GM | MP | MP – GM

III. Learning for Straggler Prediction with Imbalanced Data

How to make early and accurate straggler prediction?

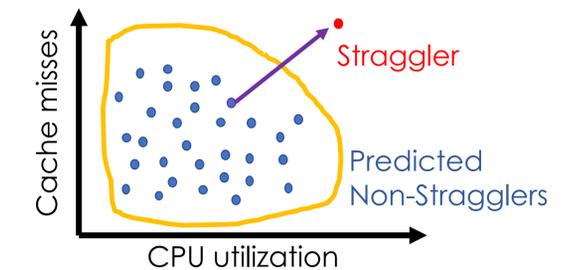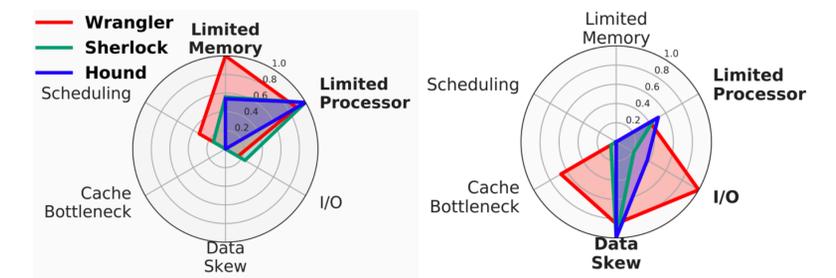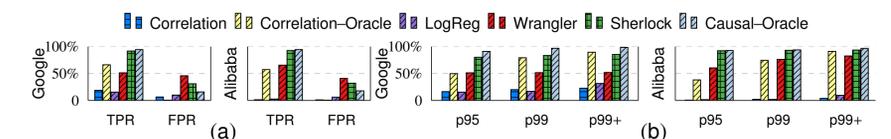### 1. Limitations of Traditional ML for Straggler Prediction

True data | w/o Oversample | Oversample

2. Our Solution: Sherlock

Sherlock: Input x → Correlation-based Learning → $f(x)$; Propensity Score Estimation → $ps(x)$; Causal Prediction $\frac{f(x)}{1 - ps(x)}$ → Predict Stragglers → Model Interpretation: Diagnose Causes

Key Insight:
Adjust predictions w/ propensity score

Cache misses / CPU utilization — Straggler, Predicted Non-Stragglers

3. Experimental Results

Correlation | Correlation–Oracle | LogReg | Wrangler | Sherlock | Causal–Oracle

(a) TPR, FPR — Google, Alibaba
(b) p95, p99, p99+ — Google, Alibaba

Wrangler | Sherlock | Hound

(a) Job 6343946350   (b) Job 6308689702