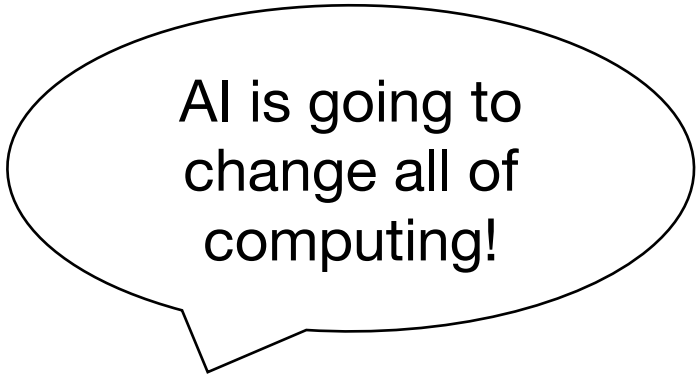
What are the Unique Research Opportunities in Systems for ML?

Matei Zaharia

Stanford Platform Lab Review, Feb 2020



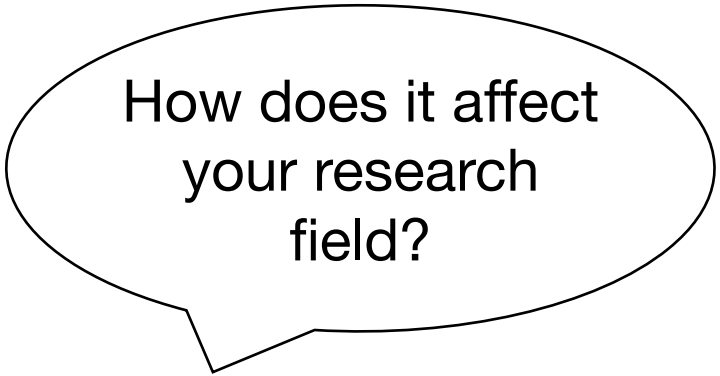
AI Researcher

Systems Researcher

It’s intelligent and you don’t need to program anymore and you just differentiate things...

AI Researcher

Systems Researcher



AI Researcher

Systems Researcher

How does it affect your research field?

AI Researcher

Umm, I figured out a way to save some system calls!

Systems Researcher

How does it affect your research field?

AI Researcher

I designed a new accelerator

Architecture Researcher

AI Researcher

Architecture Researcher

Motivation

ML workloads can certainly influence a lot of systems, but what are the **unique** research challenges they raise?

Turns out there are a lot! ML is very different from traditional software, and we should look at how

How Does ML Differ from Traditional Software?

Traditional Software	Machine Learning
Goal: meet a functional specification	Goal: optimize a metric (e.g. accuracy)
Quality depends only on application code	Quality depends on input data and tuning parameters
Mostly deterministic	Stochastic

Some Interesting Opportunities

Data-oriented model training, QA and debugging tools

Optimizations that leverage the stochastic nature of ML

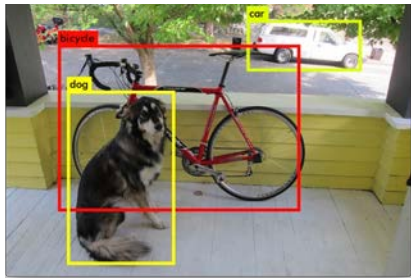
New dimensions for parallelizing ML

ML-Aware System Optimization: NoScope & BlazeIt

The ML Inference Bottleneck

Inference cost is often 100x higher than training overall, and greatly limits deployments

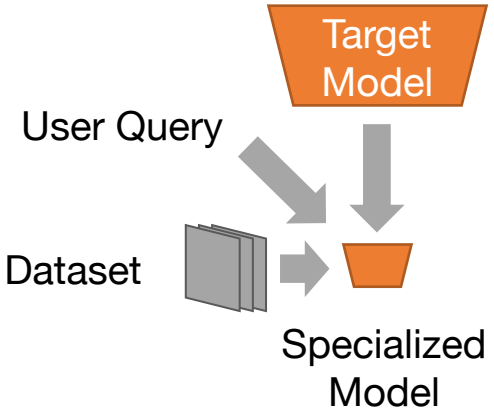
Example: processing 1 video stream in real time with CNNs requires a $1000 GPU



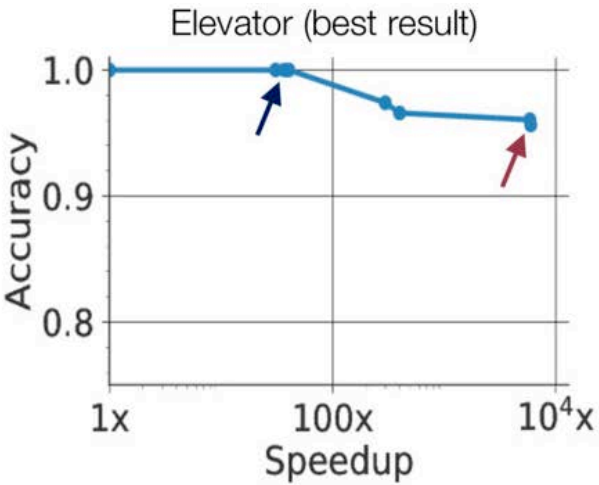
Inference Optimization in NoScope

Idea: optimize execution of ML models for a *specific* application or query

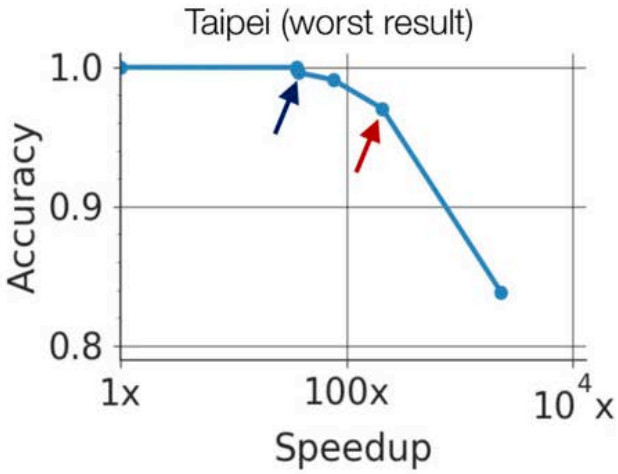
- **Model specialization:** train a small DNN to recognize the specific class in the dataset (e.g. “buses in street video”)
- **Query optimization:** tune a cascade of models to achieve a target accuracy



NoScope Results



40x faster @ 99.9% accuracy
5858x faster @ 96% accuracy



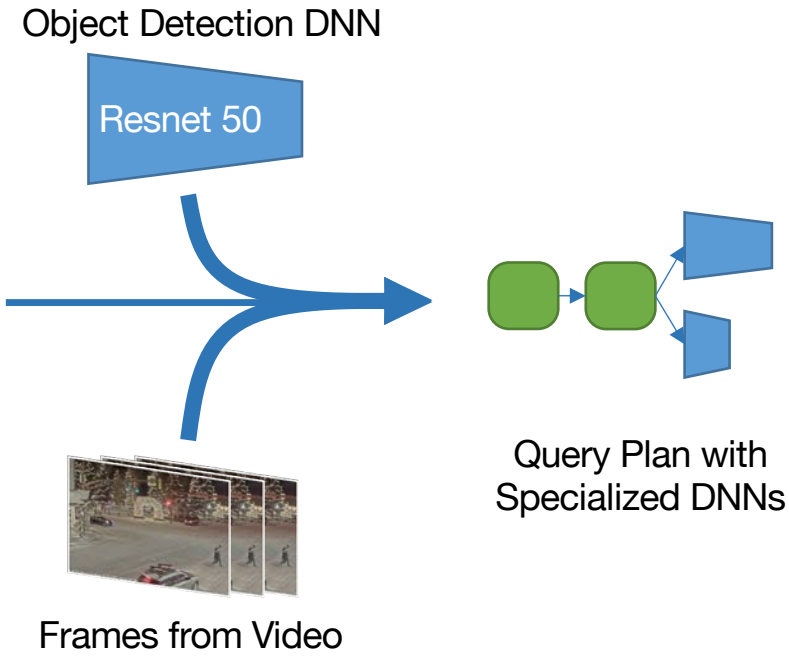
36.5x faster @ 99.9% accuracy
206x faster @ 96% accuracy

VLDB ‘17, github.com/stanford-futuredata/noscope

Optimizing ML + SQL in BlazeIt

```
SELECT timestamp
FROM taipei
GROUP BY timestamp
HAVING SUM(class='bus')>=1
   AND SUM(class='car')>=5
LIMIT 10 GAP 300
```

SQL Query



[Kang et al, VLDB 2020]

BlazeIt Optimizations

Aggregation Queries

Accelerate approximate queries by using specialized model's output as a *control variate* for sampling

E.g.: find average # of cars/frame

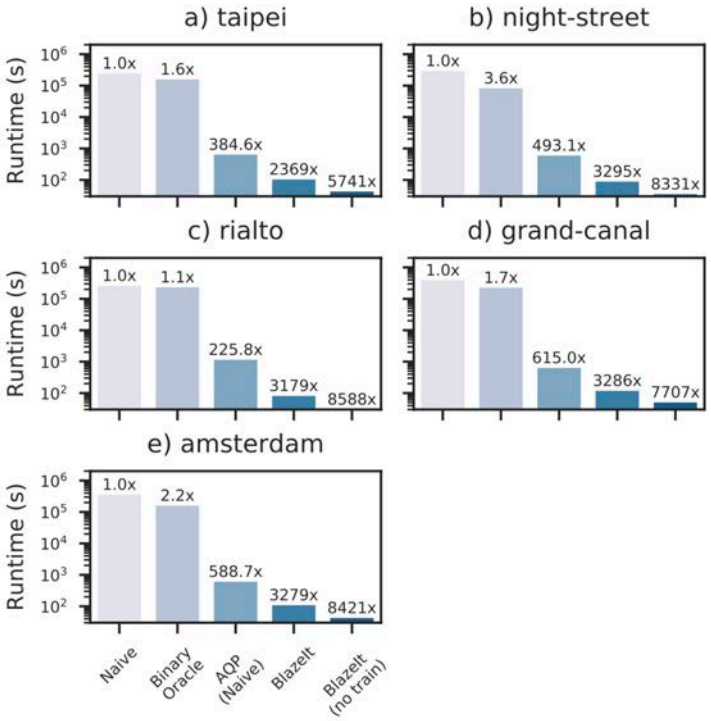
Limit Queries

Use specialized models to sort frames by likelihood of matching query, then run full model

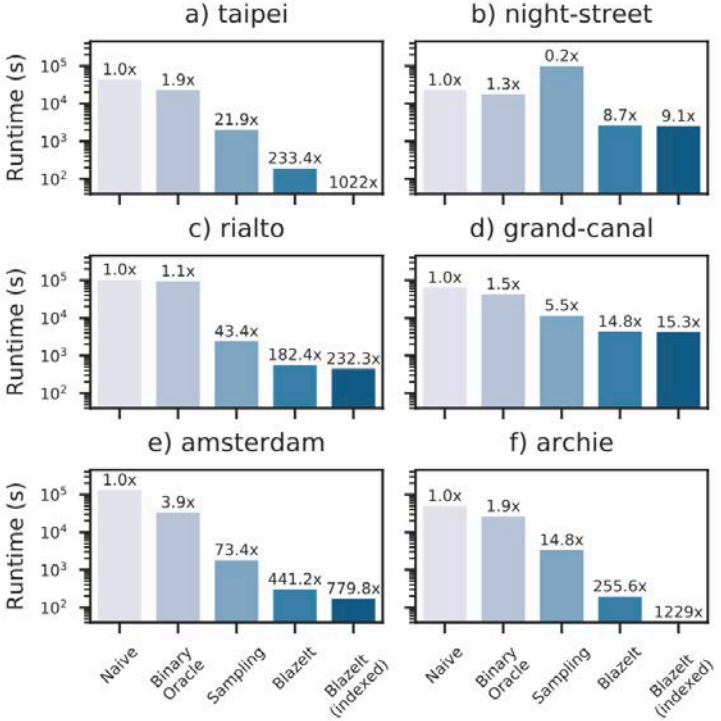
E.g.: SELECT * FROM frames WHERE #(red buses) > 3 LIMIT 5

BlazeIt Results

Aggregation Queries



Limit Queries



Quality Assurance for ML with Model Assertions

Motivation

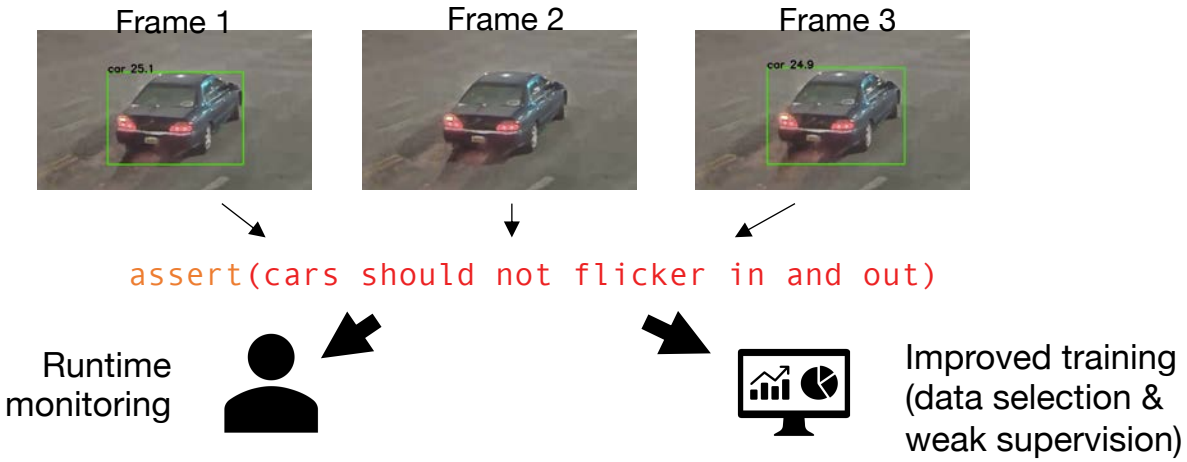
ML applications fail in complex, hard-to-debug ways

- Tesla cars crashing into lane dividers
- Gender classification incorrect based on race

How can we **test** and **improve quality of** ML apps?

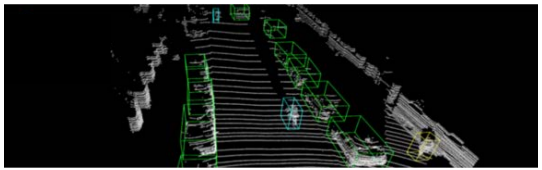
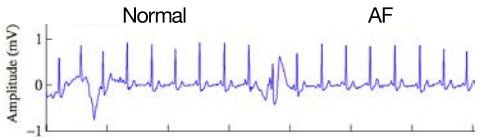
Model Assertions

Predicates on input/output of an ML application (similar to software assertions)



[Kang, Raghavan et al, MLSys 2020]

Example Assertions

Problem Domain	Assertion	
Video analytics	Objects should not flicker in and out across frames	
Autonomous vehicles	LIDAR and video object detectors should agree	
Heart rhythm classification	Output class should not change frequently	

Using Model Assertions

Inference time

» Runtime monitoring

» Corrective action

Training time

» Active learning

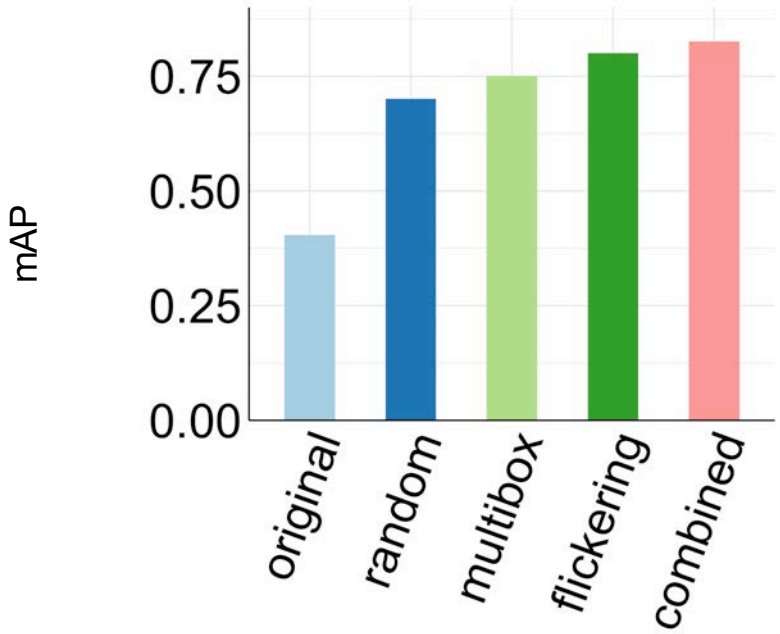
» Weak supervision via correction rules

Active Learning with Assertions:
Can assertions help select data to label & train on?

Key idea: new active learning algorithm samples data that is most likely to reduce # failing assertions

Active Learning with Assertions:

Can assertions help select data to label & train on?



Using assertions for active learning improves model quality.

Weak Supervision with Assertions:
Can assertions improve quality without human labeling?

Key idea: consistency constraints API lets devs say which attributes should stay constant across outputs in a dataset

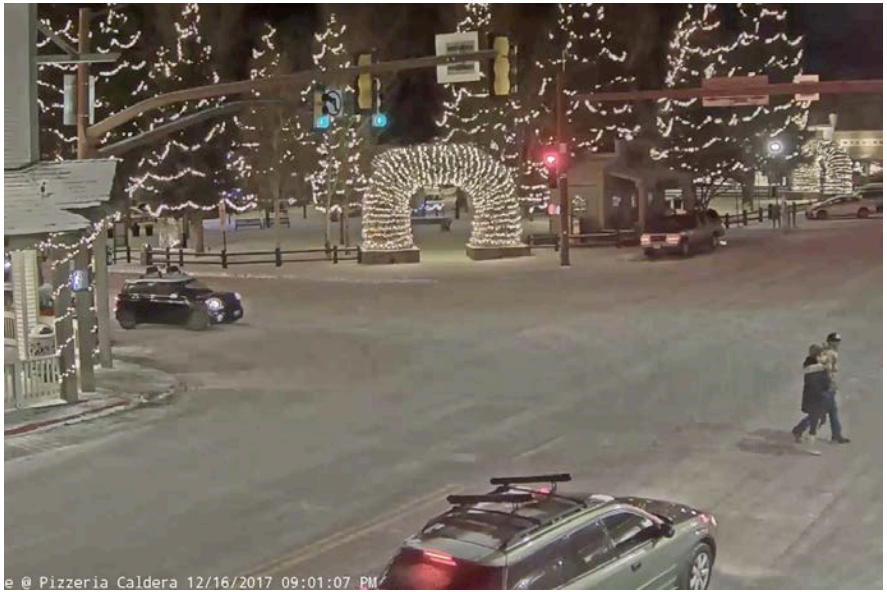
E.g. “each tracked object should always have same class”

Weak Supervision with Assertions:
Can assertions improve quality without human labeling?

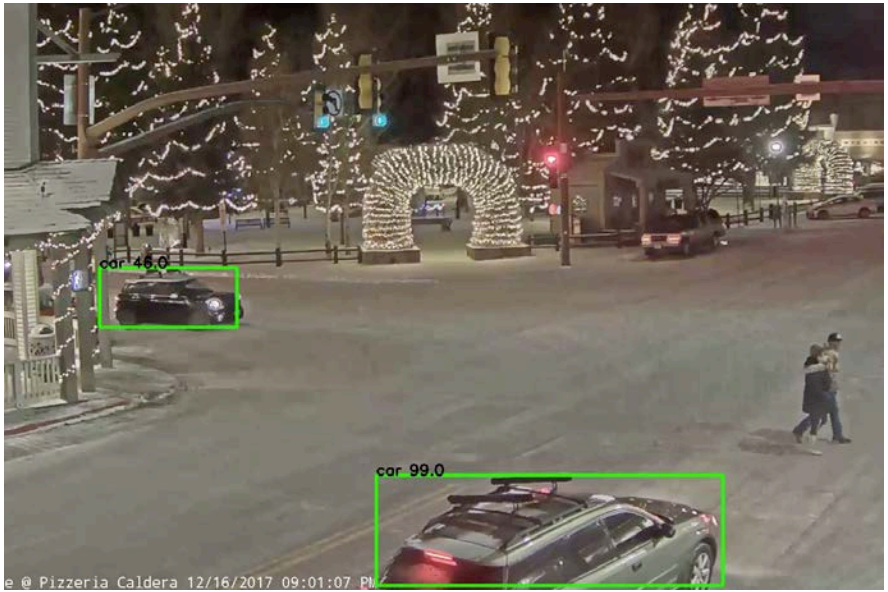
Task	Pretrained	Weakly Supervised
AV perception (mAP)	10.6	14.1 (+33%)
Object detection (mAP)	34.4	49.9 (+45%)
ECG (% accuracy)	70.7	72.1 (+2%)

Model Quality After Retraining

Original SSD Model



Retrained SSD Model



[Kang, Raghavan et al, NeurIPS MLSys 2018]

New Dimensions for Parallel ML

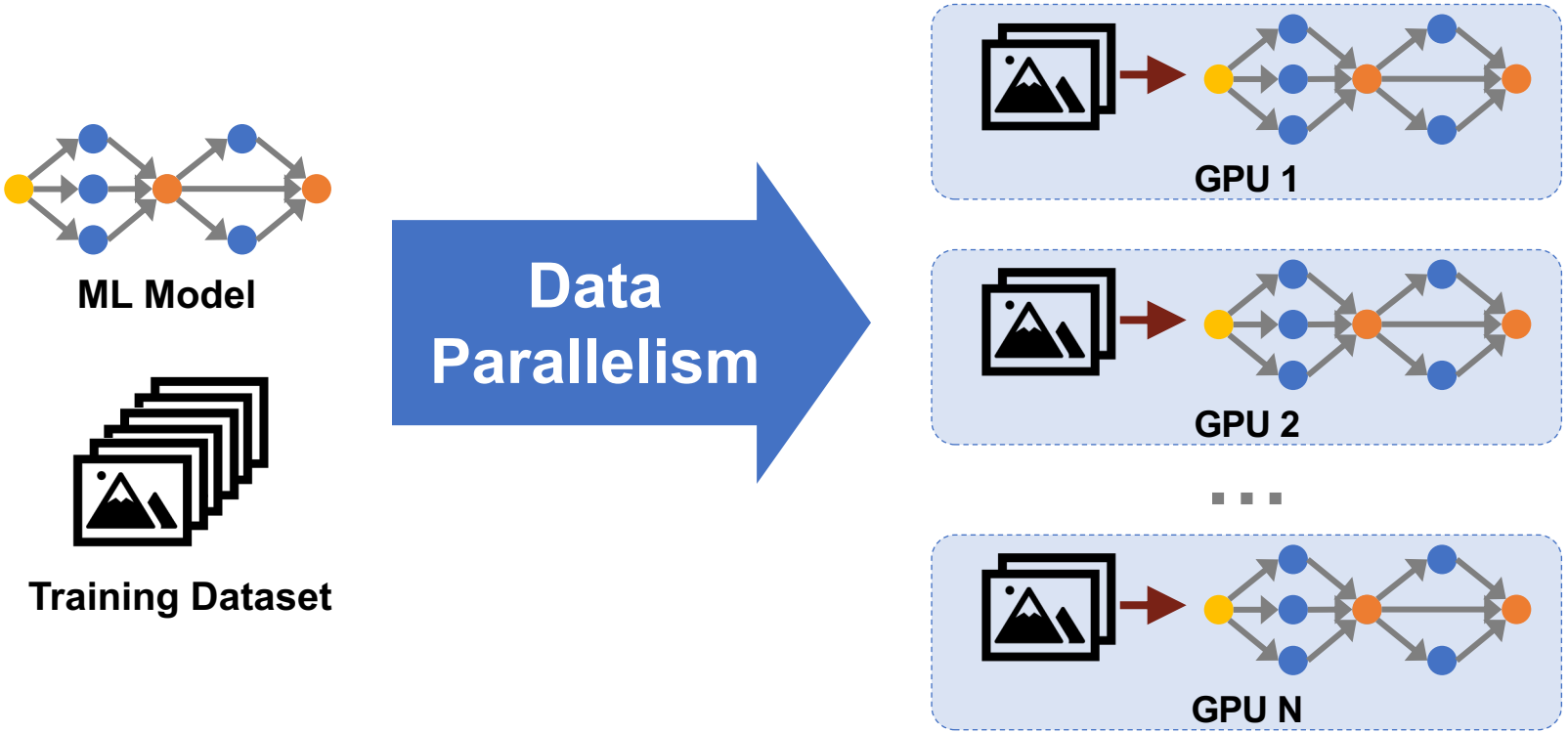
Motivation

Unlike most software, ML is dominated by dense linear algebra operators

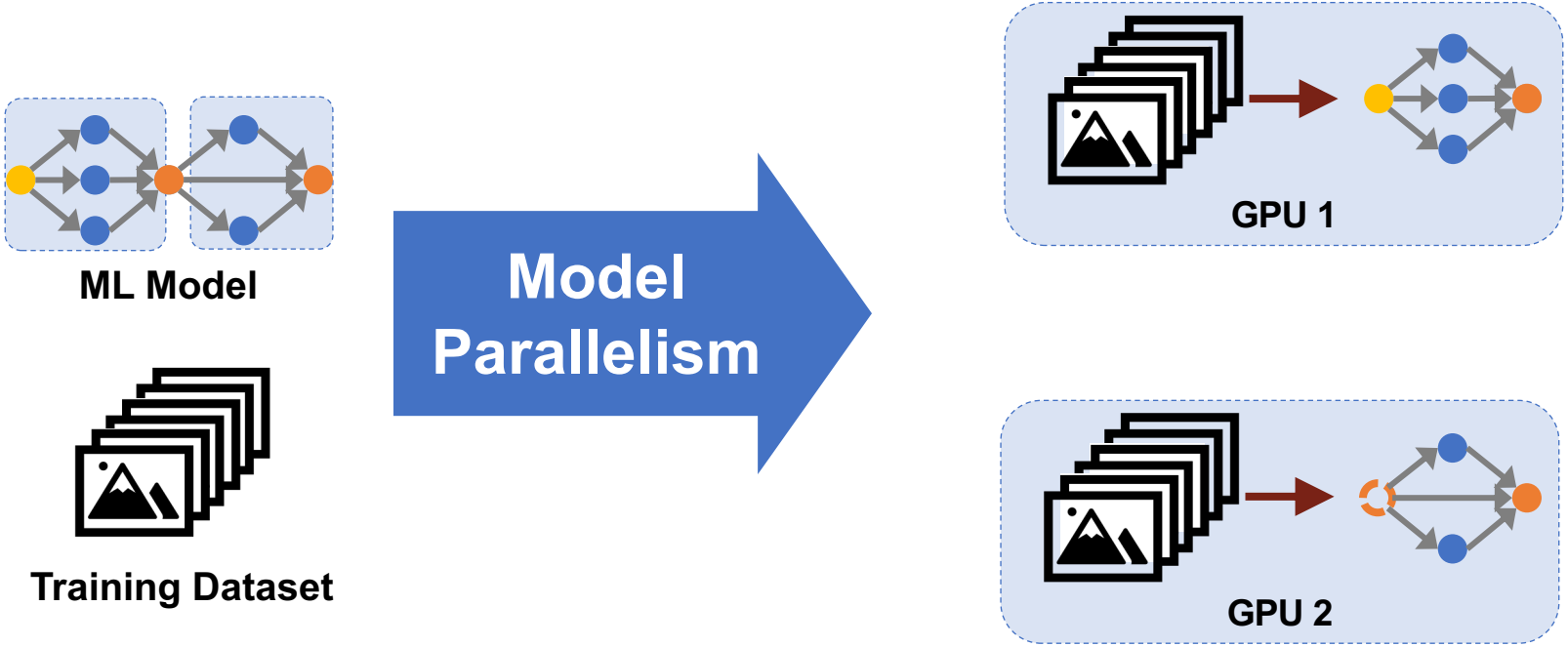
Many opportunities for **algebraic rewrites** that improve application performance

Example: FlexFlow project [Jia, Zaharia & Aiken, SysML'19]

Current Strategies to Parallelize ML Training: Data and Model Parallelism



Current Strategies to Parallelize ML Training: Data and Model Parallelism



FlexFlow

- Consider a significantly **larger search space** of possible parallelization strategies beyond data and model parallelism
- **Fast and automated discovery** of high-performance strategies
- **Result**: accelerates training by up to **10x**

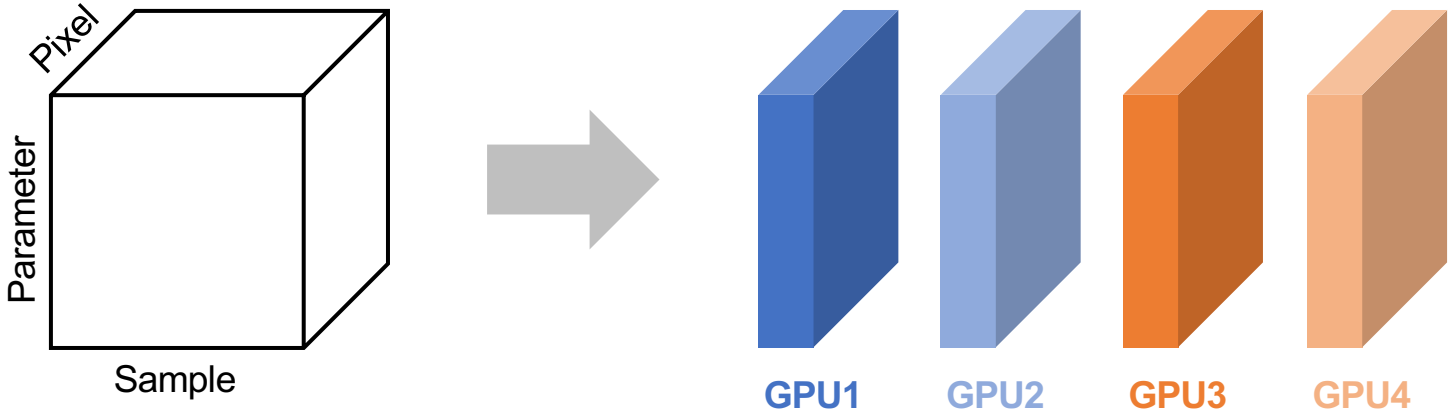
[Jia, Zaharia and Aiken, SysML 2019]

The SOAP Search Space

- **S**amples
- **O**perators
- **A**ttributes
- **P**arameters

The SOAP Search Space

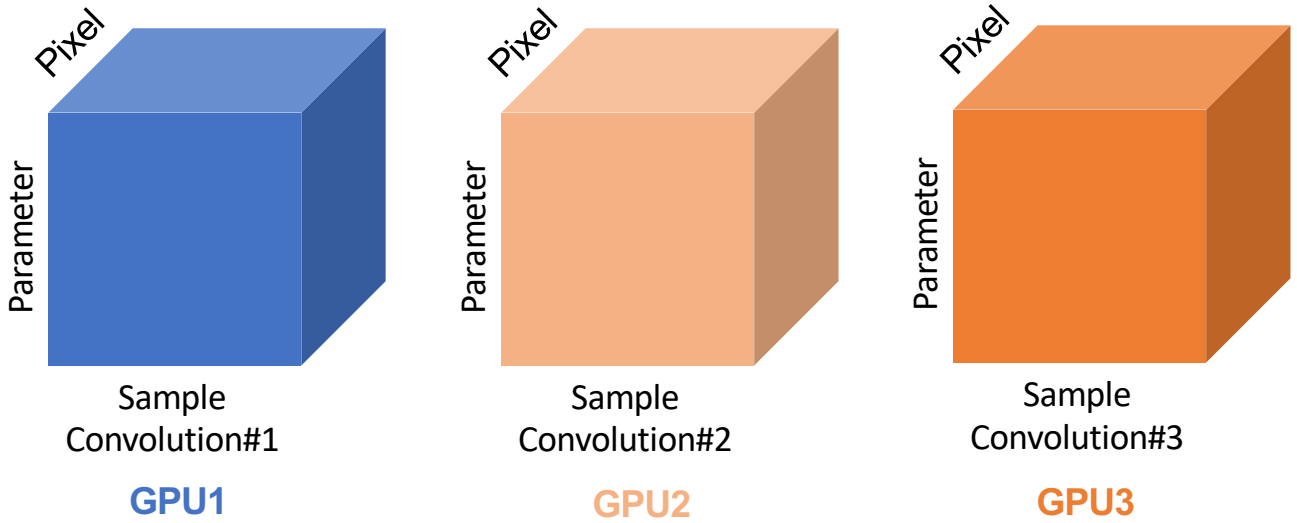
- **S**amples: partitioning training samples (Data Parallelism)
- **O**perators
- **A**ttributes
- **P**arameters



Parallelizing a 1D convolution

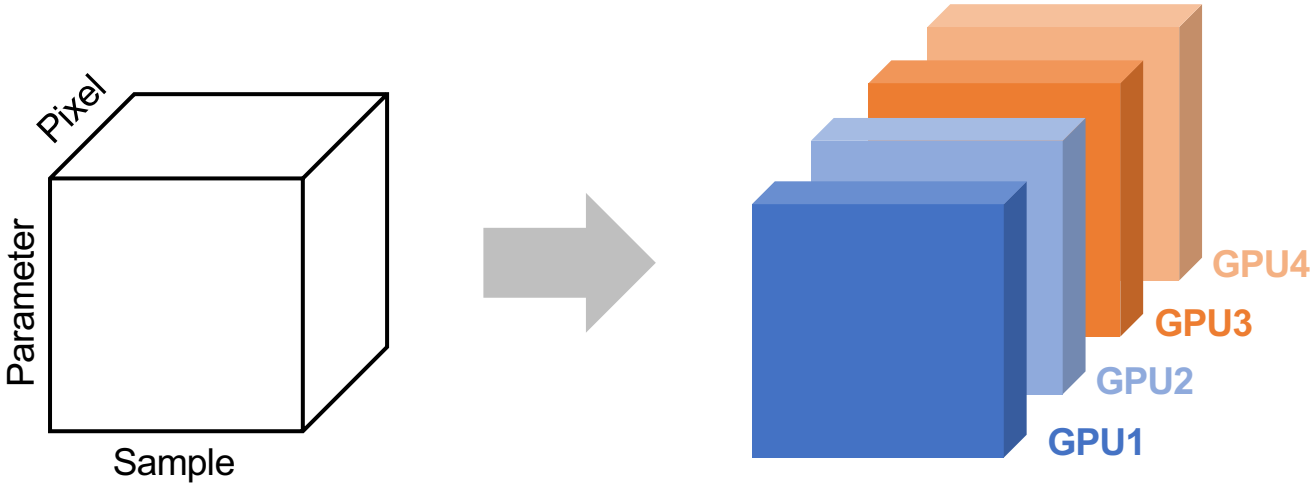
The SOAP Search Space

- **S**amples: partitioning training samples (Data Parallelism)
- **O**perators: partitioning DNN operators (Model Parallelism)
- **A**ttributes
- **P**arameters



The SOAP Search Space

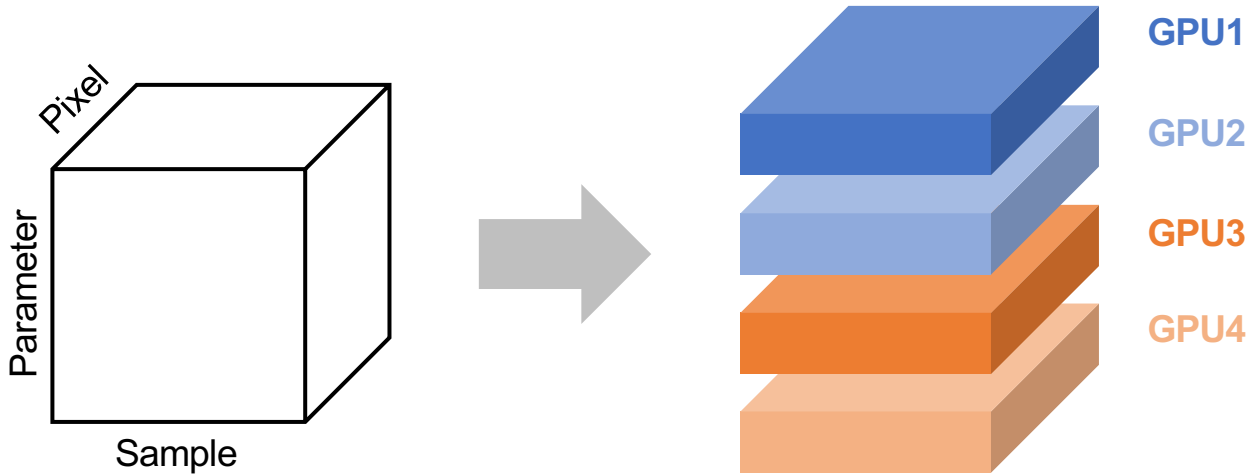
- **S**amples: partitioning training samples (Data Parallelism)
- **O**perators: partitioning DNN operators (Model Parallelism)
- **A**ttributes: partitioning attributes in a sample (e.g., pixels)
- **P**arameters



Parallelizing a 1D convolution

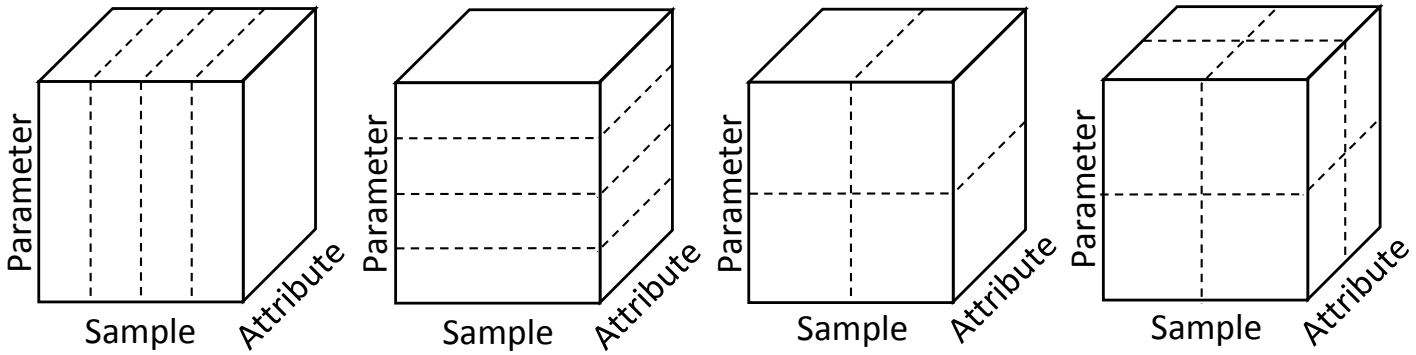
The SOAP Search Space

- **S**amples: partitioning training samples (Data Parallelism)
- **O**perators: partitioning DNN operators (Model Parallelism)
- **A**ttributes: partitioning attributes in a sample (e.g., pixels)
- **P**arameters: partitioning parameters in an operator



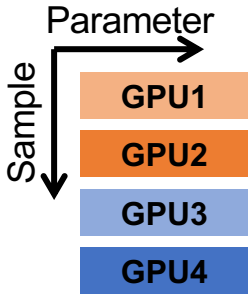
Parallelizing a 1D convolution

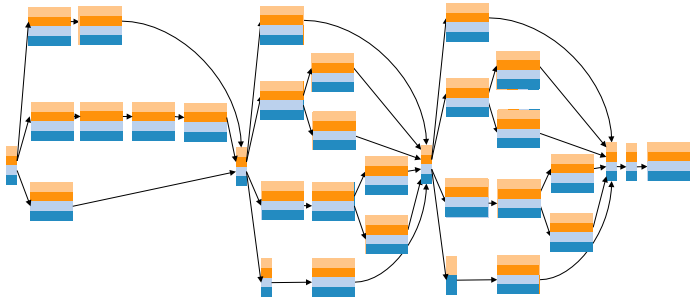
Hybrid Parallelism in SOAP



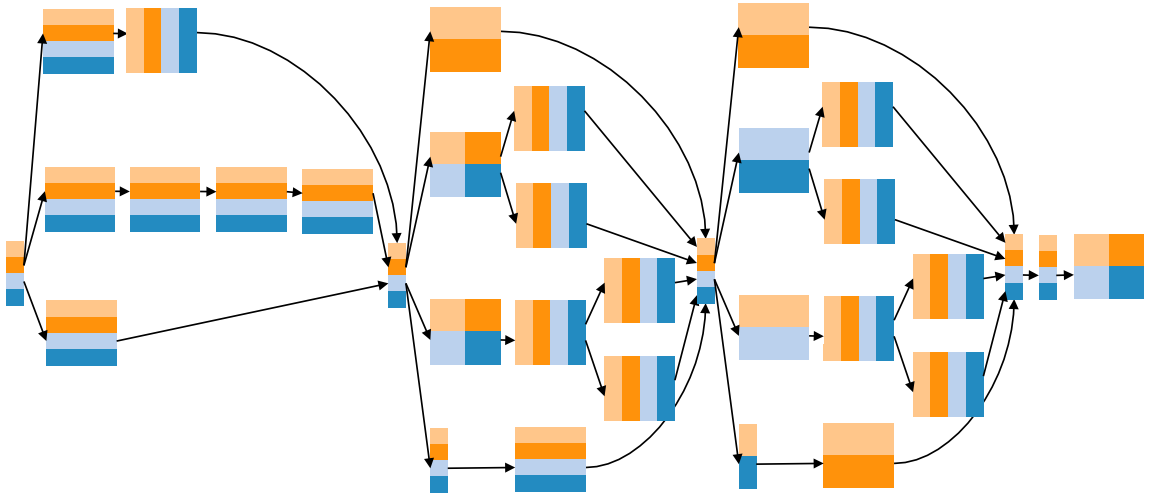
Example parallelization strategies for 1D convolution

Different strategies perform the same computation.





Data parallelism



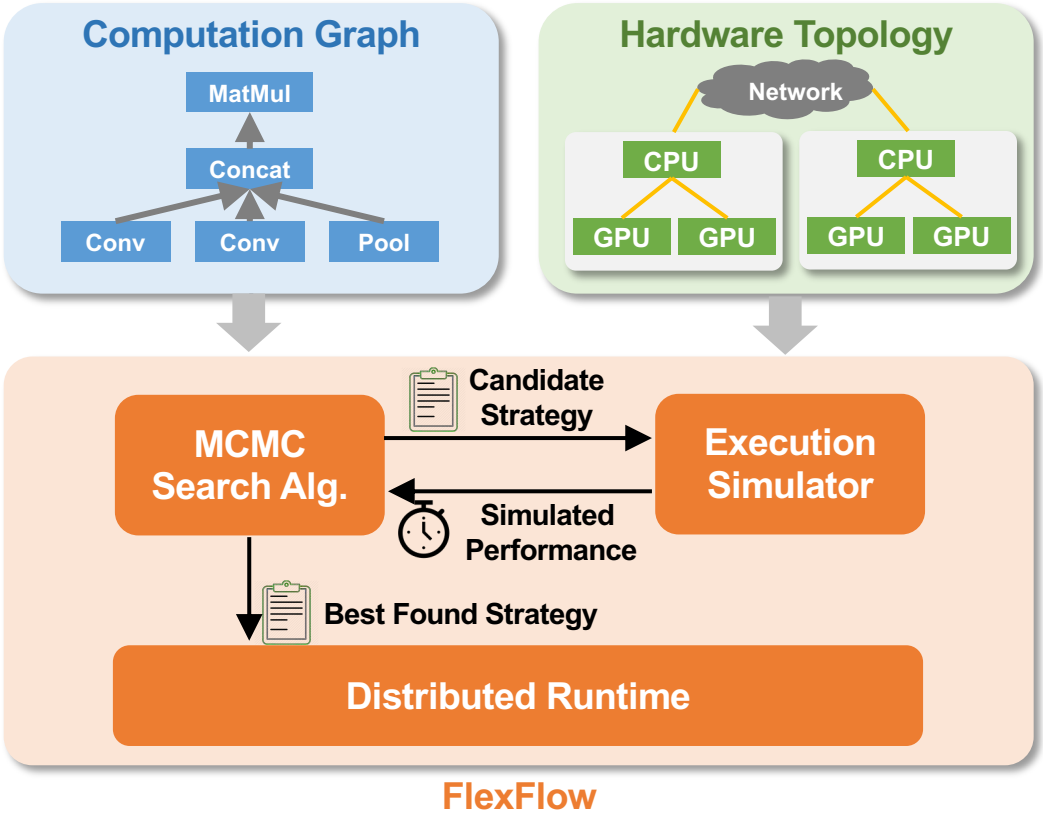
A parallelization strategy in SOAP **(1.2x faster)**

Challenges of Discovering Fast Strategies in SOAP

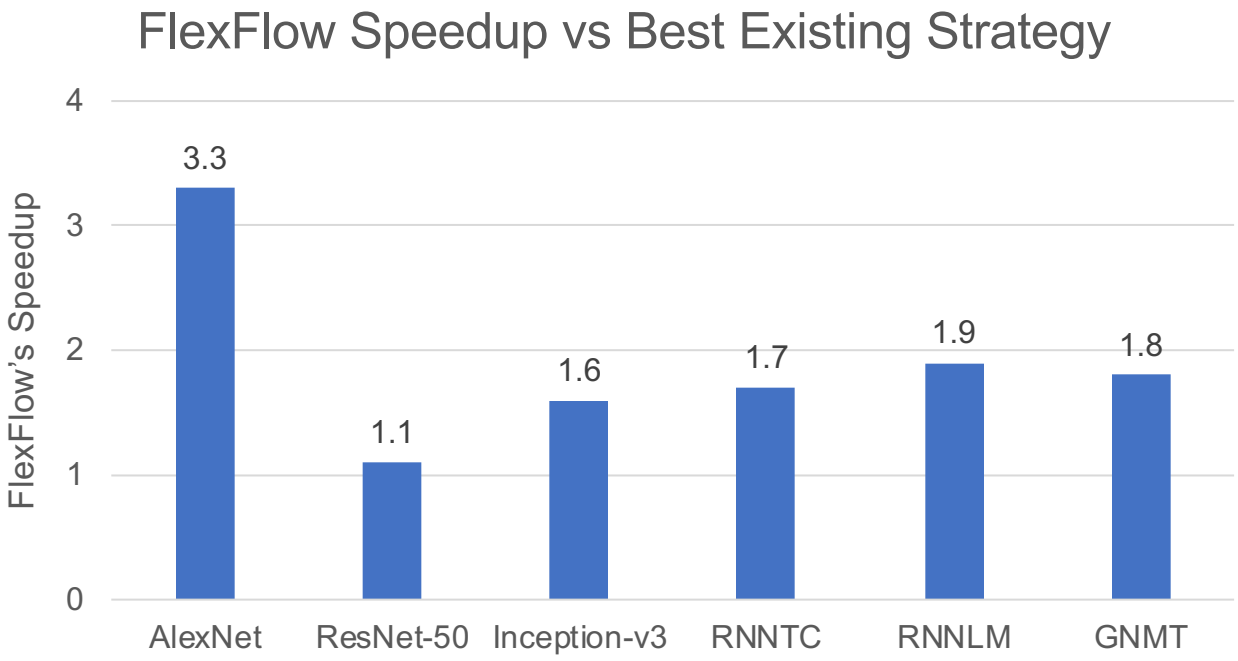
1. SOAP contains billions or more possible strategies

2. Evaluating a strategy on hardware is too slow

FlexFlow Implementation



End-to-end Training Performance



No changes in training semantics!

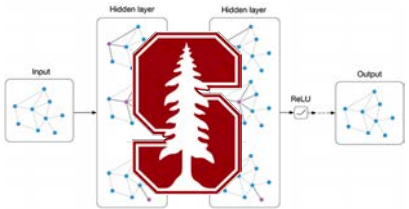
FlexFlow Impact

facebook

Accelerated training production models by **10x**.

Los Alamos NATIONAL LABORATORY

Reduced training time from **days** to **hours**.



Improved **scale** & **accuracy** for graph neural networks.

Conclusion

ML is very different from traditional applications and offers new opportunities for systems research

- Leverage the stochastic and algebraic nature of ML
- Manage the data coming in and out of ML
- Map to hardware in novel ways

Follow my group at cs.stanford.edu/~matei/