Generative and Multi-phase Learning for Computer Systems Optimization

Yi Ding, Nikita Mishra, Henry Hoffmann

THE UNIVERSITY OF CHICAGO

Computer Systems Optimization

- Optimizing modern computer systems requires tradeoffs:
  - Deliver reliable performance
  - Minimize energy consumption

Computer Systems Optimization

- Optimizing modern computer systems requires tradeoffs:
  - Deliver reliable performance
  - Minimize energy consumption

- Resource management via system configuration:
  - Resources have complex, non-linear effects on performance and energy
  - Resource interactions create local optima

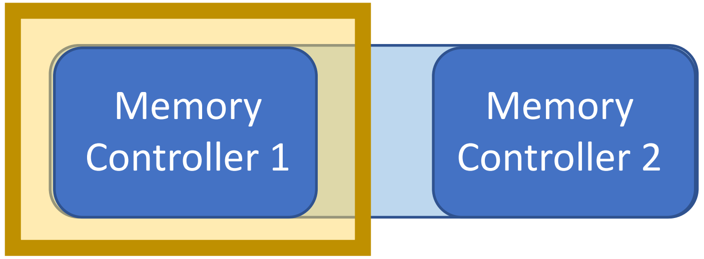
Computer Systems Optimization

- Optimizing modern computer systems requires tradeoffs:
  - Deliver reliable performance
  - Minimize energy consumption
- Resource management via system configuration:
  - Resources have complex, non-linear effects on performance and energy
  - Resource interactions create local optima
- How to find the optimal system configuration?

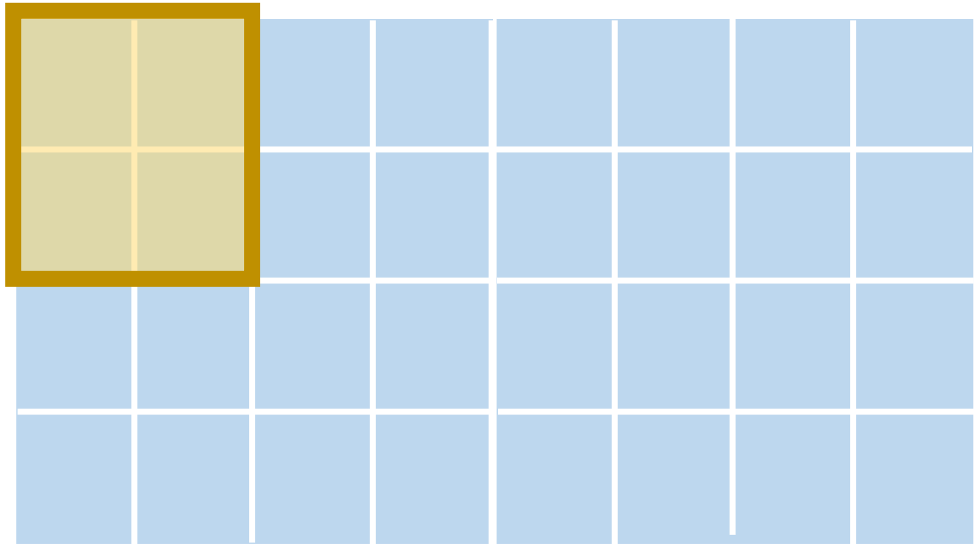
Example of a Configuration Space C

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

Clock Speed



Memory controller



Cores

Machine Learning to the Rescue

Machine Learning to the Rescue

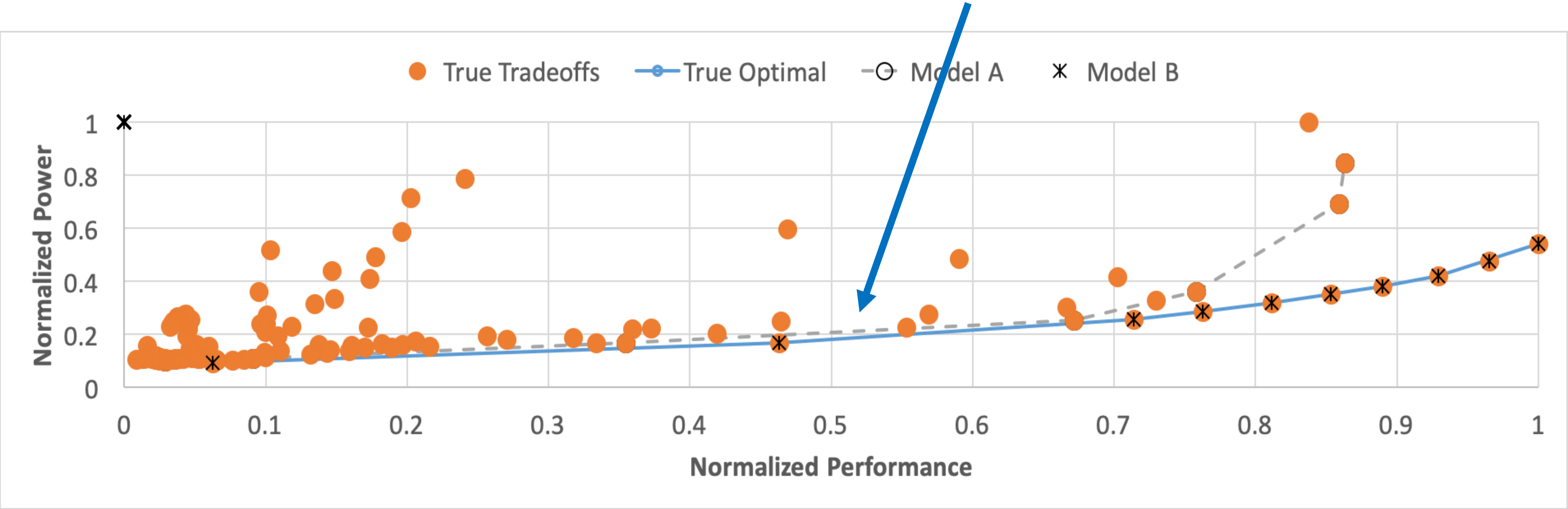
- However…
  - **Scarce data**: expensive collection, limited range behavior

Machine Learning to the Rescue

- However …

  Scarce data: expensive collection, limited range behavior

  Asymmetric benefits: only configs on **optimal frontier** useful

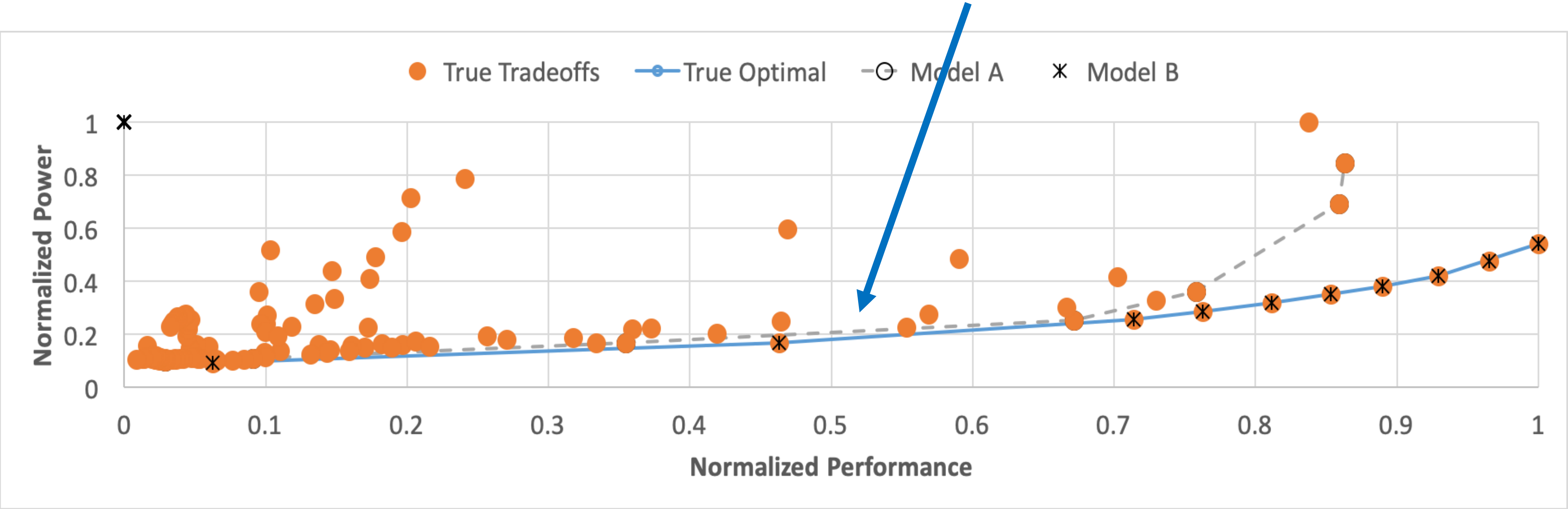


Machine Learning to the Rescue

- However …

Scarce data: expensive collection, limited range behavior →Generative model

Asymmetric benefits: only configs on **optimal frontier** useful →Multi-phase sampling

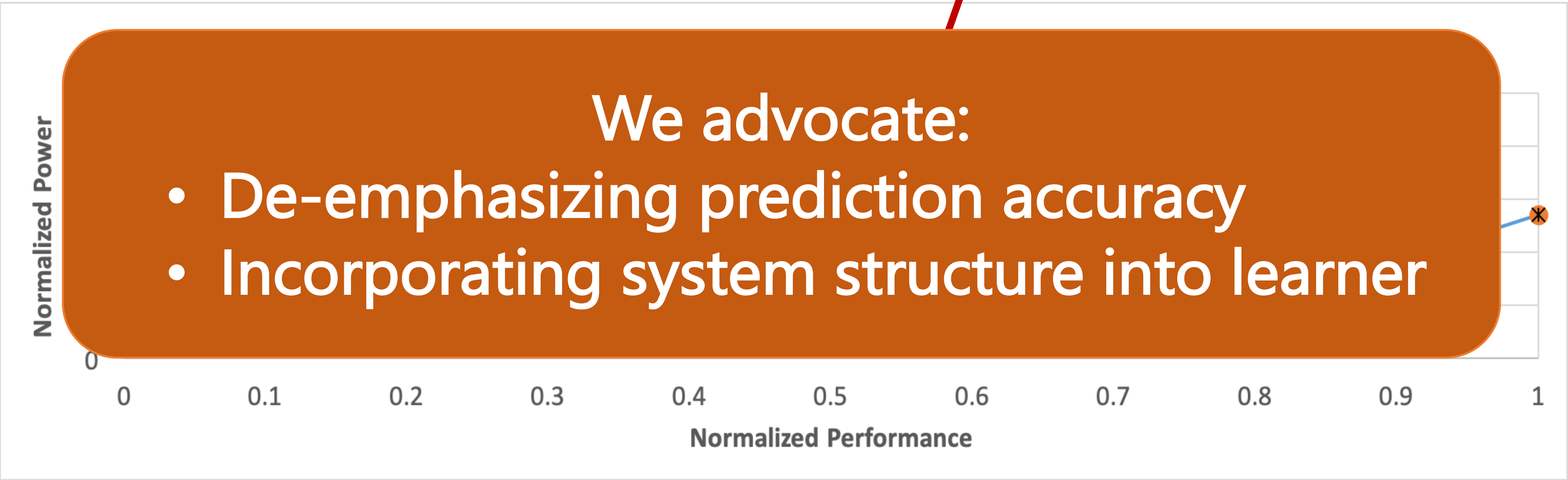


Machine Learning to the Rescue

- However …

  Scarce data: expensive collection, limited range behavior →Generative model

  Asymmetric benefits: only configs on **optimal frontier** useful →Multi-phase sampling



We advocate:

- De-emphasizing prediction accuracy
- Incorporating system structure into learner

Problem Formulation

Meet latency constraints with minimal energy via system configs

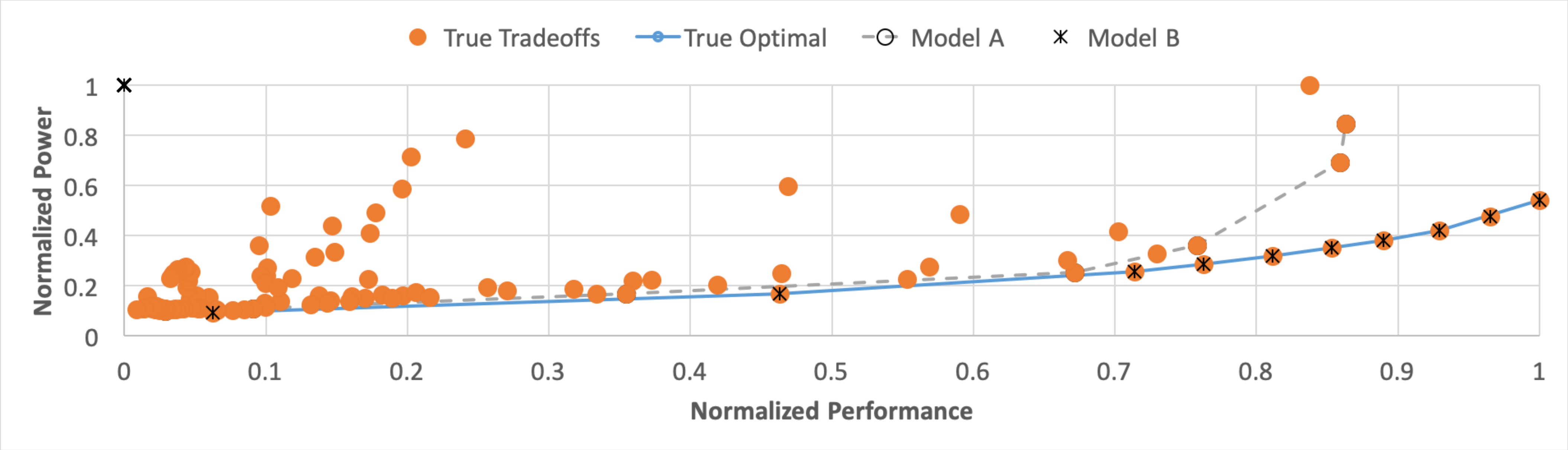
Config: an allocation of hardware resources to an application

Require the power and performance profile for applications

Expensive

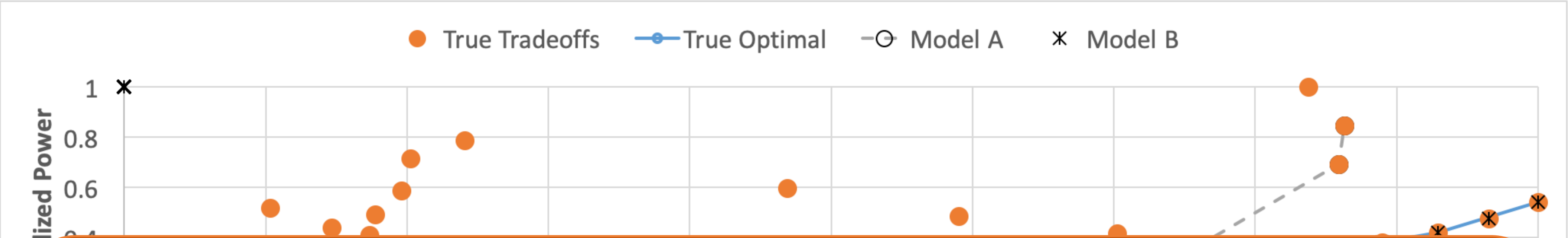
Learn to estimate these values

SRAD on ARM 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 ✅

SRAD on ARM big.LITTLE system



Key Insight:
High accuracy ≠ good system results

Optimal points	Just far enough	True data
Non-optimal points	True data	Just far enough
Goodness of fit	99%	0
Energy over optimal	22% ❌	0 ✅

Recommender Systems -> Learning by Examples

https://www.muvi.com/blogs/deciphering-the-unstoppable-netflix-and-the-role-of-big-data.html
https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf

1. Paragon: QoS-Aware Scheduling for Heterogeneous Datacenters. Christina Delimitrou and Christos Kozyrakis. (ASPLOS 2013)
2. Quasar: Resource-Efficient and QoS-Aware Cluster Management. Christina Delimitrou and Christos Kozyrakis (ASPLOS 2014)

COVER FEATURE

MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS

Yehuda Koren, *Yahoo Research*
Robert Bell and Chris Volinsky, *AT&T Labs—Research*

As the Netflix Prize competition has demonstrated, matrix factorization models are superior to classic nearest-neighbor techniques for producing product recommendations, allowing the incorporation of additional information such as implicit feedback, temporal effects, and confidence levels.

Modern consumers are inundated with choices. Electronic retailers and content providers offer a huge selection of products, with unprecedented opportunities to meet a variety of special needs and tastes. Matching consumers with the most appropriate products is key to enhancing user satisfaction and loyalty. Therefore, more retailers have become interested in recommender systems, which analyze patterns of user interest in products to provide personalized recommendations that suit a user's taste. Because good personalized recommendations can add another dimension to the user experience, e-commerce leaders like Amazon.com and Netflix have made recommender systems a salient part of their websites.

Such systems are particularly useful for entertainment products such as movies, music, and TV shows. Many customers will view the same movie, and each customer is likely to view numerous different movies. Customers have proven willing to indicate their level of satisfaction with particular movies, so a huge volume of data is available about which movies appeal to which customers. Companies can analyze this data to recommend movies to particular customers.

RECOMMENDER SYSTEM STRATEGIES

Broadly speaking, recommender systems are based on one of two strategies. The *content filtering* approach creates a profile for each user or product to characterize its nature. For example, a movie profile could include attributes regarding its genre, the participating actors, its box office popularity, and so forth. User profiles might include demographic information or answers provided on a suitable questionnaire. The profiles allow programs to associate users with matching products. Of course, content-based strategies require gathering external information that might not be available or easy to collect.

A known successful realization of content filtering is the Music Genome Project, which is used for the Internet radio service Pandora.com. A trained music analyst scores

Published by the IEEE Computer Society
0018-9162/09/$26.00 © 2009 IEEE

An Analogy

Users

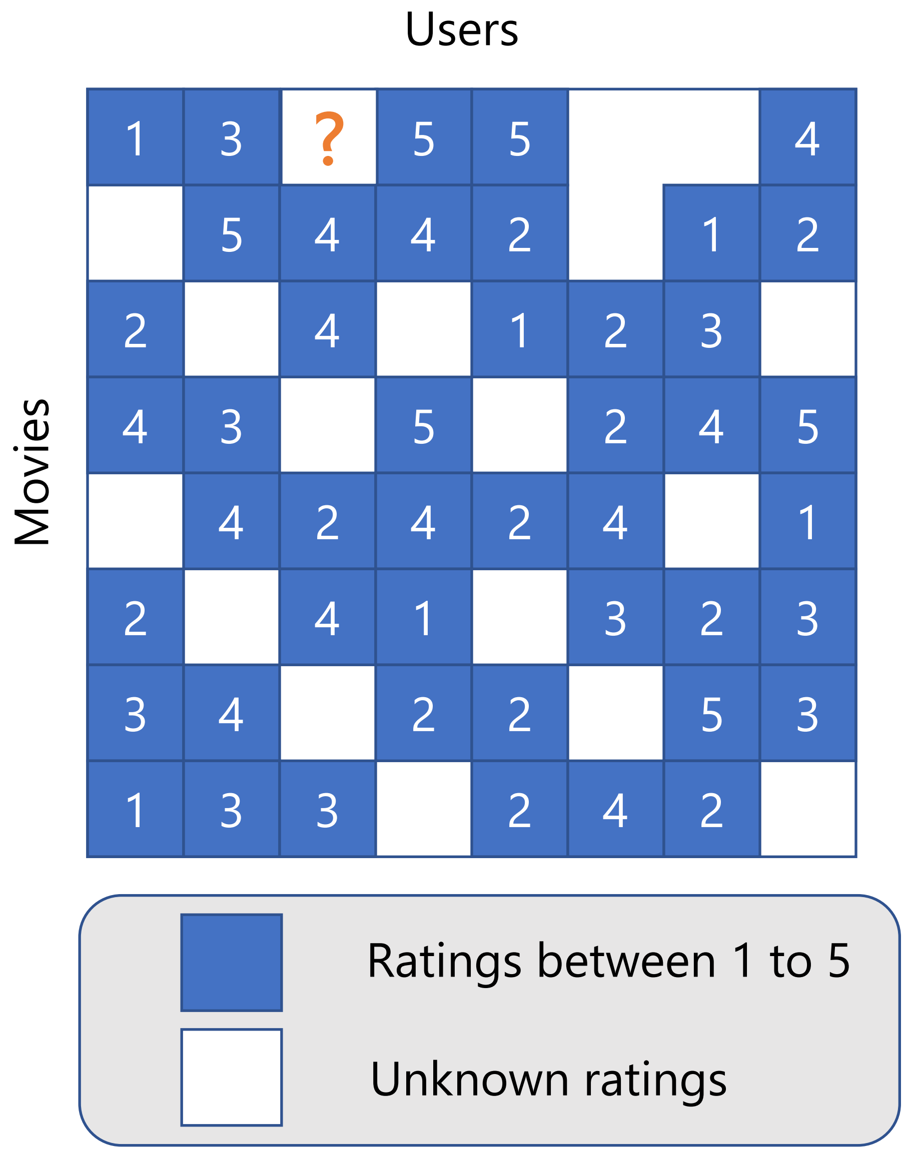
Movies

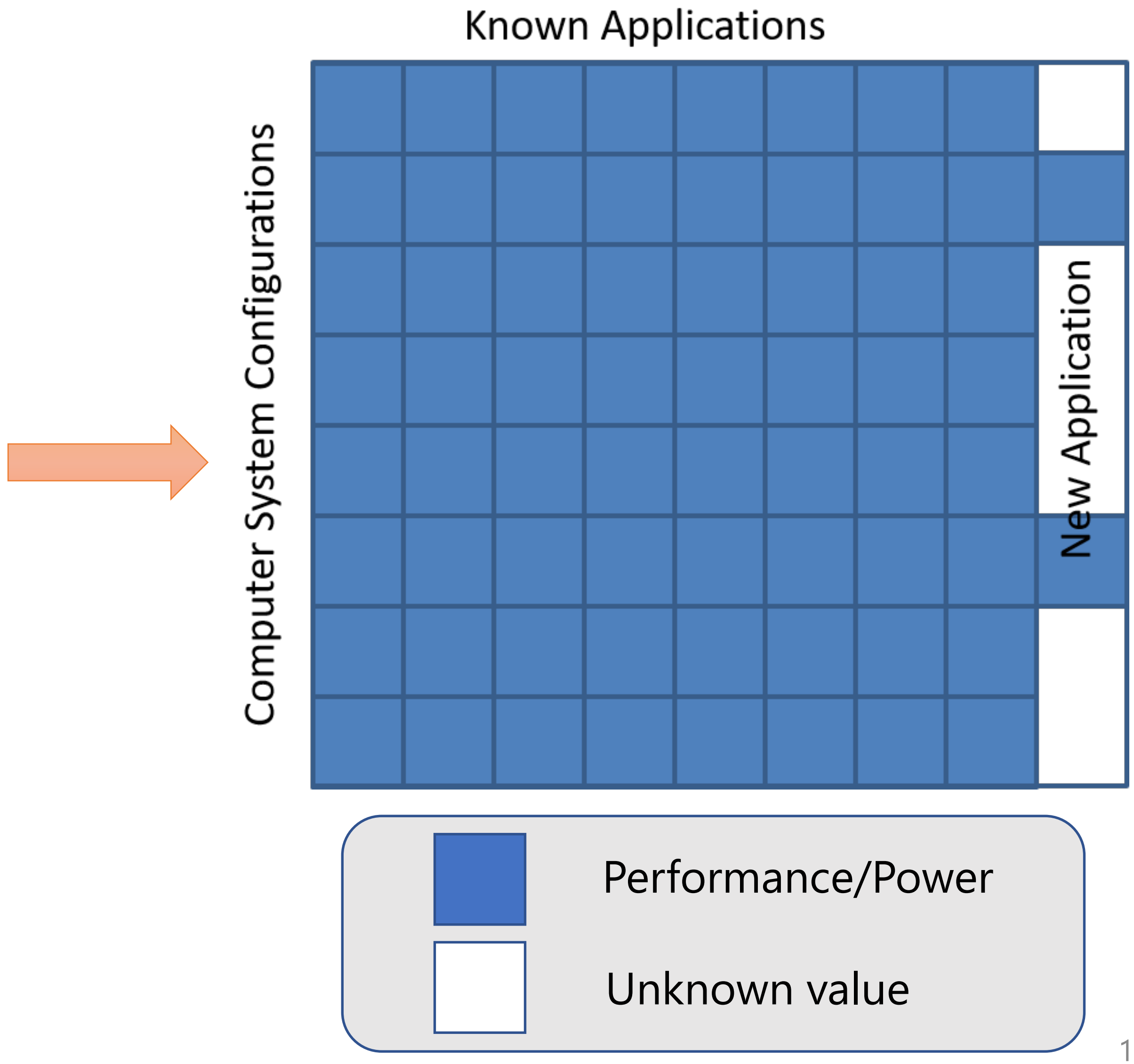
1	3	?	5	5			4
	5	4	4	2		1	2
2		4		1	2	3	
4	3		5		2	4	5
	4	2	4	2	4		1
2		4	1		3	2	3
3	4		2	2		5	3
1	3	3		2	4	2	

Ratings between 1 to 5

Unknown ratings

An Analogy


Users
Movies
1 3 ? 5 5 4
5 4 4 2 1 2
2 4 1 2 3
4 3 5 2 4 5
4 2 4 2 4 1
2 4 1 3 2 3
3 4 2 2 5 3
1 3 3 2 4 2
Ratings between 1 to 5
Unknown ratings


Known Applications
Computer System Configurations
New Application
Performance/Power
Unknown value

Outline

- Motivation
- Methods
- Experimental Results
- Conclusion

Generating Data for Accuracy

- Goal: *different* enough but still *realistic* to be plausible

Generating Data for Accuracy

- Goal: *different* enough but still *realistic* to be plausible
- How:

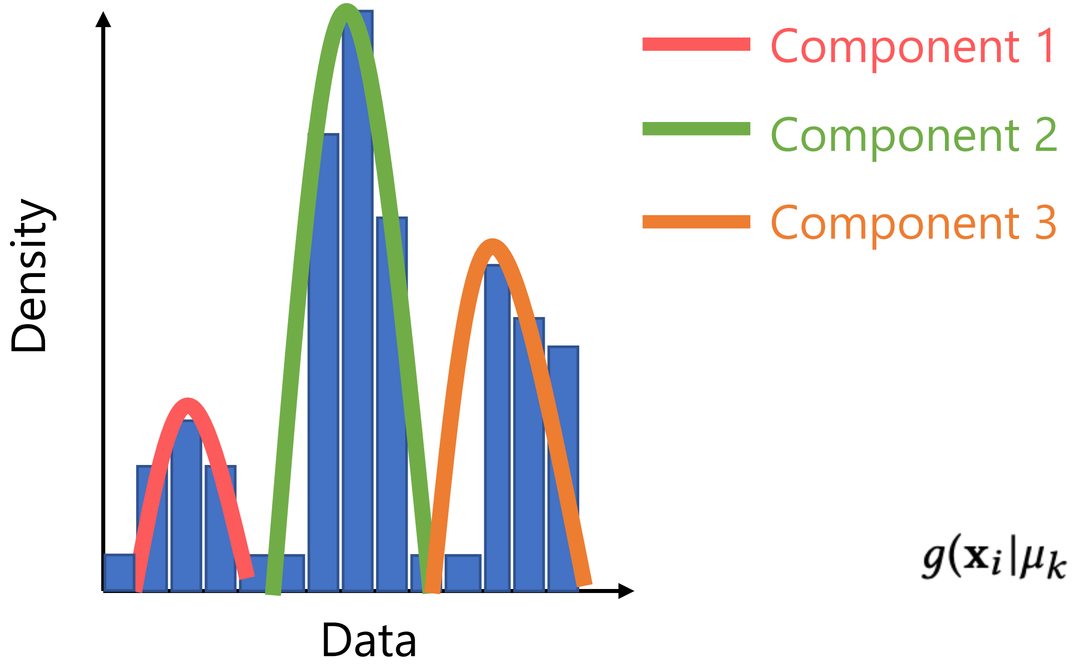
  Random number generator →different but not plausible

Generating Data for Accuracy

- Goal: *different* enough but still *realistic* to be plausible
- How:

  Random number generator →different but not plausible

  Gaussian Mixture Model (GMM) →plausible but not different



K: number of components
x_i : data points, i=1,…,N
w_k: weight of k-th component

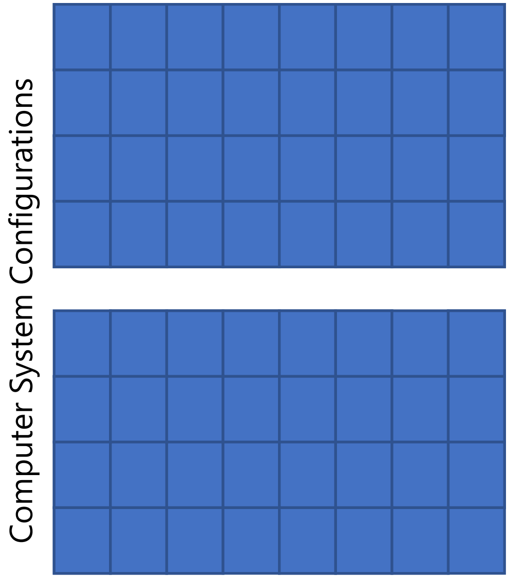
Probability that x_i belongs to k-th comp:

$$p(\mathbf{x}_i) = \sum_{k=1}^{K} w_k g(\mathbf{x}_i | \mu_k, \Sigma_k)$$

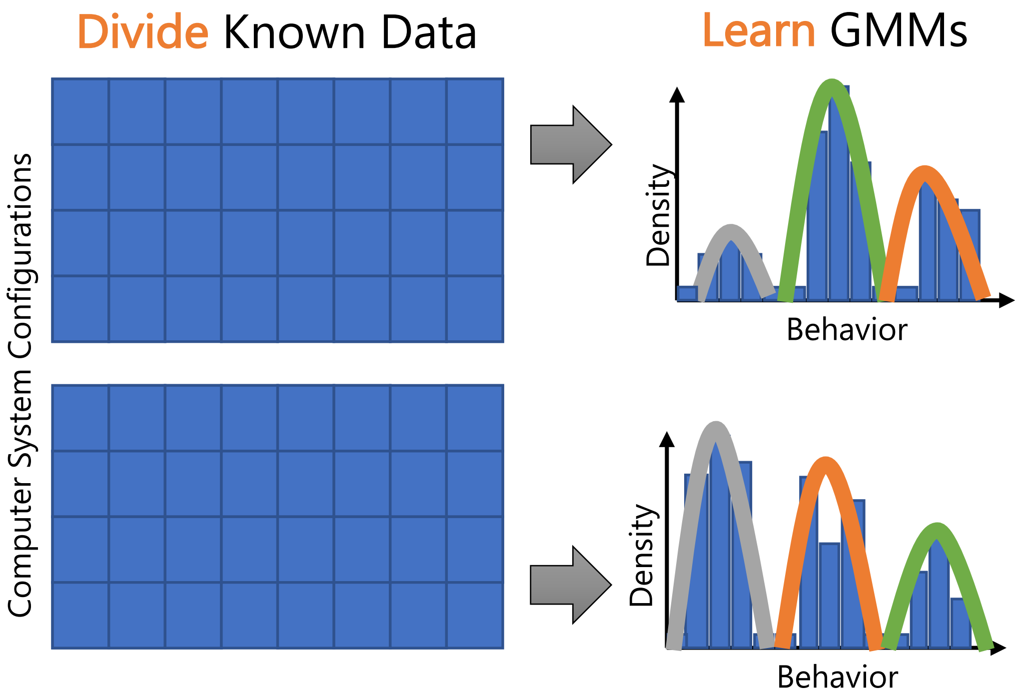
$$g(\mathbf{x}_i | \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} (\mathbf{x}_i - \mu_k)^\top \Sigma_i^{-1} (\mathbf{x}_i - \mu_k) \right)$$

Generating Data with a GMM

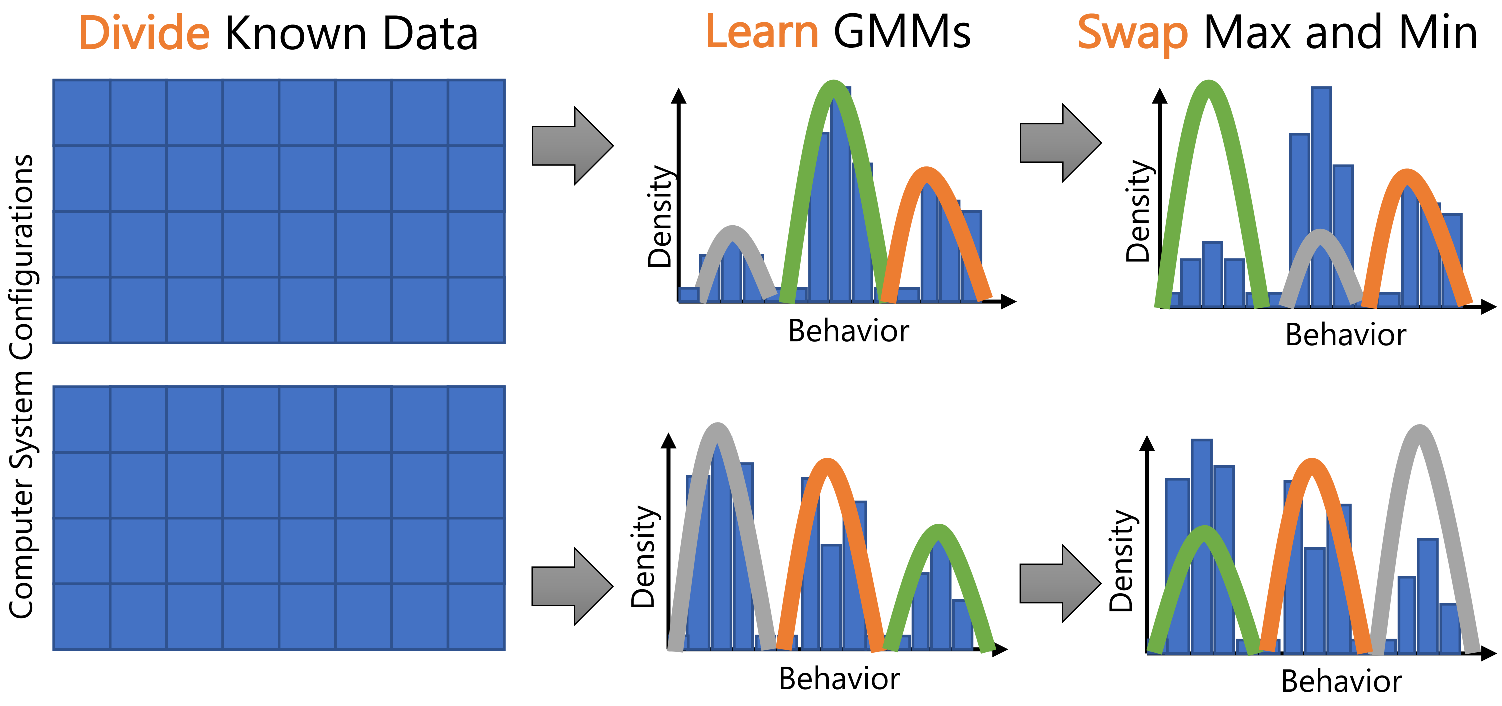
Divide Known Data



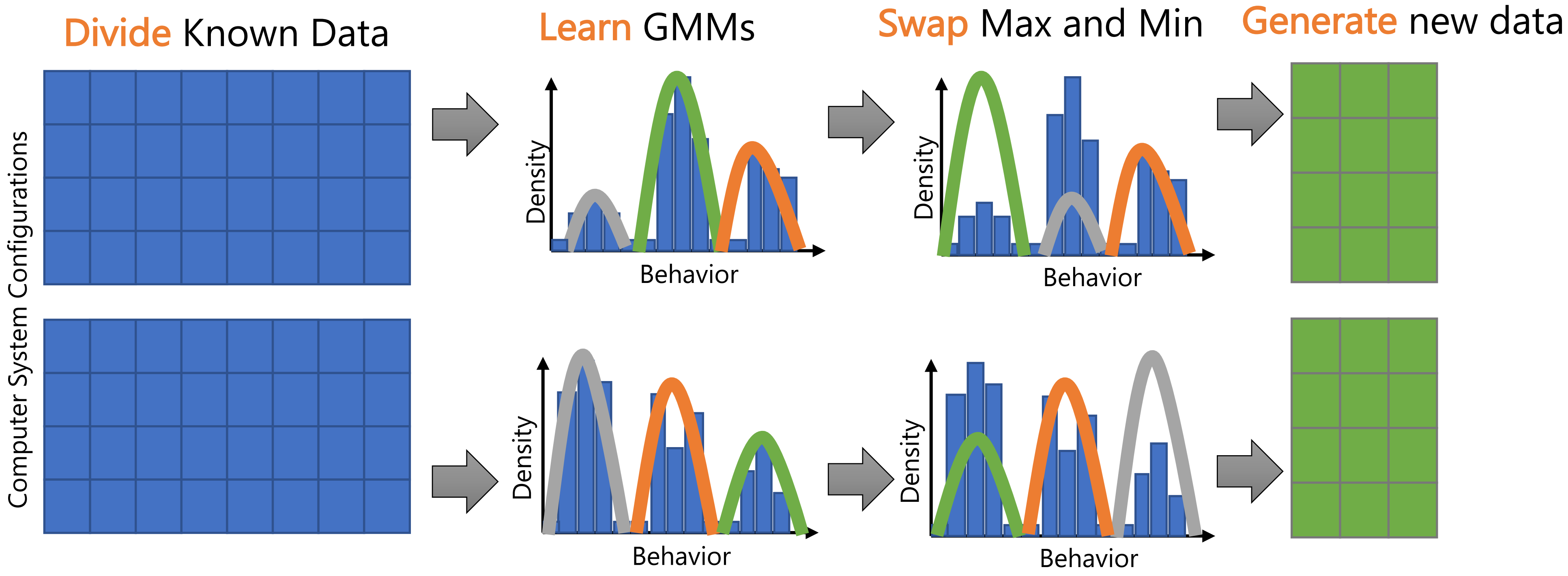
Generating Data with a GMM



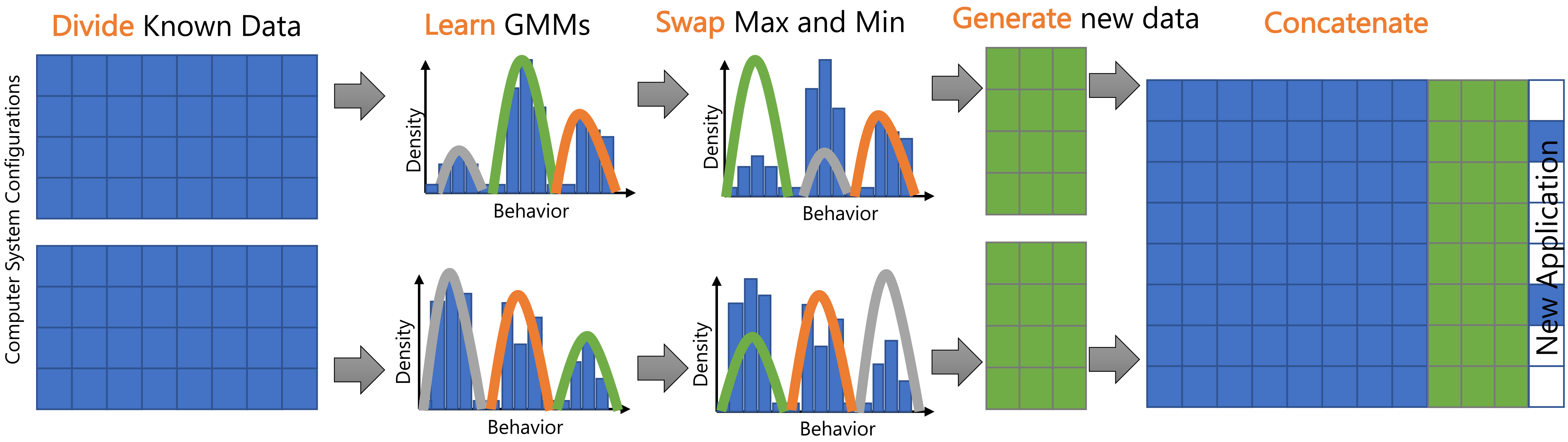
Generating Data with a GMM



Generating Data with a GMM



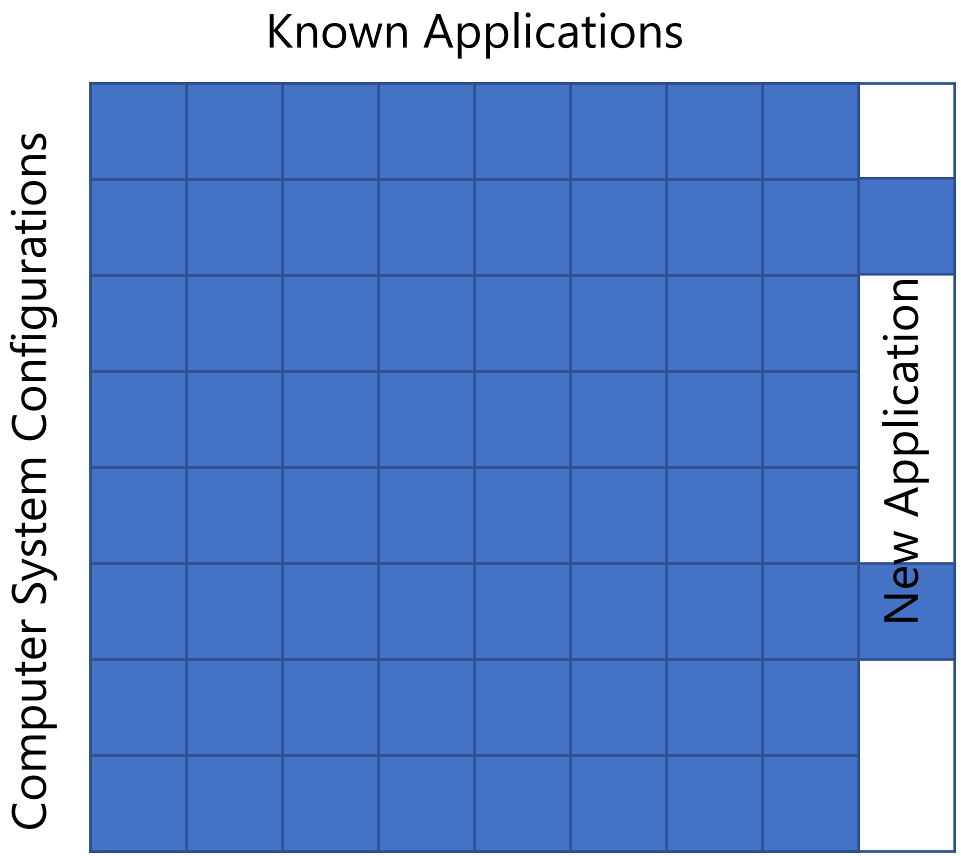
Generating Data with a GMM



Multi-phase Sampling

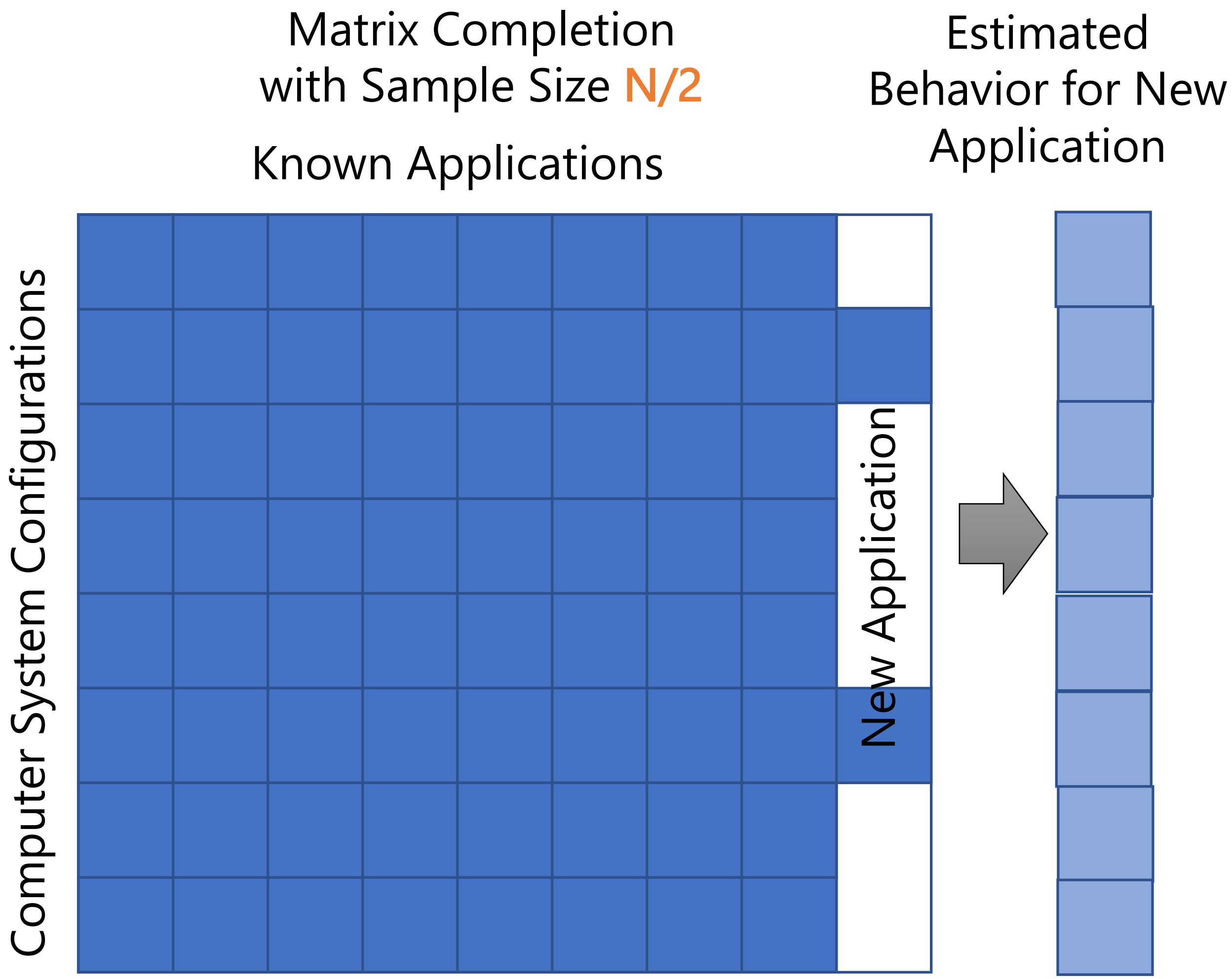
Input: Configuration-Application data matrix, Sampling budget N

Matrix Completion
with Sample Size N/2



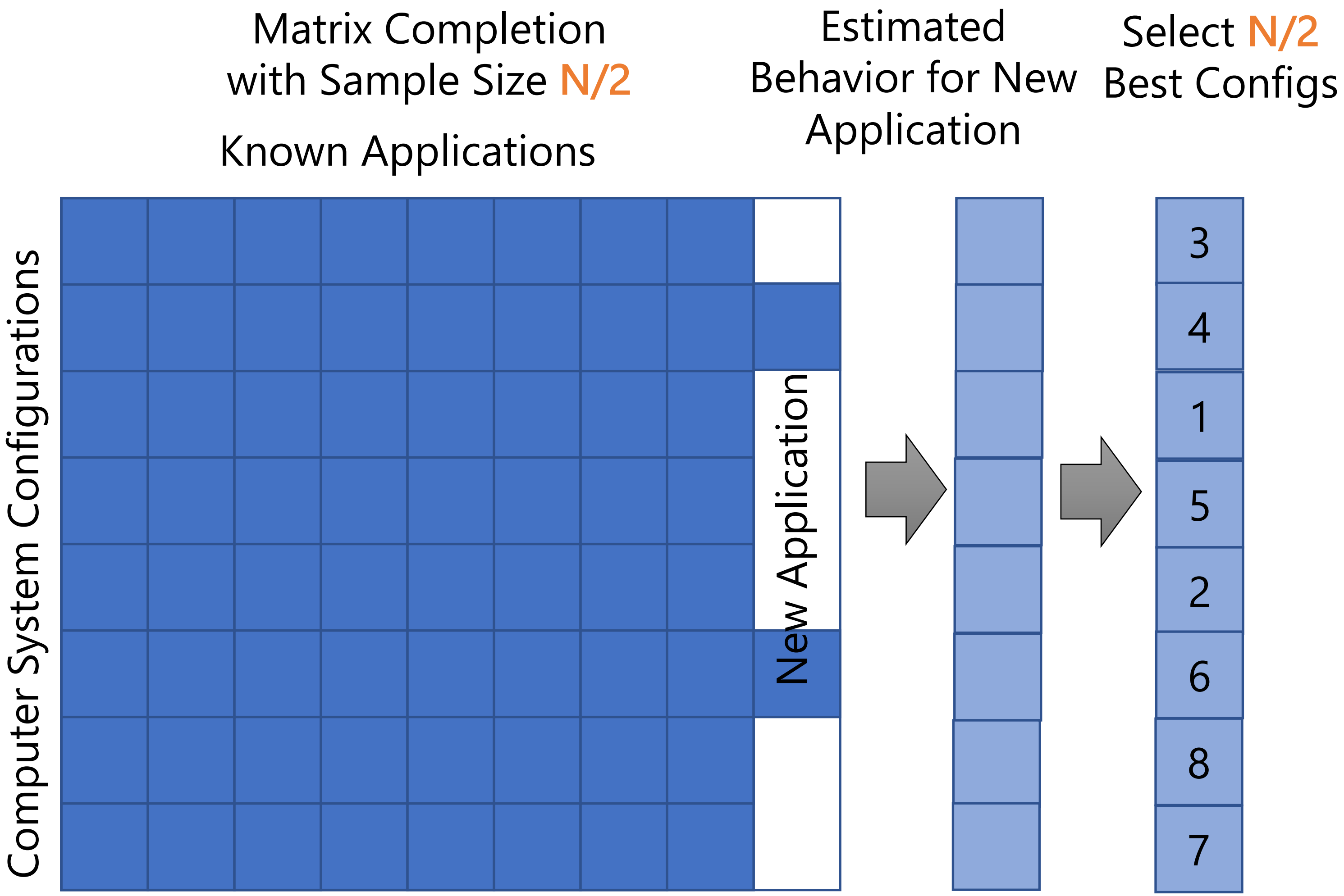
Multi-phase Sampling

Input: Configuration-Application data matrix, Sampling budget N



Multi-phase Sampling

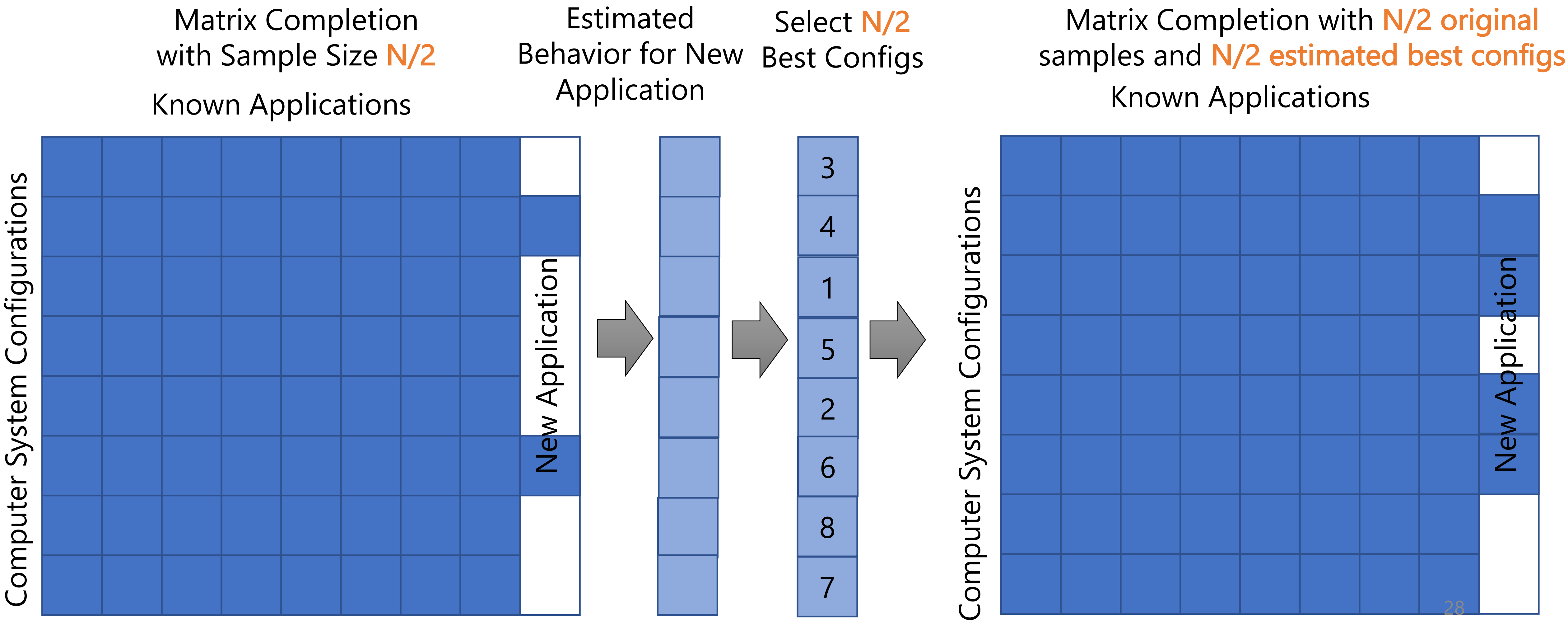
Input: Configuration-Application data matrix, Sampling budget N



$$\text{efficiency} = \frac{\text{estimated performance}}{\text{estimated power}}$$

Multi-phase Sampling

Input: Configuration-Application data matrix, Sampling budget N

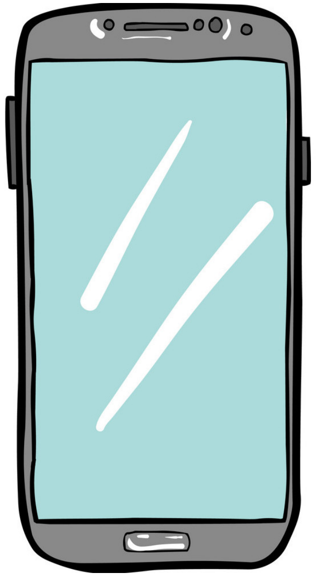


Outline

- Motivation
- Methods
- Experimental Results
- Conclusion

Experimental Setup

	Mobile	Server
System	Ubuntu 14.04	Linux 3.2.0 system
Architecture	ARM big.LITTLE	Intel Xeon E5-2690
# Applications	21	22
# Configurations	128	1024



Learning Models and Frameworks

Learning Models	Category
MCGD	MC
MCMF	MC
Nuclear	MC
WNNM	MC
HBM	Bayesian

First comprehensive study of matrix completion (MC) algorithms for systems optimization task

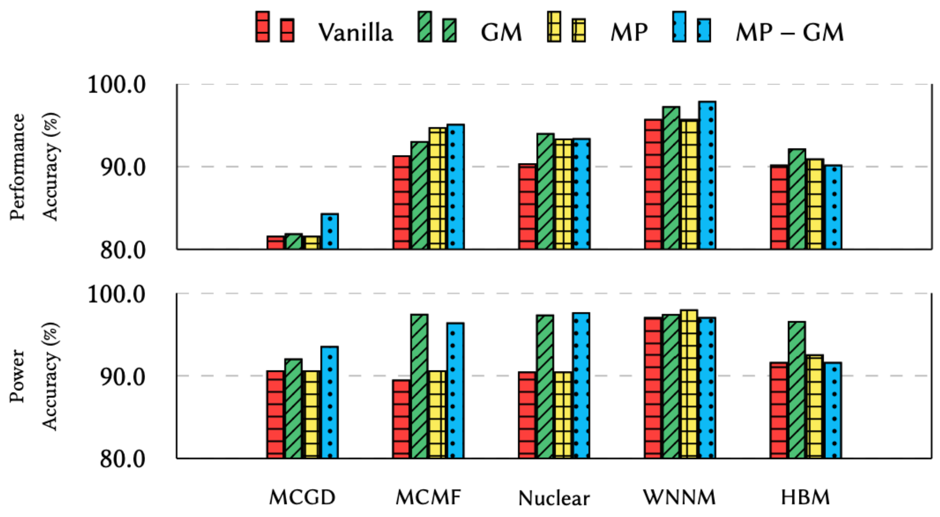
Learning Models and Frameworks

Learning Models	Category
MCGD	MC
MCMF	MC
Nuclear	MC
WNNM	MC
HBM	Bayesian

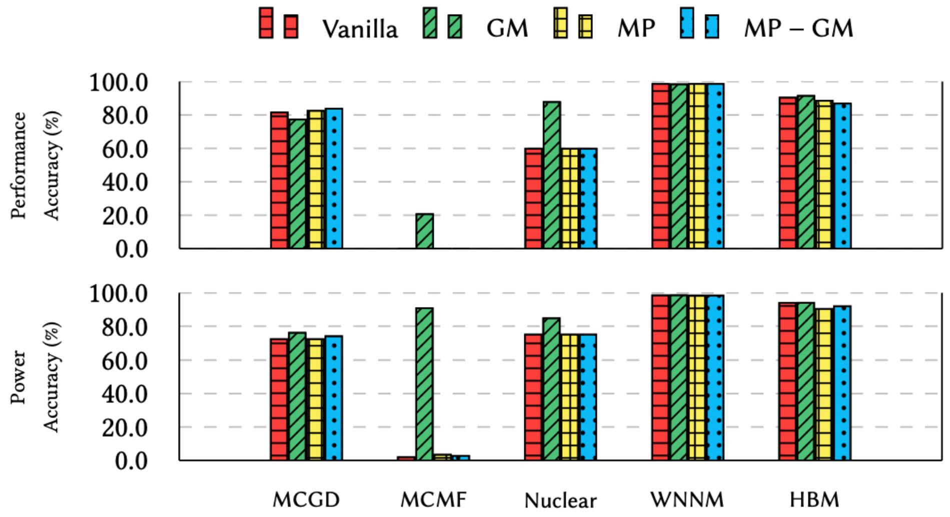
First comprehensive study of matrix completion (MC) algorithms for systems optimization task

Frameworks	Definitions
Vanilla	Basic learners
GM	Generative model
MP	Multi-phase sampling
MP-GM	Combine GM and MP

Improve Prediction Accuracy w/ GM



Mobile



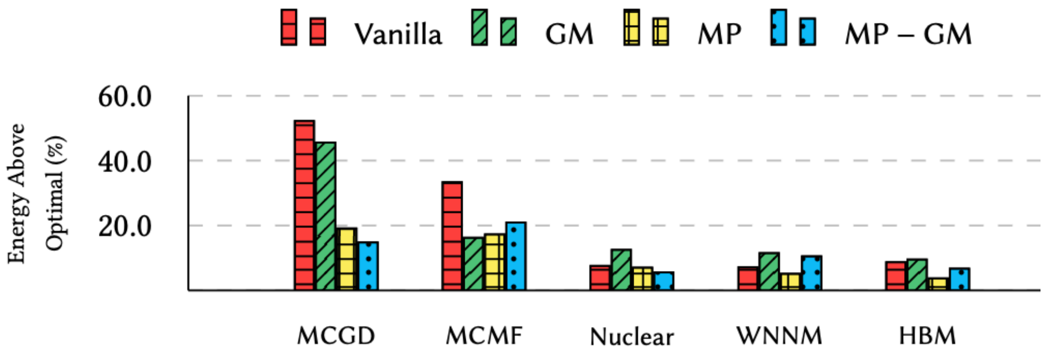
Server

High is Better

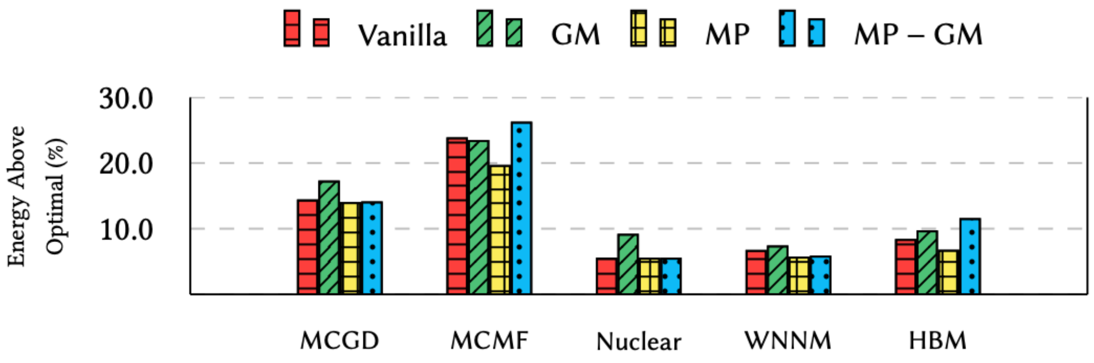
Average percentage points of accuracy improvement

		GM	MP	MP – GM
Mobile	Performance	1.8	1.4	2.3
	Power	4.3	0.6	3.4
Server	Performance	9.0	−0.2	−0.3
	Power	20.5	−0.4	0.1
Average		8.9	0.4	1.4

Improve Energy Savings w/ MP



Mobile



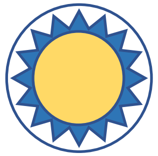
Server

Lower is Better

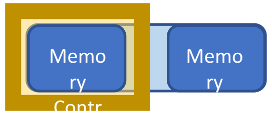
Average energy improvement

	GM	MP	MP – GM
Mobile	−14%	41%	22%
Server	−22%	11%	−6.5%

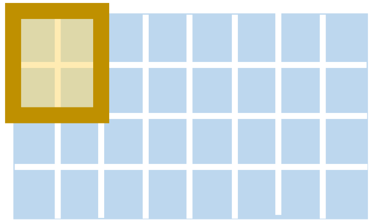
Conclusion



Clock Speed

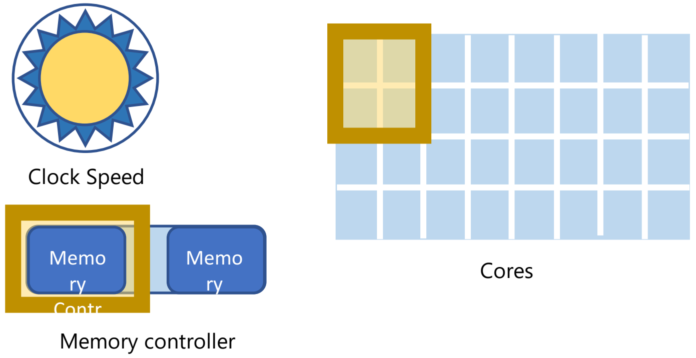


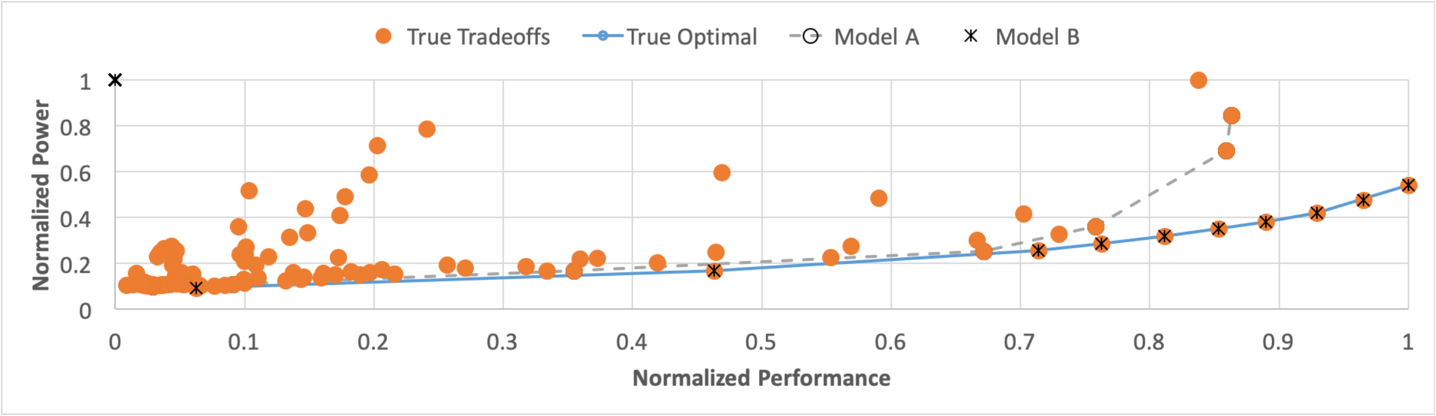
Memory controller



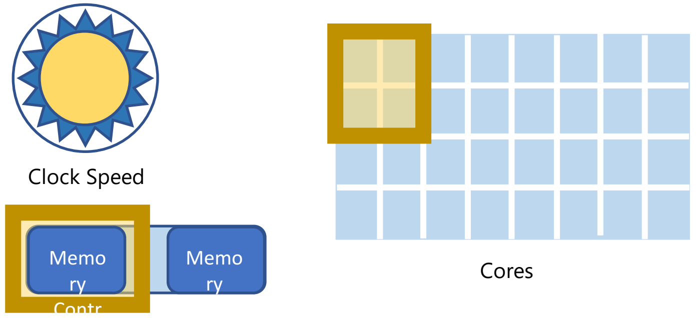
Cores

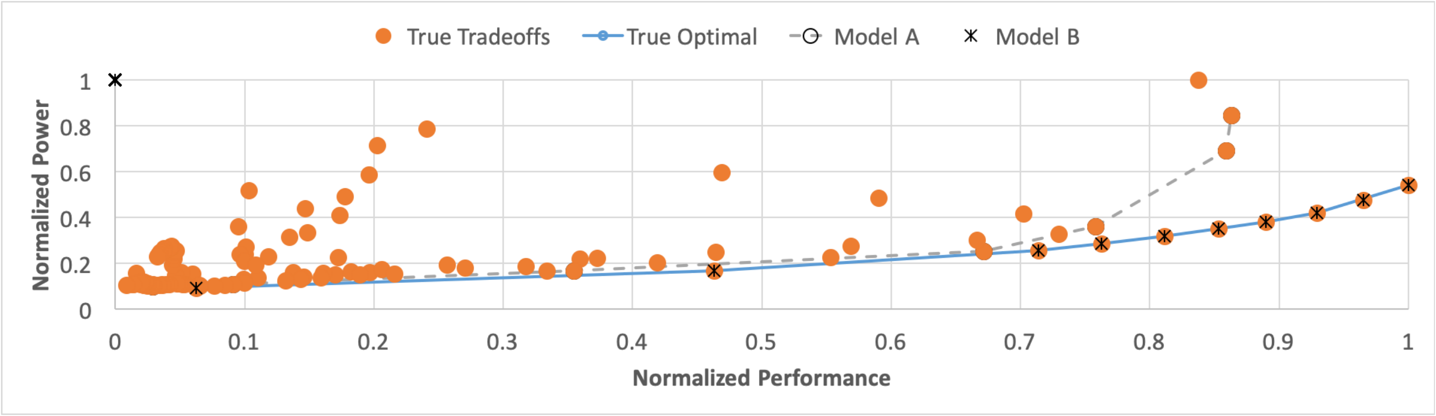
Conclusion

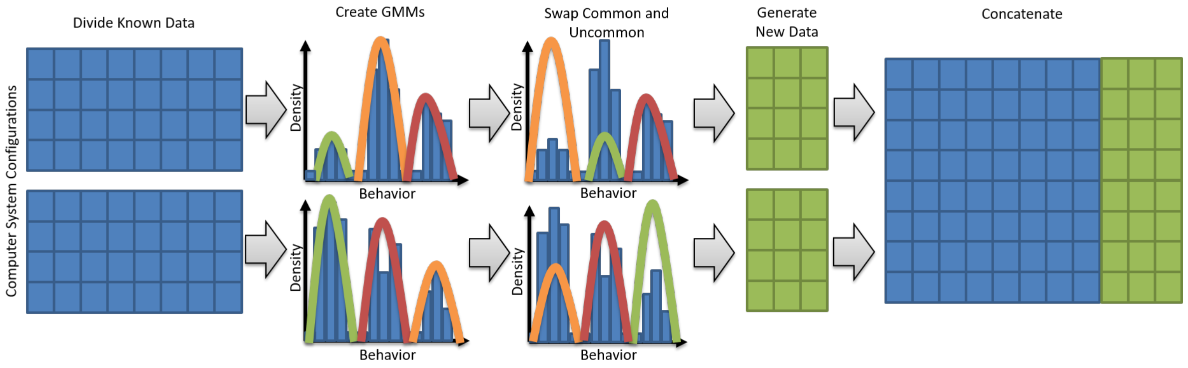

Clock Speed
Memo ry
Contr
Memo ry
Memory controller
Cores

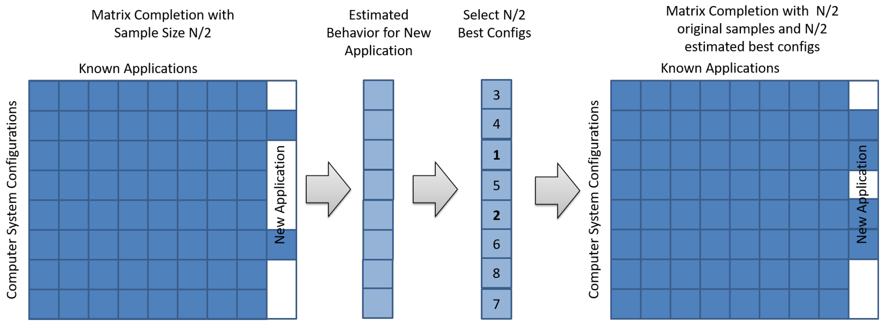

True Tradeoffs
True Optimal
Model A
Model B
Normalized Power
1
0.8
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Normalized Performance

Conclusion

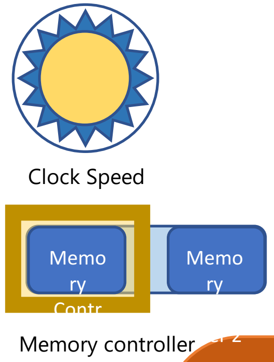

Clock Speed
Memo ry
Memo ry
Memory controller
Cores

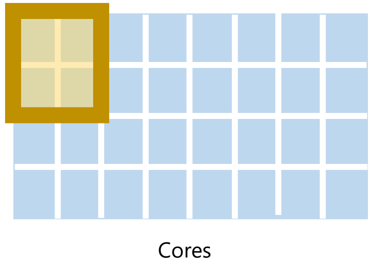

True Tradeoffs
True Optimal
Model A
Model B
Normalized Power
Normalized Performance
0
0.2
0.4
0.6
0.8
1
0.1
0.3
0.5
0.7
0.9

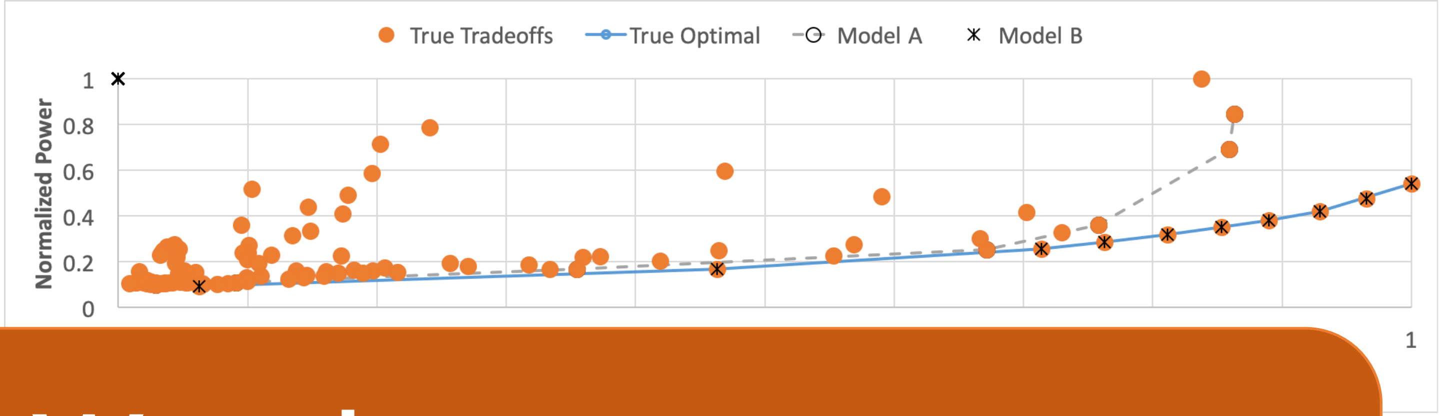

Divide Known Data
Create GMMs
Swap Common and Uncommon
Generate New Data
Concatenate
Computer System Configurations
Density
Behavior

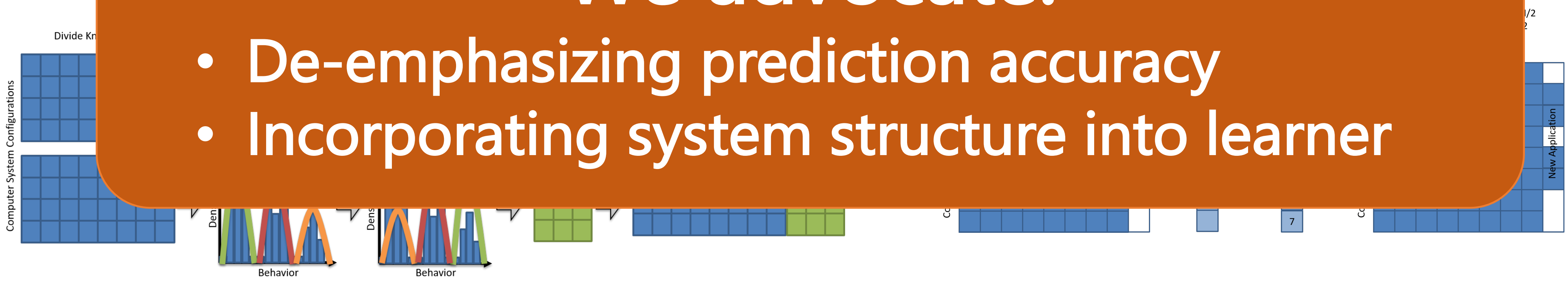

Matrix Completion with Sample Size N/2
Estimated Behavior for New Application
Select N/2 Best Configs
Matrix Completion with N/2 original samples and N/2 estimated best configs
Known Applications
Computer System Configurations
New Application
3
4
1
5
2
6
8
7

Conclusion









We advocate:

- De-emphasizing prediction accuracy
- Incorporating system structure into learner

Yi Ding, Nikita Mishra, and Henry Hoffmann. 2019. Generative and Multiphase Learning for Computer Systems Optimization. In The 46th Annual International Symposium on Computer Architecture (ISCA '19)