ICS '23

Transfer-Learning-Based Autotuning Using Gaussian Copula

Thomas Randall,
Jaehoon Koo,
PhD Candidate Associate
Professor
Clemson University Hanyang
University

Brice Videau, Michael Kruse, Xingfu Wu, Paul Hovland, and Mary Hall,
Argonne National Laboratory; University of Utah

Rong Ge,
Clemson University

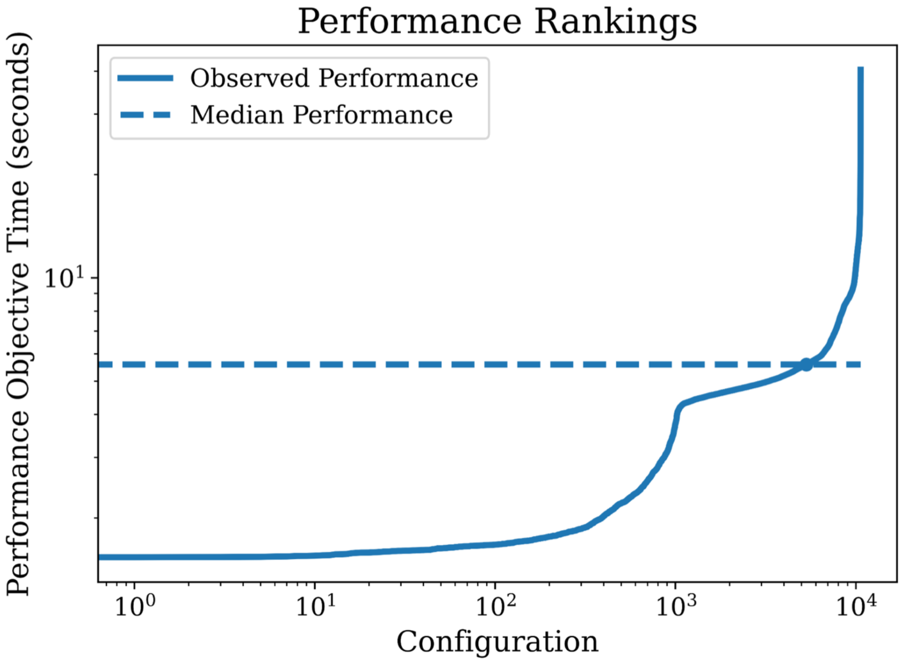
Prasanna Balaprakash
Oak Ridge National Laboratory

Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.

https://github.com/tlranda/GC_TLA

Performance Autotuning: Necessary but Costly

- Empirical tuning and optimization
  - Large space
  - *Sophisticated* search
- Tuning is perpetually necessary
  - New systems: Aurora
  - New applications: Exascale Computing Project
- Empirical testing is **costly**
  - Efficiency is key!



Performance autotuning navigates very large search spaces and identifies high-performing configurations, ie: top-100 of 10,000

Even Simple Kernels Are Expensive!

- Simple matmul kernel:
  - (A×B)×(C×D)
  - Ten tunable Polly parameters
    - *376,320* configurations
  - <25 seconds per evaluation

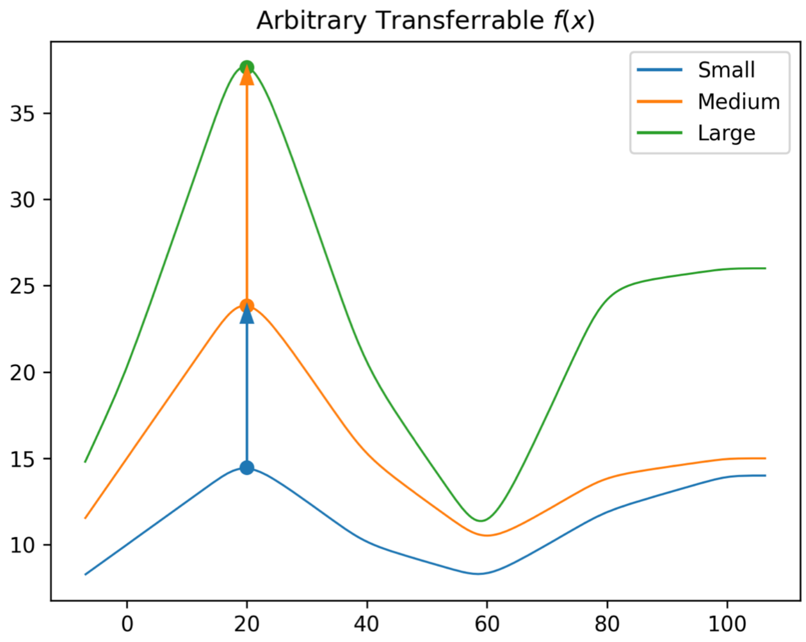
Parameter	Values
Tile Sizes	[4-2048], [4-2048], [4-2048]
Loop Interchange	[Yes, N/A]
Array Packing	[Yes, N/A] × 6

- **100+ days** tuning to try each configuration *once*!
- Same kernel, different input sizes:
  - ***Different*** optimum configurations

	Input Scale		
	Small	Medium	Large
Packed Arrays	A,E,F	F	A,B,E
Loop Interchanges	N/A	N/A	Outer Exchange
Tile Sizes	16, 2048, 4	96, 16, 4	4, 2048, 4

Transfer Learning (TL) Autotuning: Few-Shot

- Reuse knowledge in related tasks
  - Limit tuning costs
- Gain knowledge from “cheap” tasks
  - Near-optimal configurations
  - Poor configurations
- Reuse it on “expensive” tasks to maximize efficiency
  - Enable few-shot
  - Converge to high performance



Existing Searches and Autotuners

- Model-Free Techniques
  - Simple to define
  - Minimal convergence guarantees, if any
- Model-Based Techniques
  - Sophisticated definition and capabilities
  - Long-term convergence usually guaranteed
    - Short-term results often lackluster
    - Restarting from scratch is **EXPENSIVE**
- Primary gap:
  - *Aggressive, transferrable* model-based search that is *simple* to define

Existing TL Shortcomings

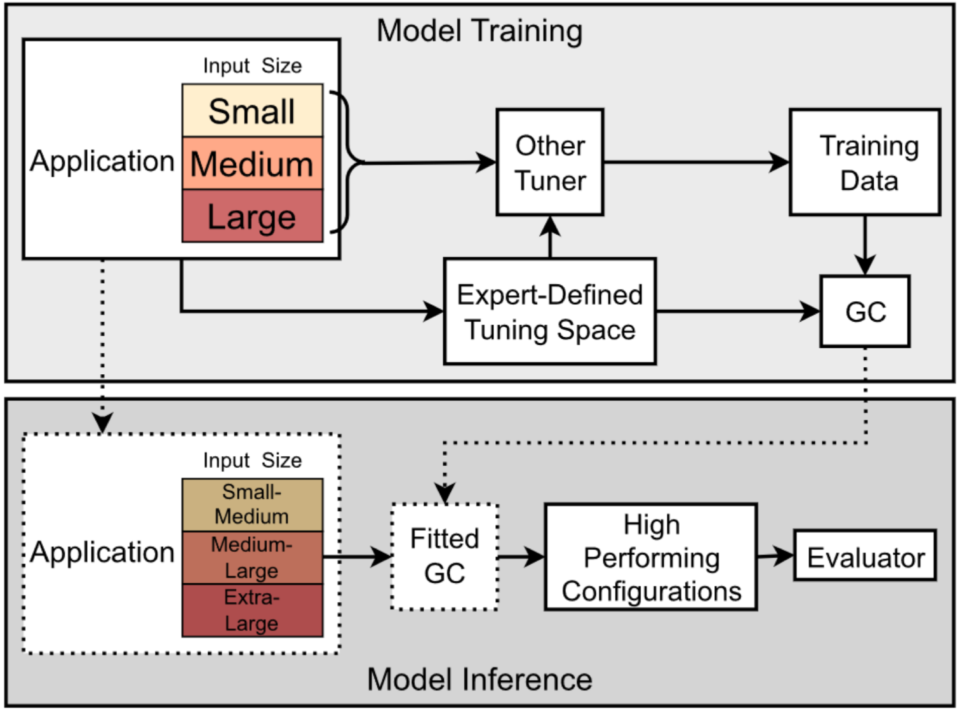
- No obvious model-free transfer technique
  - Generally TL complicates definitions
  - Would be great to have a simpler definition for TL
- Model-based regression *requires* ground truth
  - Expensive restart *NOT* completely avoided
  - Ideally, TL permits greater shortcuts
- Machine-learning scales to **BIG DATA**
  - Desirable to work with *minimal* source data
  - Long-term convergence is *too slow*
  - Better than restarting from scratch, but we can do even better!

Gaussian Copula (GC) TL-Based Autotuning

- Maximize few-shot performance for new input sizes
  - Common tuning setting for HPC
- Simple model capable of transfer without regression
  - Reduce need for ground truth
    - Scale *down* to minimal data
    - Immediate performance on new scales
  - Provide probability estimate of viability
    - Budgeting with *zero evaluations*

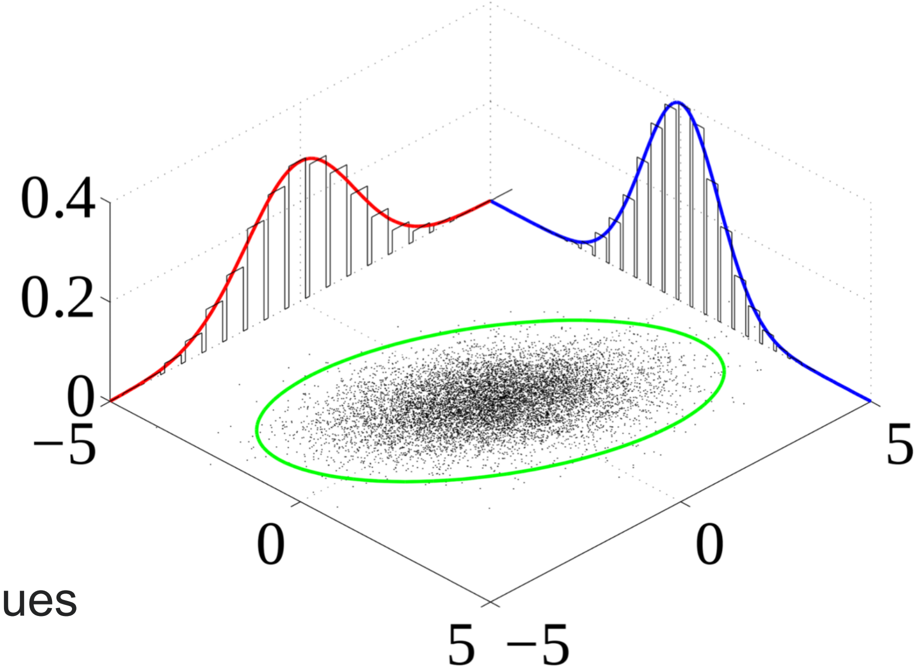
GC Few-Shot TL Autotuning

- Fit to tuning space definition and prior data from various input sizes
  - Prompt with new input size
  - Generate candidate configurations to evaluate
- Demonstrate with real benchmarks
  - *FIRST* evaluation: **64% peak** few-shot speedup
  - **12.81× higher peak** speedup (20.58→33.39×) vs previous SOTA



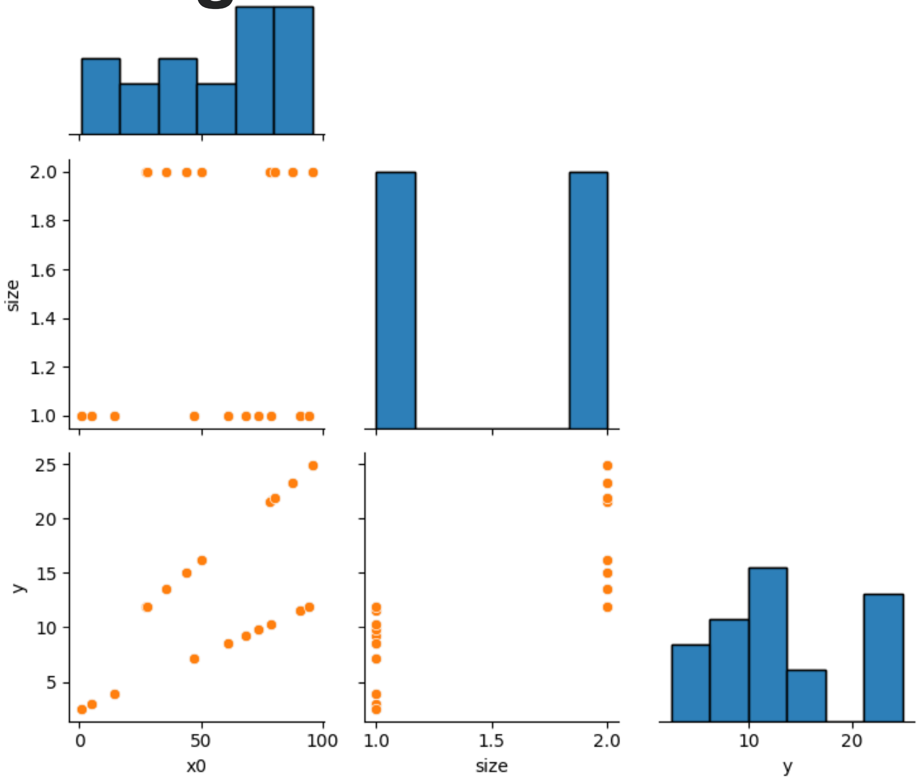
GC Model

- Multivariate probability distribution
- Components
  - Disjoint marginal per variable
  - Correlations as joint distribution
- Capabilities
  - Probability integral transform
    - Samples ↔ Distributions
  - Conditional sampling
    - Prescribe some marginal values
    - Adjust remaining variance



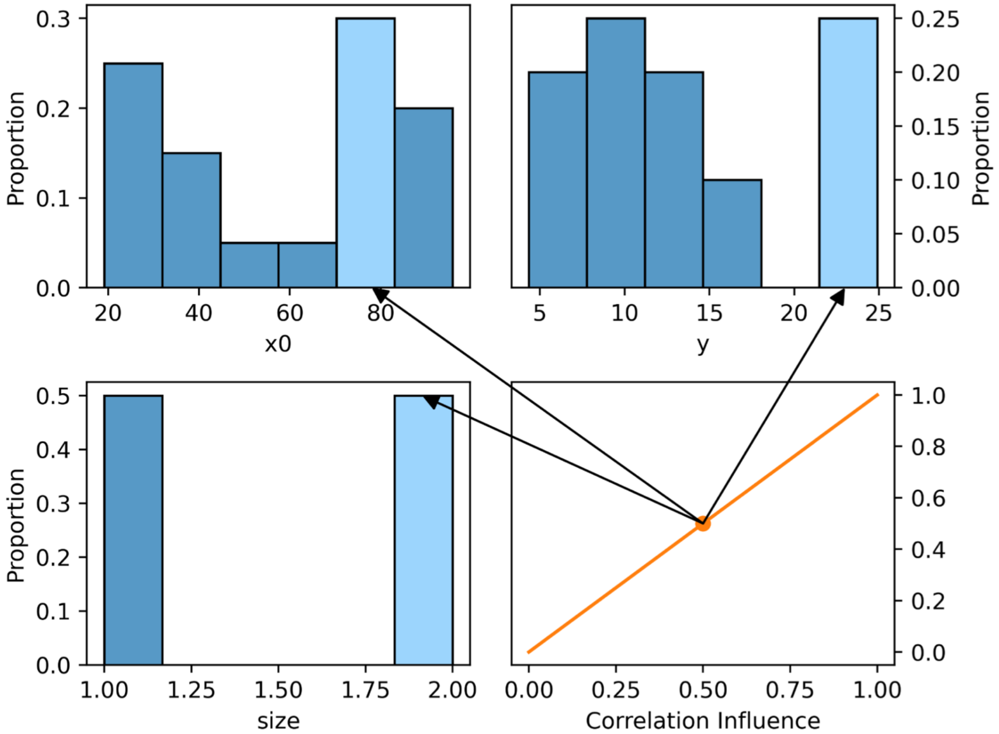
Toy Generative Transfer Tuning Problem

- Variables: x0, size, y
  - All linear relations



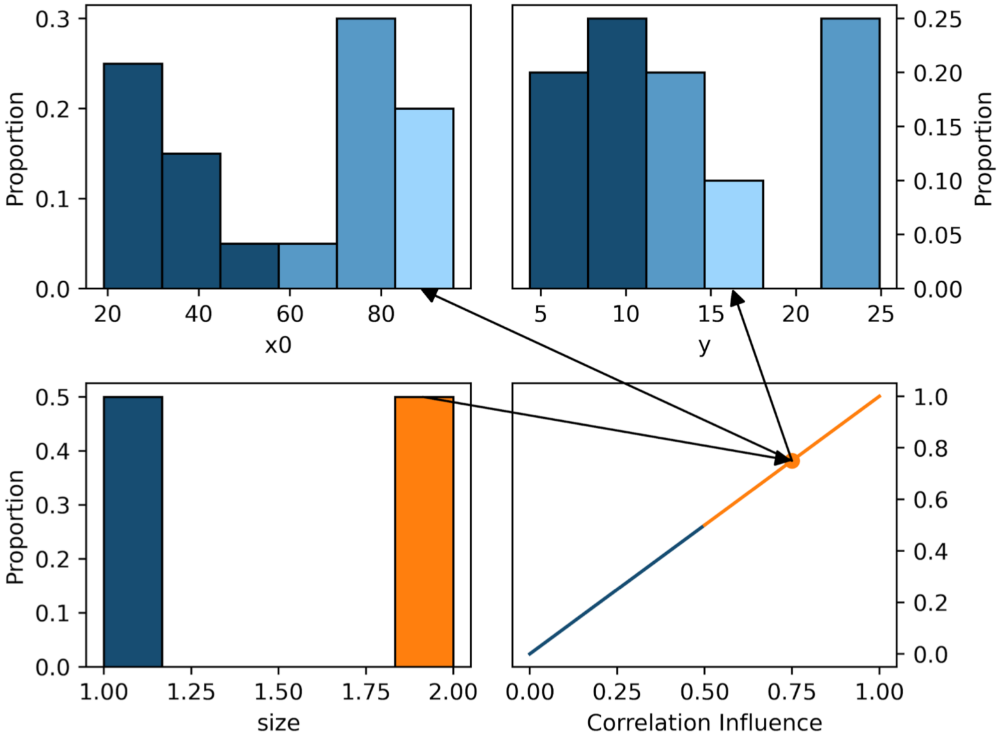
Toy Generative Transfer Tuning Problem

- Variables: x0, size, y
  - All linear relations
- Sample from distribution
  - Resemble original samples



Toy Generative Transfer Tuning Problem

- Variables: x0, size, y
  - All linear relations
- Sample from distribution
  - Resemble original samples
- Conditionally sample for specific behaviors
  - Limit expression to relevant subset



Using Distributions As Search

- GC lacks regression
  - No comparisons/ranking
  - *Minimal* data describes a distribution
- Provide search boundaries
  - Under-represented = Poor traits
  - Over-represented = Solved traits
  - Variance = Opportunity to explore
- What makes a good distribution?
- How do we use it?

“Good” Distribution from Filtered Data

- Needs limited coverage of tuning space
  - # generable / total space size
  - *Reduce*, but do not *eliminate*
- Needs specificity to match optimal area
  - KL Divergence compares probability distributions (distance metric)
  - Compare:
    - Brute-force top-10% configs
    - Filtered top-X% source data
  - *Lower* divergence = better match

Filtering Quantile (%)	Tuning Space Coverage	KL Divergence
100	1.00	0.1878
90	1.00	0.1713
80	1.00	0.1609
70	1.00	0.1525
60	0.91	0.1409
50	0.91	0.1212
40	0.91	0.1333
30	0.82	0.1713
20	0.07	0.2766
10	0.06	0.3079

Filtering: Out with the Bad

- Filter source data via observed quantiles
  - Remove poor features: < top-50%

Filtering Quantile (%)	Tuning Space Coverage	KL Divergence
100	1.00	0.1878
90	1.00	0.1713
80	1.00	0.1609
70	1.00	0.1525
60	0.91	0.1409
50	0.91	0.1212
40	0.91	0.1333
30	0.82	0.1713
20	0.07	0.2766
10	0.06	0.3079

Filtering: Preserve *Sufficient* Coverage

- Filter source data via observed quantiles
  - Remove poor features: < top-50%
- Careful! Do not filter too much!
  - Empirically require: > top-15%

Filtering Quantile (%)	Tuning Space Coverage	KL Divergence
100	1.00	0.1878
90	1.00	0.1713
80	1.00	0.1609
70	1.00	0.1525
60	0.91	0.1409
50	0.91	0.1212
40	0.91	0.1333
30	0.82	0.1713
20	0.07	0.2766
10	0.06	0.3079

Filtering: Empirical Ideal

- Filter source data via observed quantiles
  - Remove poor features: < top-50%
- Careful! Do not filter too much!
  - Empirically require: > top-15%
- Suggest: top-30%
  - Sufficient but minimized space coverage
  - Divergence not increasing too much

Filtering Quantile (%)	Tuning Space Coverage	KL Divergence
100	1.00	0.1878
90	1.00	0.1713
80	1.00	0.1609
70	1.00	0.1525
60	0.91	0.1409
50	0.91	0.1212
40	0.91	0.1333
30	0.82	0.1713
20	0.07	0.2766
10	0.06	0.3079

Conditional Sampling as Transfer Mechanism

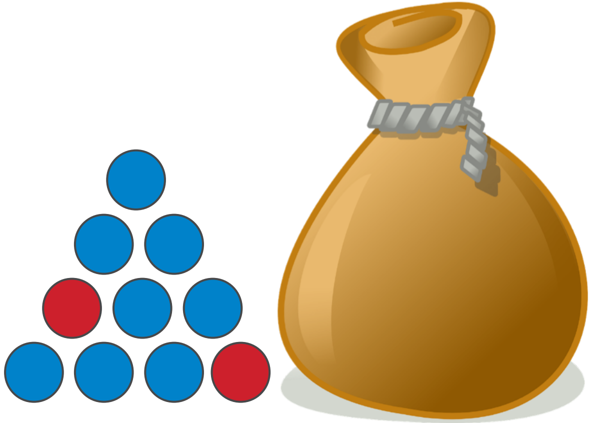
- Different scales require different solutions
  - General sampling does not respect input scale
- Add input scale feature representation (arbitrary marginal variable)
  - Inference uses conditional sampling for the target scale
- Conditioning reconstructs a scale-specific sub-distribution
  - Marginal distributions adjusted alongside correlations
  - All data utilized, dynamically transferred

U.S. DEPARTMENT OF ENERGY
Argonne National Laboratory is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC.

CLEMSON Argonne NATIONAL LABORATORY

Budget Estimation: Probability of Success

- Hypergeometric sampling (blind marble picking):
  - $|\mathrm{C}|$ configurations (marbles)
    - $|\mathrm{I}|$ near-optimal (red marbles)
  - Up to k samples
- Incomplete coverage from GC
  - Remove marbles before sampling!
- Probability estimation
  - Unique GC samples are proxy for $|\mathrm{C}|$
    - Estimate reduction in $|\mathrm{I}|$



$$P(\#Optimal \geq 1) = \sum_{i=1}^{k} \frac{\binom{|I|}{i}\binom{|C|-|I|}{k-i}}{\binom{|C|}{k}}$$

Experiment Design

- Evaluation Platform
  - 2✕ AMD EPYC 7742 (64-core; 128-logical)
  - 1✕ 40 GB NVIDIA A100
  - Clang with Polly LLVM loop optimizer
- Each application source sizes:
  - Bayesian Optimization with Random Forest
  - 200✕ each for Small, Medium, Large
- Each application target sizes:
  - 30✕ each for Small-Medium, Medium-Large, Extra-Large

Benchmark	#Params	# Configurations
3mm	10	376,320
Covariance	5	5,324
Floyd–Warshall	5	5,324
Heat3d	6	10,648
LU	5	5,324
Syr2k	6	10,648
AMG	9	1,180,980
RSBench	9	5,196,312
XSBench	8	577,368
SW4Lite	8	4,752

Compared Approaches

- Baseline
  - Parameters derived from original source
  - Reference for speedup
- Bayesian Optimization (BO)
  - From scratch without TL; same settings as training dataset
- All TL use the same prior dataset from BO
  - GPTune DTLA
    - **SOTA** TL autotuner using Gaussian Processes
  - GC-TLA (**ours**)
    - Fit to top-30% source data; conditionally sample for TL

Polybench: High Efficiency and Performance

- 3mm XL: **12.81×** more speedup than prior SOTA

App.	Scale	Peak Speedup (# Evaluation Discovered)				
		GC			BO	GPTune
		1^{st}	Budget	Best	Best	Best
3mm	SM	5.09	5.70 (23)	5.70 (23)	3.03 (26)	5.53 (30)
	ML	5.25	5.57 (29)	5.57 (29)	3.29 (30)	5.16 (16)
	XL	27.10	33.39 (18)	33.39 (18)	20.58 (30)	18.96 (25)

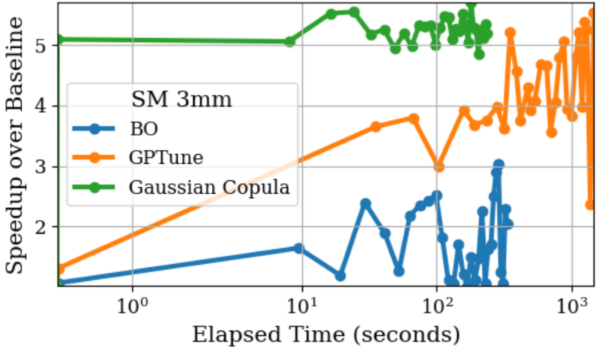
Polybench: High Efficiency and Performance

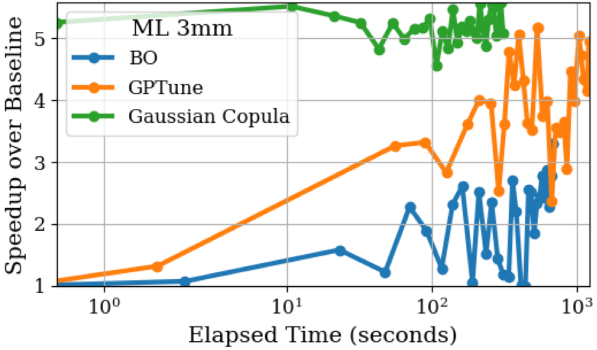
- 3mm XL: **12.81✕** more speedup than prior SOTA
- GC exceeds prior SOTA performance
  - 1st evaluation: **50%**
  - Within budget: **80%**
- Worst margin of performance is **-0.24✕** speedup

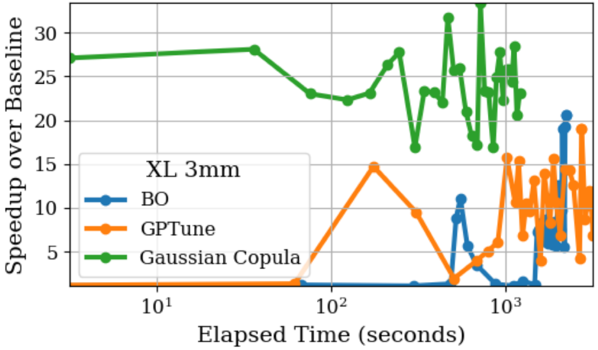
App.	Scale	Peak Speedup (# Evaluation Discovered)				
		GC			BO	GPTune
		1^{st}	Budget	Best	Best	Best
3mm	SM	5.09	5.70 (23)	5.70 (23)	3.03 (26)	5.53 (30)
	ML	5.25	5.57 (29)	5.57 (29)	3.29 (30)	5.16 (16)
	XL	27.10	33.39 (18)	33.39 (18)	20.58 (30)	18.96 (25)
Cov.	SM	21.10	21.98 (21)	21.98 (21)	21.83 (28)	13.30 (30)
	ML	4.13	4.27 (26)	4.27 (26)	3.87 (25)	4.07 (30)
	XL	23.04	23.96 (2)	23.96 (2)	8.43 (12)	17.88 (9)
Floyd-W.	SM	1.01	1.02 (17)	1.02 (17)	1.02 (20)	1.01 (26)
	ML	1.02	1.02 (1)	1.02 (1)	1.01 (25)	1.01 (3)
	XL	0.99	1.00 (29)	1.00 (29)	1.01 (16)	1.01 (20)
Heat3d	SM	1.83	2.03 (5)	2.06 (18)	2.21 (15)	2.30 (28)
	ML	1.89	1.89 (1)	2.06 (10)	2.12 (25)	1.80 (6)
	XL	1.50	2.92 (2)	3.09 (18)	2.16 (13)	2.75 (29)
LU	SM	1.16	1.18 (25)	1.18 (25)	1.12 (30)	1.11 (19)
	ML	1.15	1.20 (24)	1.20 (24)	1.17 (26)	1.19 (5)
	XL	1.00	1.00 (3)	1.00 (3)	0.98 (13)	1.00 (29)
Syr2k	SM	2.06	2.90 (2)	3.32 (18)	2.34 (12)	2.41 (11)
	ML	0.80	1.17 (2)	1.22 (16)	0.93 (29)	0.85 (30)
	XL	0.95	1.09 (2)	1.09 (2)	0.42 (23)	0.85 (26)

Polybench Demonstrates Consistency

- GC selects better configuration than prior work almost every single evaluation







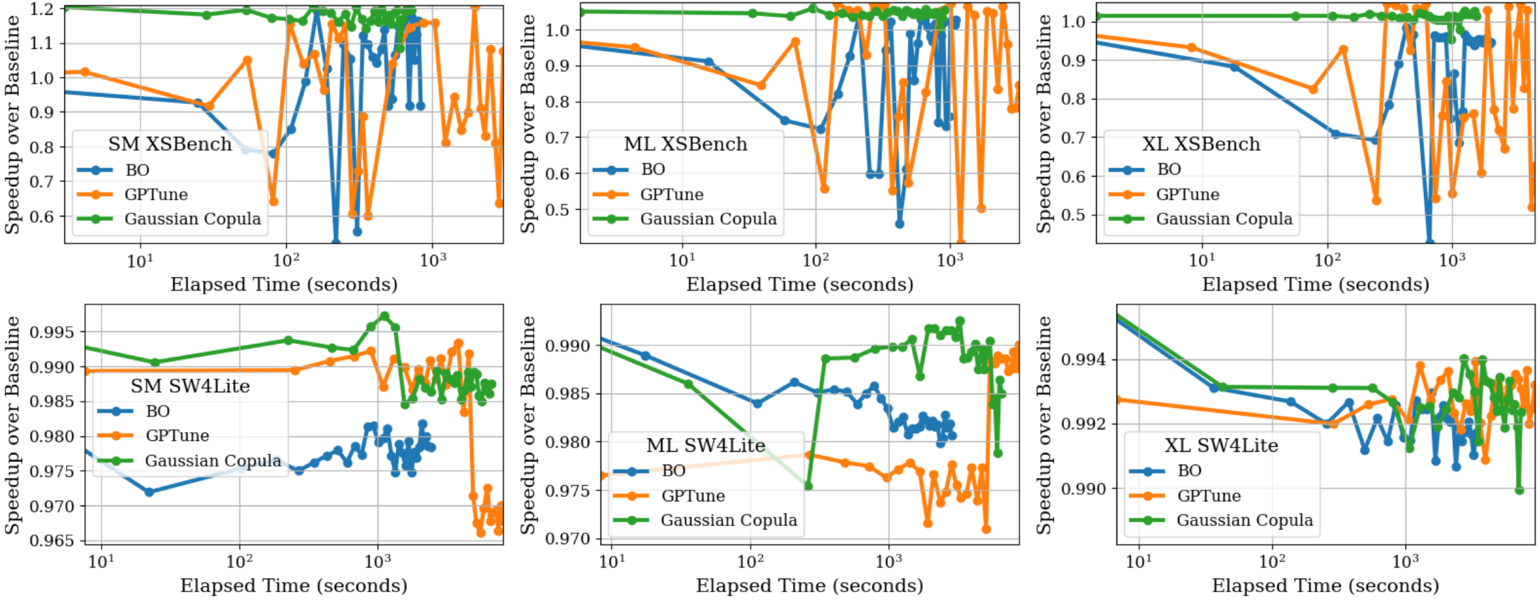
ECP Demonstrates Sophistication

- Speedup is difficult!!
- GC's best results achieved **on-budget**
- GC continues to **succeed** with *complex* spaces
- Worst margin of performance is **-0.02×** speedup

App.	Scale	Peak Speedup (# Evaluation Discovered)				
		GC			BO	GPTune
		1^{st}	Budget	Best	Best	Best
AMG	SM	0.87	0.91 (3)	0.91 (3)	0.92 (19)	0.90 (19)
	ML	0.93	0.93 (1)	0.93 (1)	0.93 (20)	0.87 (3)
	XL	0.95	0.95 (5)	0.98 (23)	0.97 (27)	0.93 (25)
RSBench	SM	1.40	1.40 (3)	1.40 (8)	1.25 (29)	1.13 (22)
	ML	1.02	1.04 (2)	1.04 (15)	0.97 (22)	1.04 (27)
	XL	1.00	1.00 (1)	1.01 (10)	0.97 (14)	1.02 (18)
XSBench	SM	1.20	1.20 (7)	1.21 (28)	1.17 (24)	1.21 (24)
	ML	1.05	1.06 (4)	1.06 (4)	1.04 (6)	1.07 (5)
	XL	1.01	1.02 (5)	1.03 (24)	0.99 (6)	1.05 (5)
SW4Lite	SM	0.99	1.00 (6)	1.00 (6)	0.98 (26)	0.99 (17)
	ML	0.99	0.99 (10)	0.99 (16)	0.99 (3)	0.99 (30)
	XL	0.99	0.99 (12)	0.99 (12)	0.99 (1)	0.99 (14)

Continued Success with Greater Complexity

- Better budget result in less time than prior work



Conclusions and Future Work

- Few-shot TL with GC
  - Simple definition
  - Aggressive search for high-performing results
  - Able to predict search budget
    - Minimize costs, estimate utility
- Future work
  - Enhance GC
  - Apply to full ECP applications

Open Source: https://github.com/tlranda/GC_TLA

Contact: tlranda@clemson.edu

This research was partially supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy Office of Science and the National Nuclear Security Administration, and by U.S. National Science Foundation under Grant CCF-1942182. This material is based upon work supported by the U.S. Department of Energy, Office of Science, under contract number DE-AC02-06CH11357.