Importance of Generalizability in Machine Learning for Systems

Varun Gohil — MIT

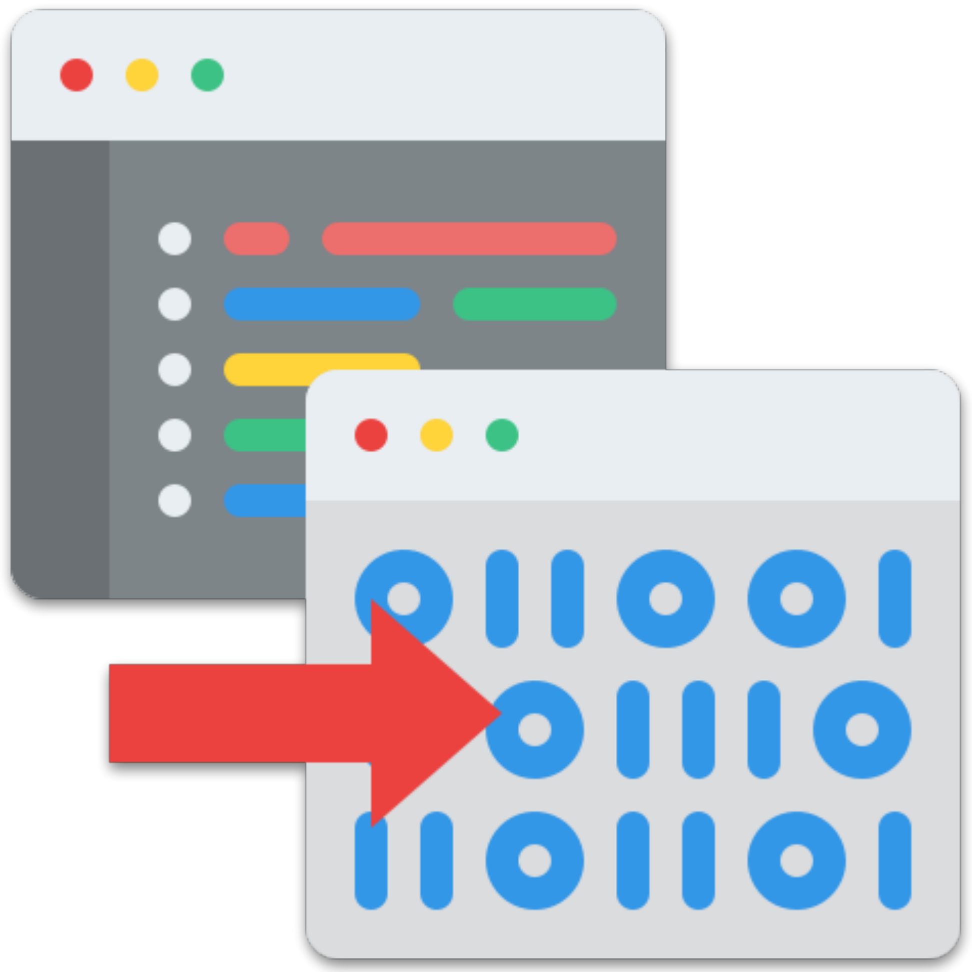
Sundar Dev
Gaurang Upasani
David Lo
Partha Ranganathan — Google

Christina Delimitrou — MIT

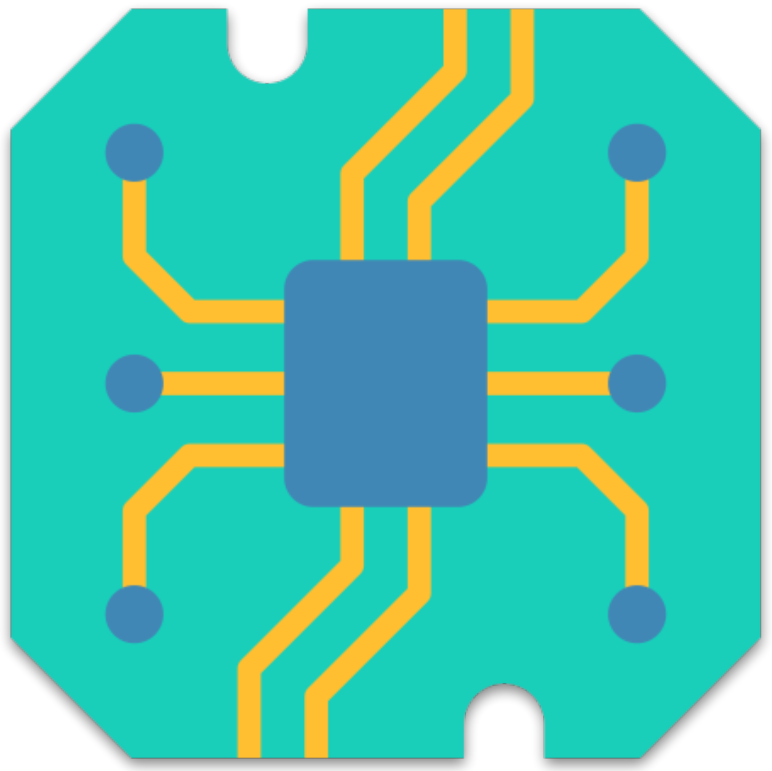
Machine Learning for Systems

Growing field with demonstrated effectiveness across multiple domains

Datacenter Scheduling



Compilers & Runtime

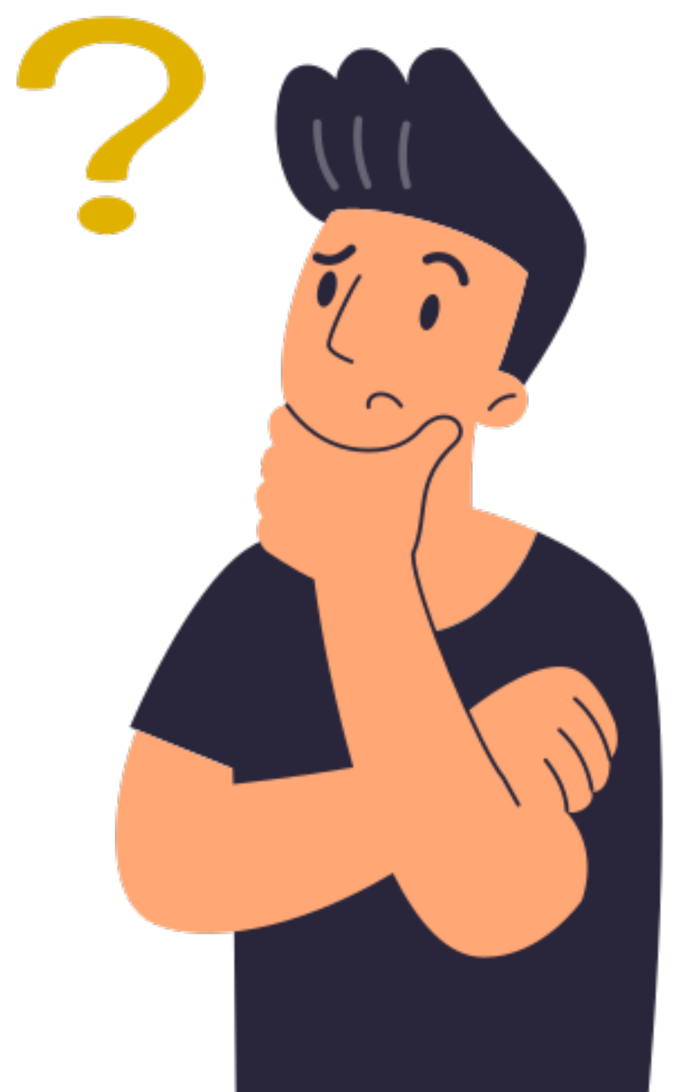


Hardware Design

Machine Learning is not reliable…

Mispredictions can be costly and difficult to debug

Can we avoid mispredictions to improve reliability?



Can we avoid mispredictions to improve reliability?

Yes! Don’t use predictions on data where model has poor generalizability

Can we avoid mispredictions to improve reliability?

Yes! Don’t use predictions on data where model has poor generalizability

Model’s ability to predict accurately on data which is not independent and identically distributed as training data

Proactively measure the model's generalizability on input data point

and ignore model prediction if generalizability is poor …

Proactively measure the model's generalizability on input data point

Best measure of generalizability is accuracy which cannot be measured proactively …

Proactively measure the model's ~~generalizability~~ on input data point uncertainty

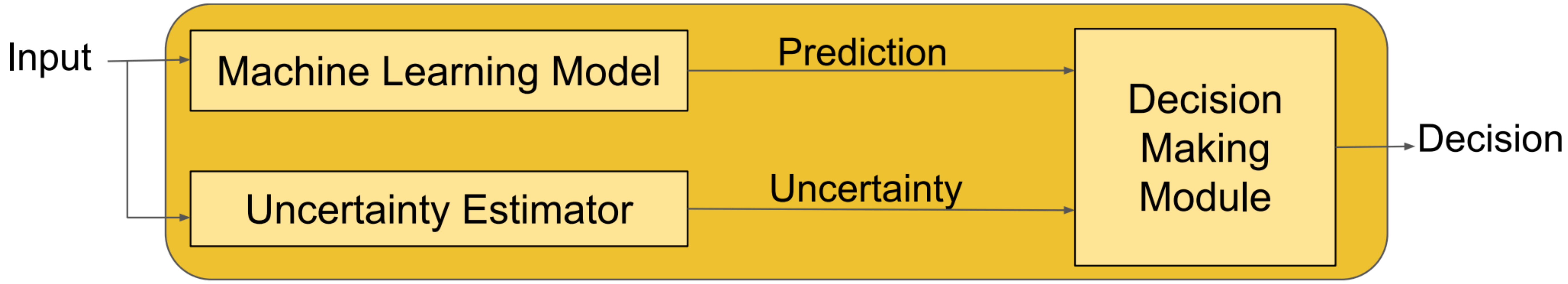
and ignore model prediction if uncertainty is high …

Executive Summary

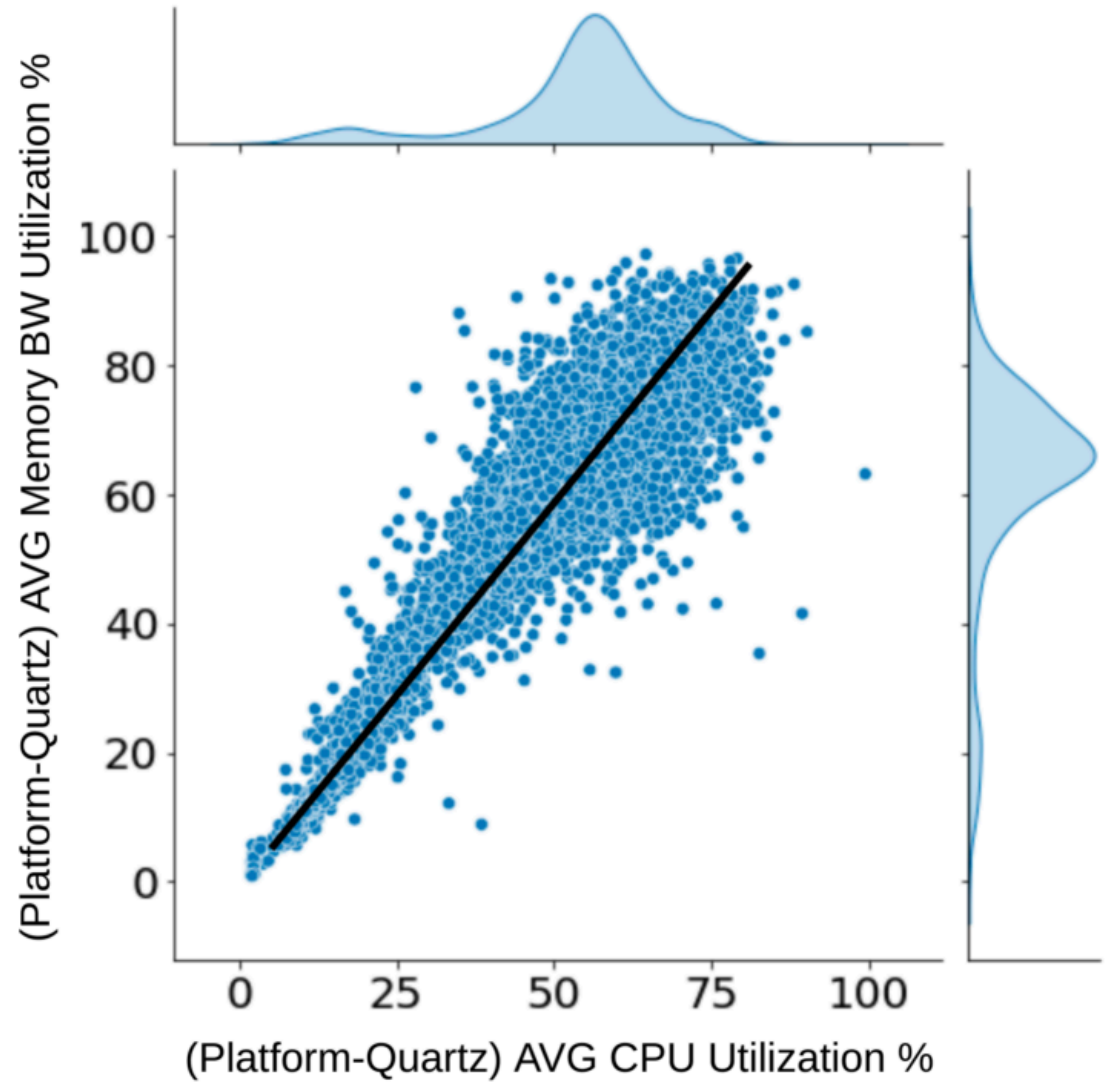
Problem: Models with **poor generalizability** causing costly mispredictions which make machine learning for systems solutions unreliable

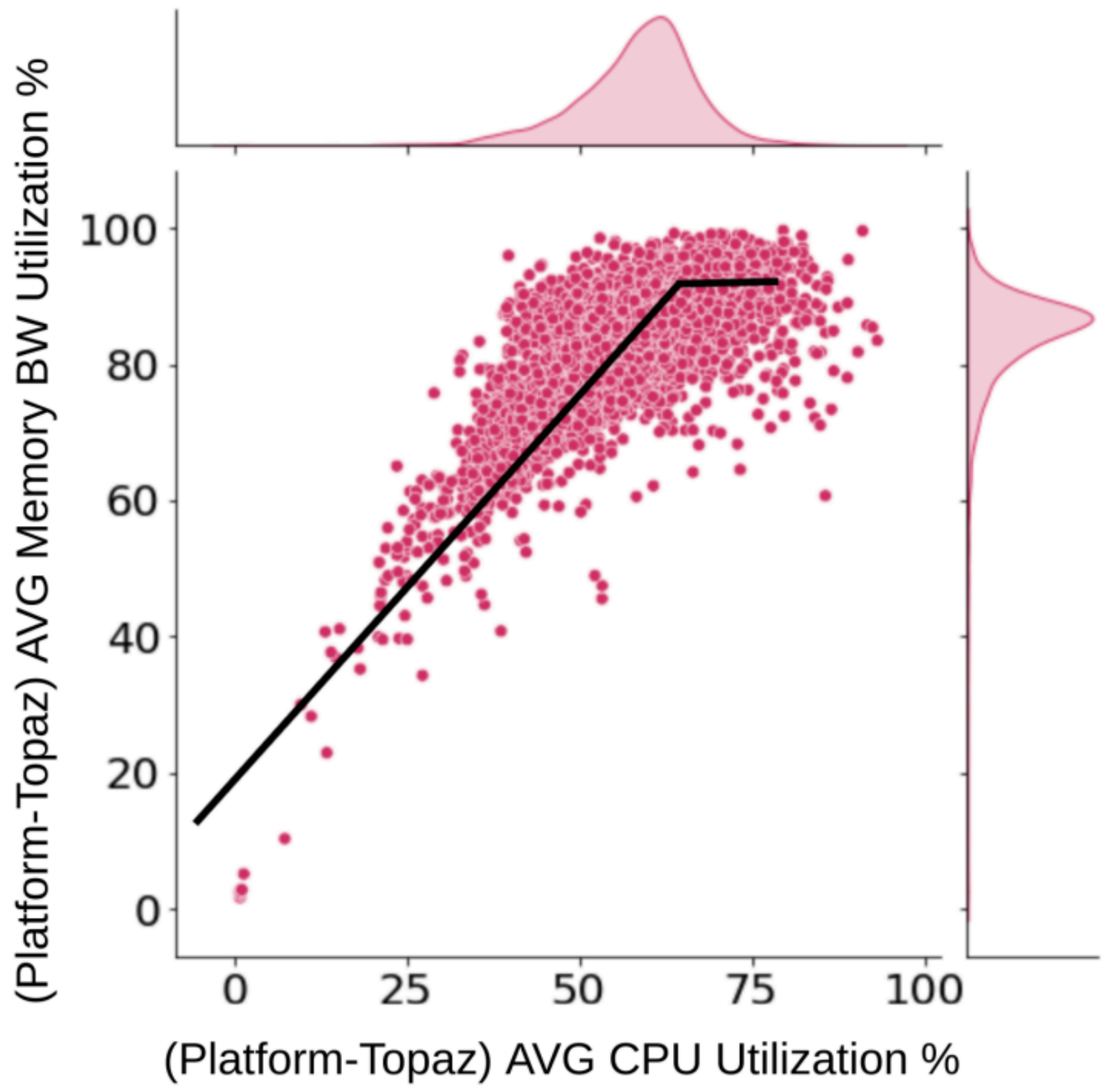
Goal: **Improve reliability** of machine learning for systems techniques by avoiding mispredictions

Solution: Workflow that **proactively uses proxies of generalizability** like uncertainty to guide model usage

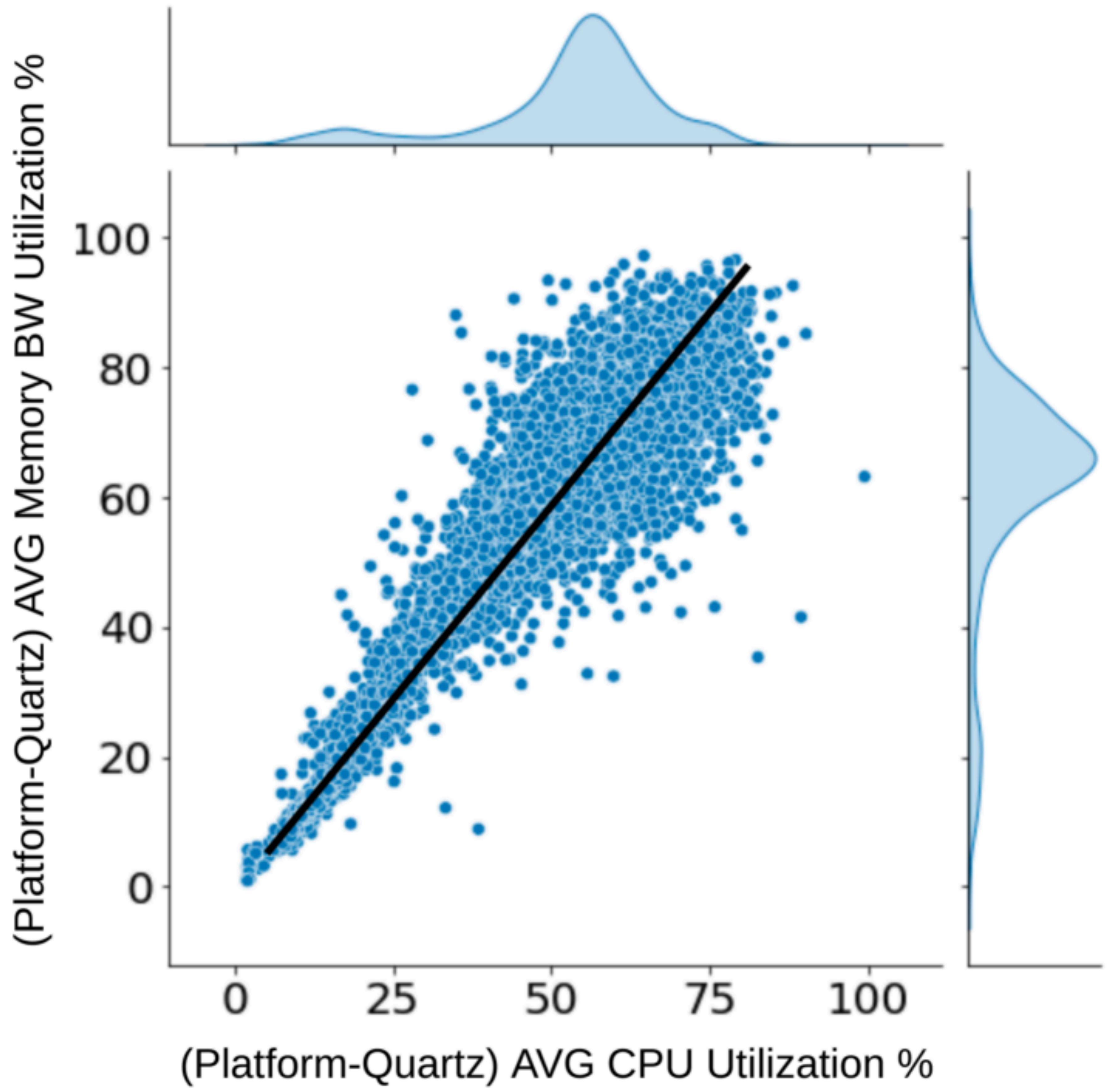


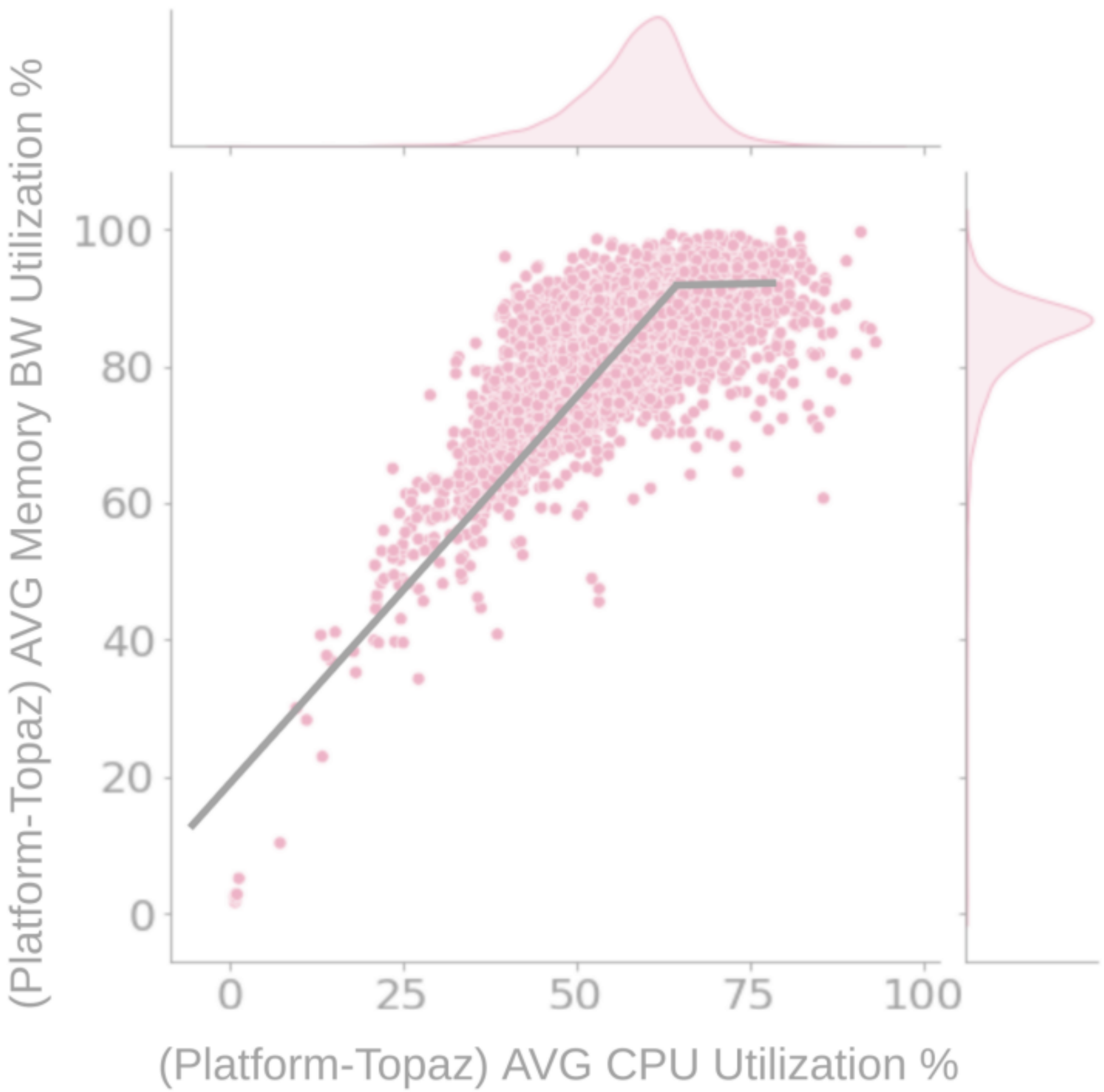
Under-provisioned Memory Bandwidth Capacity @ Google



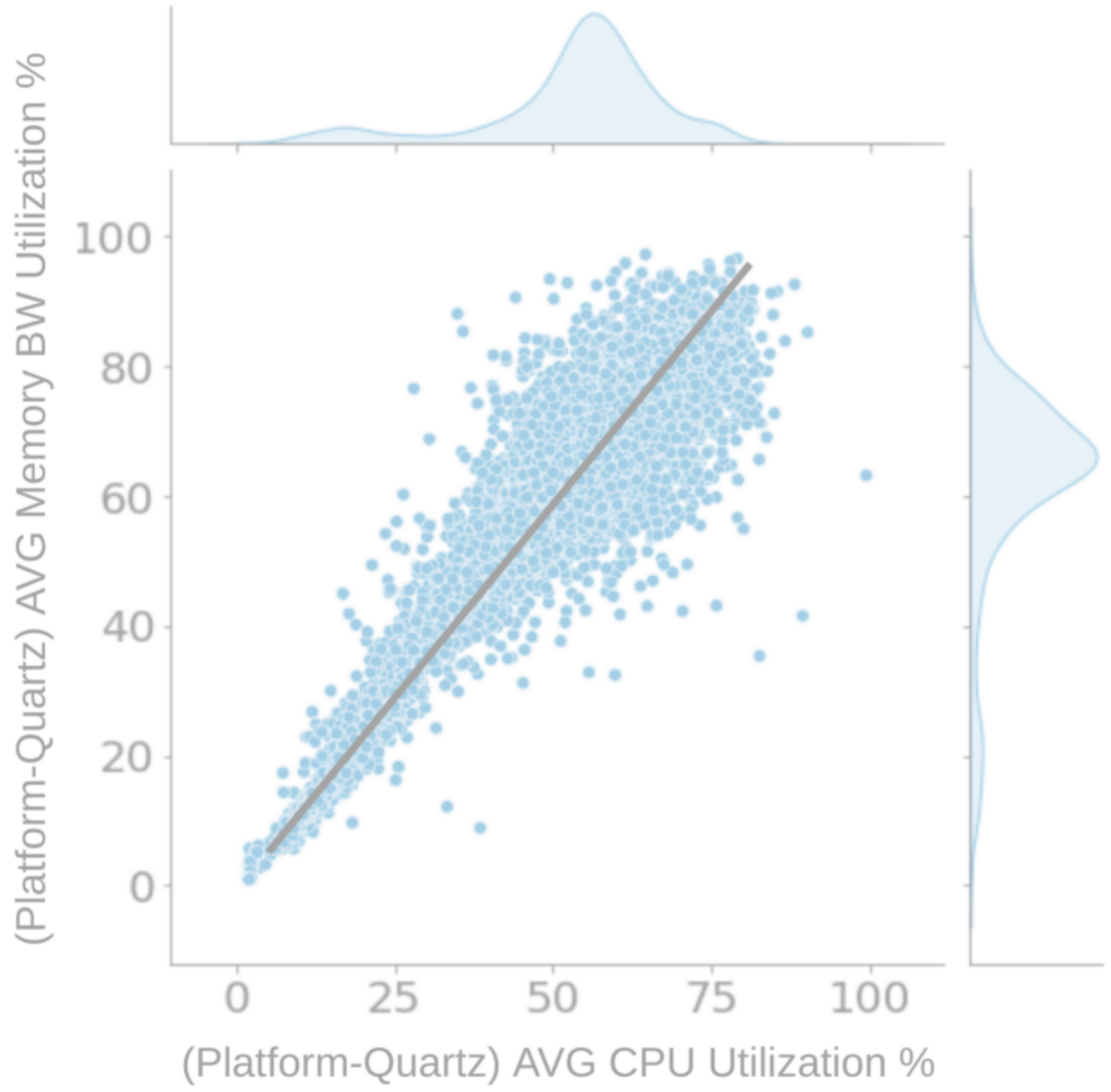


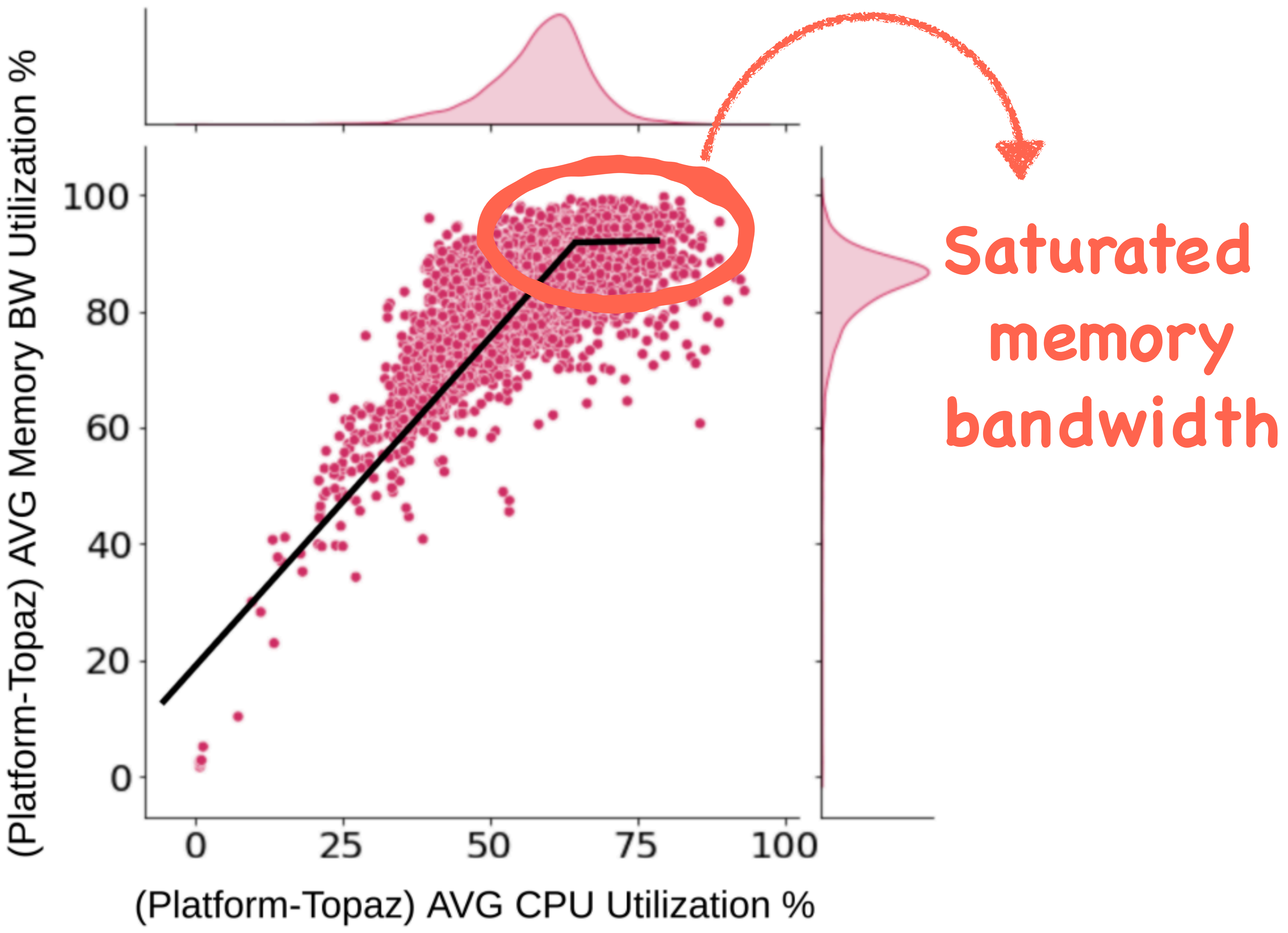
Under-provisioned Memory Bandwidth Capacity @ Google





Under-provisioned Memory Bandwidth Capacity @ Google

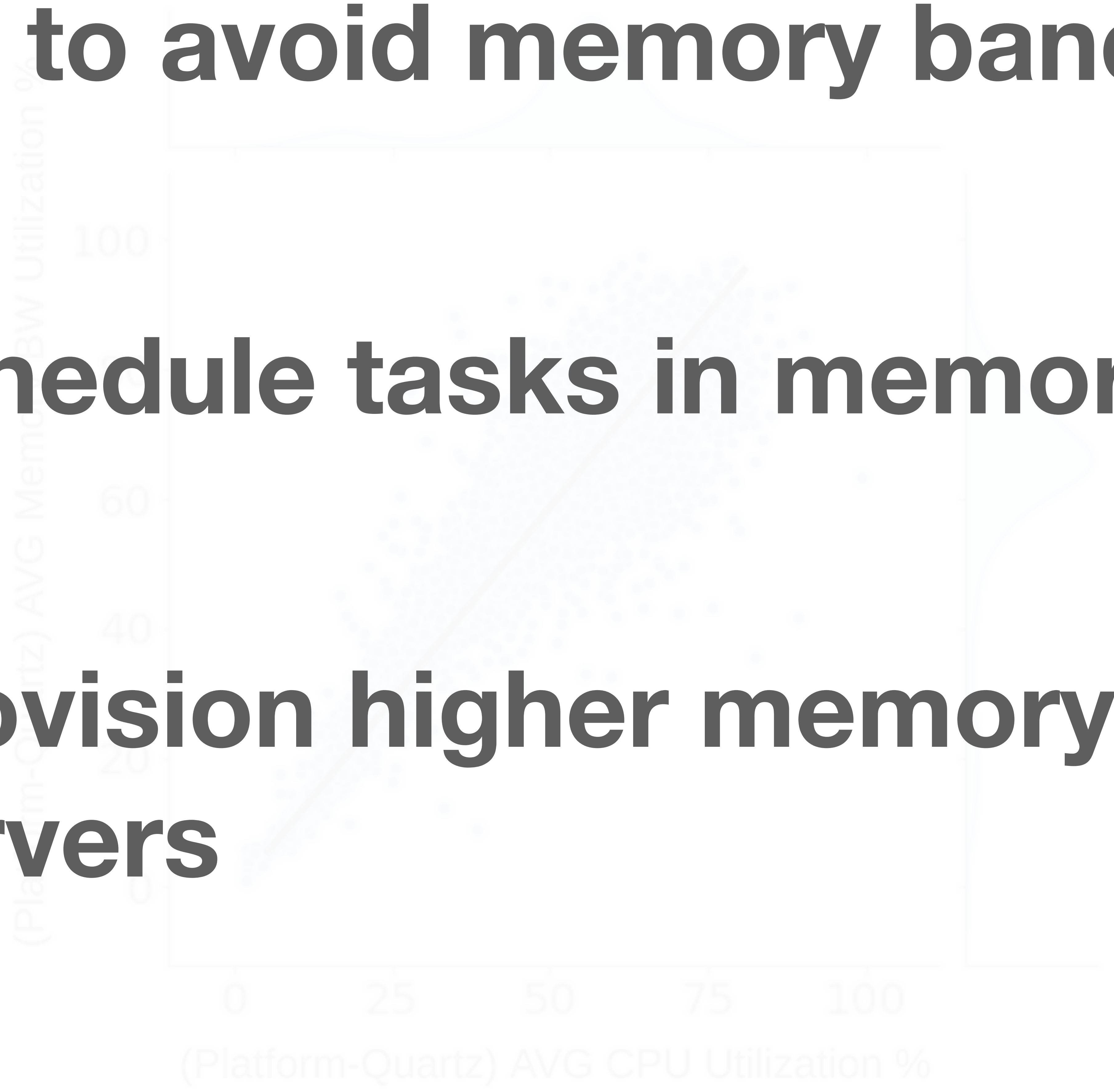


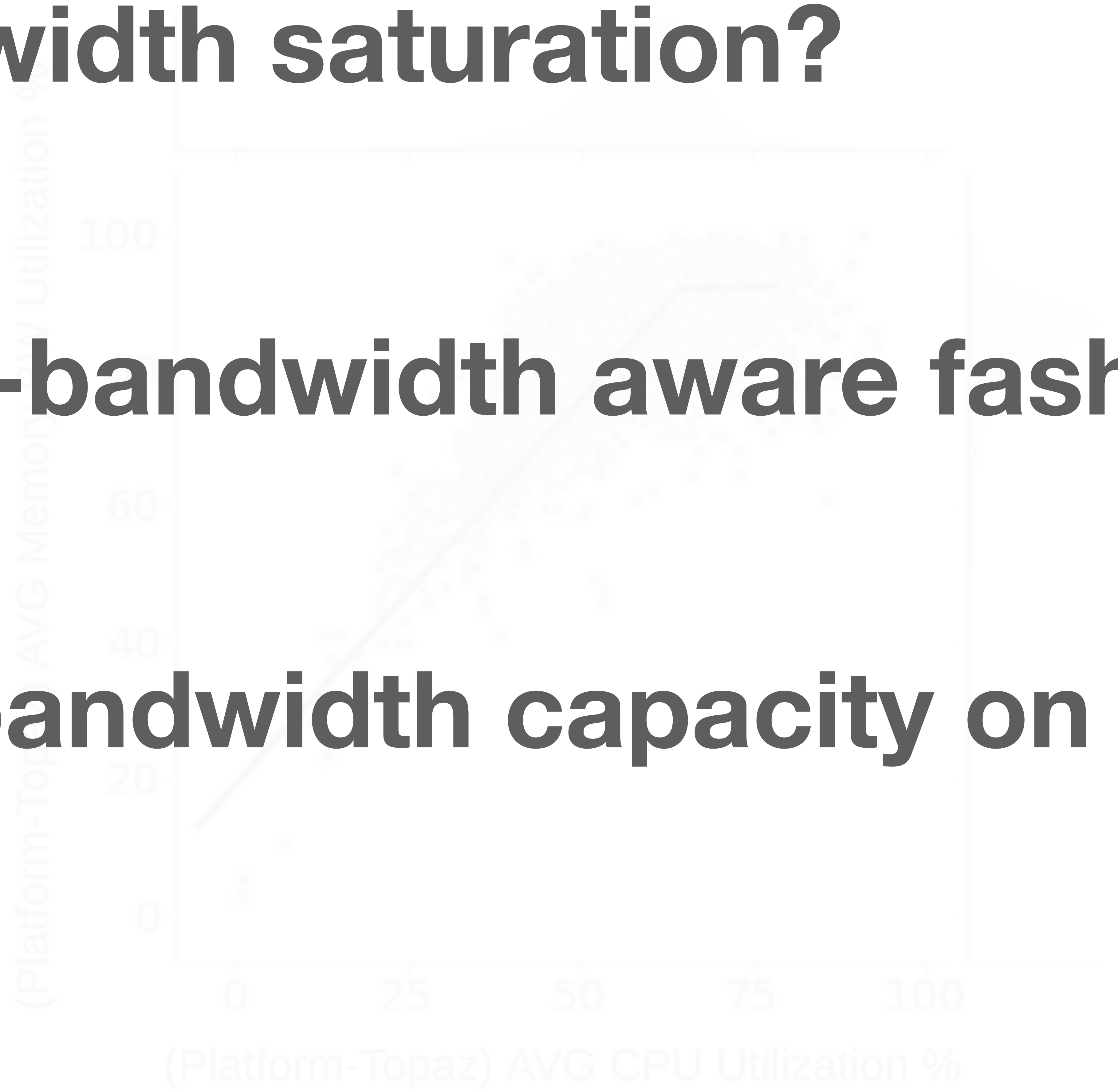


Under-provisioned Memory Bandwidth Capacity @ Google

How to avoid memory bandwidth saturation?

- **Schedule tasks in memory-bandwidth aware fashion**
- **Provision higher memory bandwidth capacity on future servers**



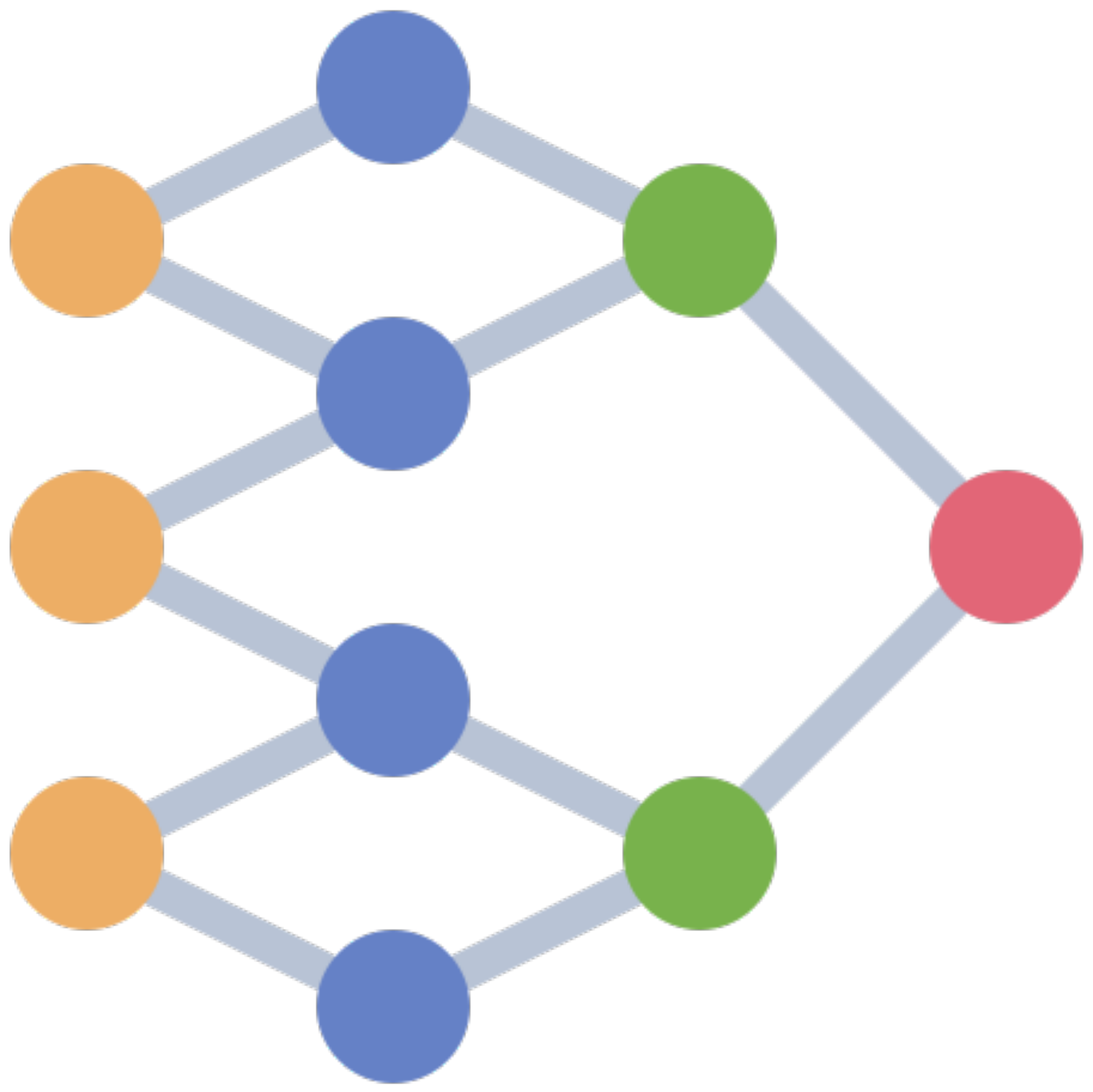


Under-provisioned Memory Bandwidth Capacity @ Google

Machine Learning to predict memory bandwidth usage!

Workload Mix
Resource Utilizations

Server Resource
Capacities



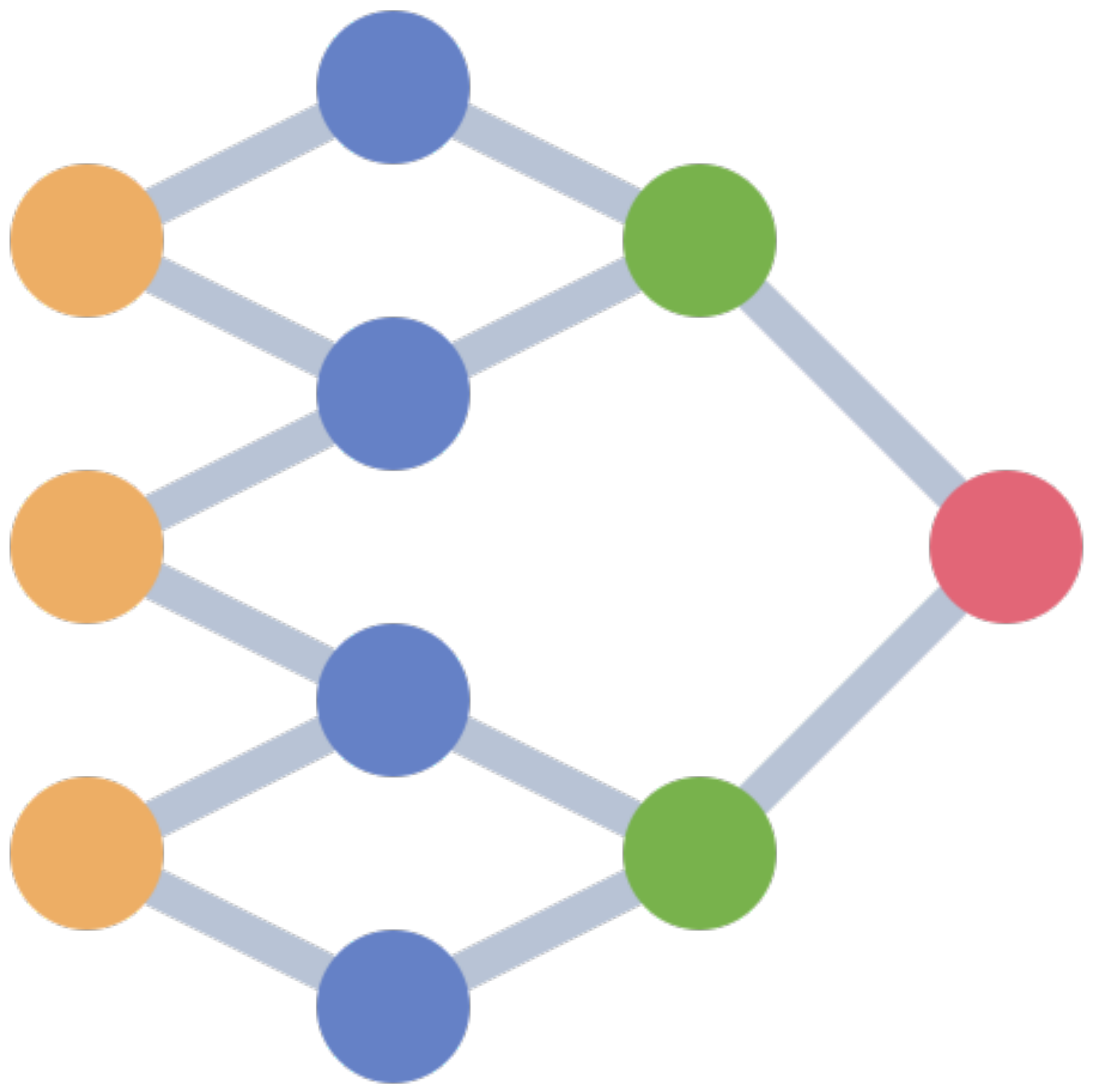
P90 Memory
Bandwidth Usage

Under-provisioned Memory Bandwidth Capacity @ Google

Machine Learning to predict memory bandwidth usage!

Workload Mix
Resource Utilizations

Server Resource
Capacities



P90 Memory
Bandwidth Usage

Obtain prediction; schedule / provision to avoid bandwidth saturation

Vary features to match the spec of your hypothetical server and workload mix

Under-provisioned Memory Bandwidth Capacity @ Google

Training data collected using Google Wide Profiling:

- From 4 server types: Quartz, Topaz, Jade, and Opal
- Consists of ~1 million samples

Testing data:

- From seen server types: Quartz, Topaz, Jade, and Opal
- From unseen server type: Amber

Under-provisioned Memory Bandwidth Capacity @ Google

Model	MAPE on test data from seen server types:
Random Forest	6.6%
Hist. Gradient Boosters	6.9%
Bagging Regressor	6.9%
Neural Network	7.0%
Decision Tree	7.4%
Linear Regression	8.5%

Seems to be a useful model !

Under-provisioned Memory Bandwidth Capacity @ Google

Model	MAPE on test data from seen server types:	MAPE on test data from unseen server types: Amber
Random Forest	6.6%	56%
Hist. Gradient Boosters	6.9%	57%
Bagging Regressor	6.9%	61%
Neural Network	7.0%	94%
Decision Tree	7.4%	57%
Linear Regression	8.5%	73%

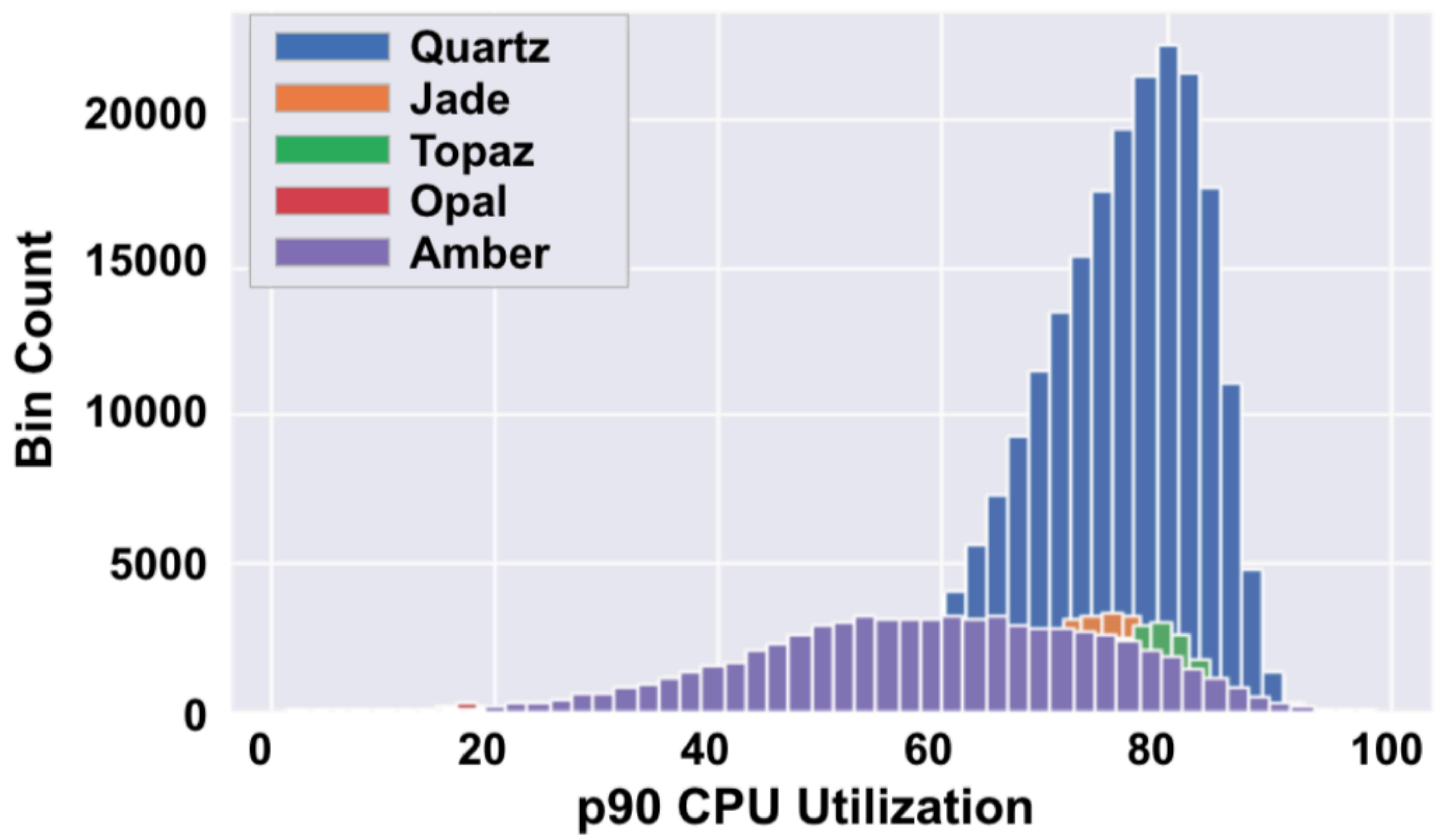
Models do not generalize to unseen server types …

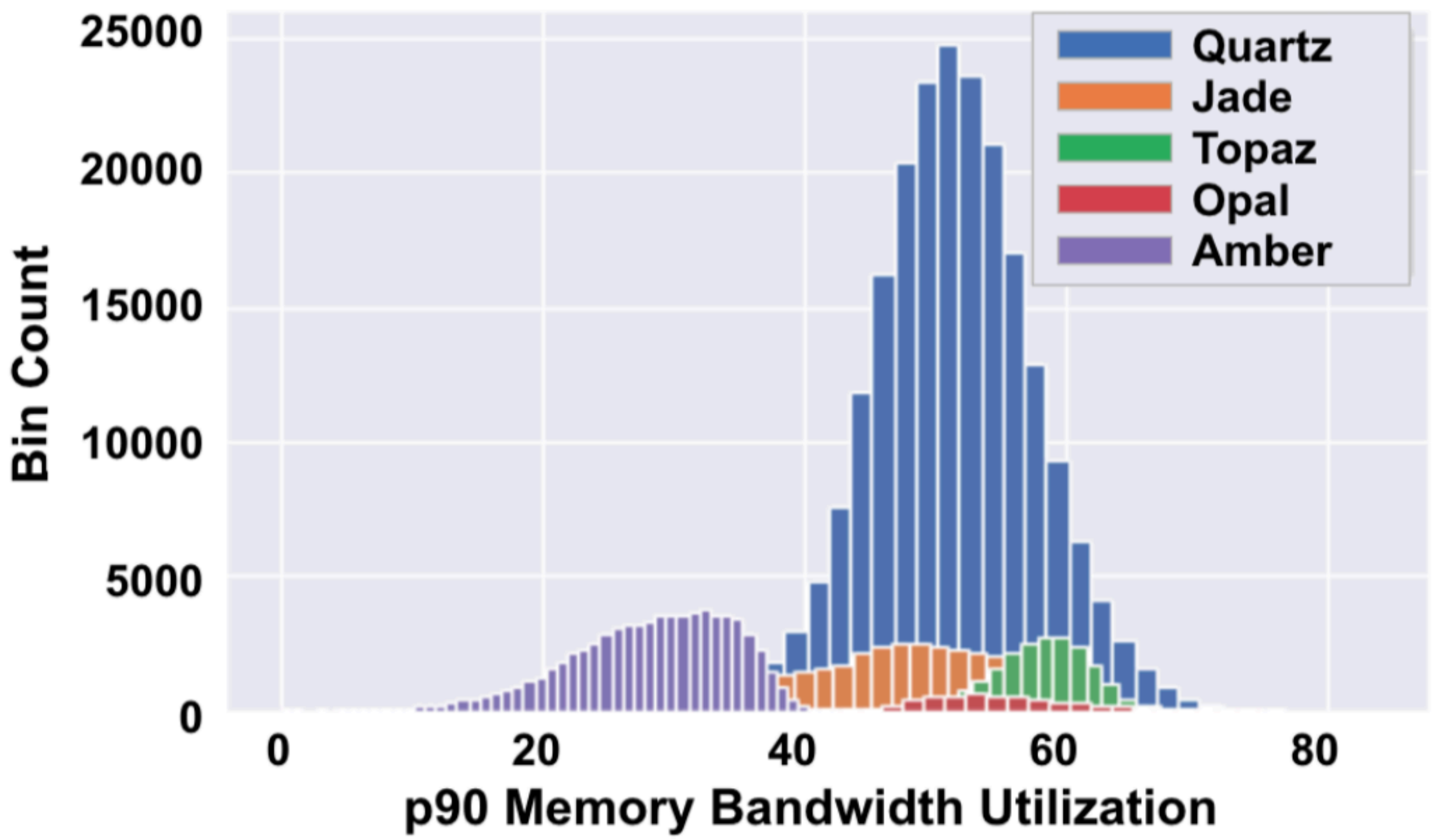
Holds for:

- Simple models like linear regression
- Complex models like neural networks
- Bagging and boosting ensembles

Why is the model not generalizing to unseen servers?

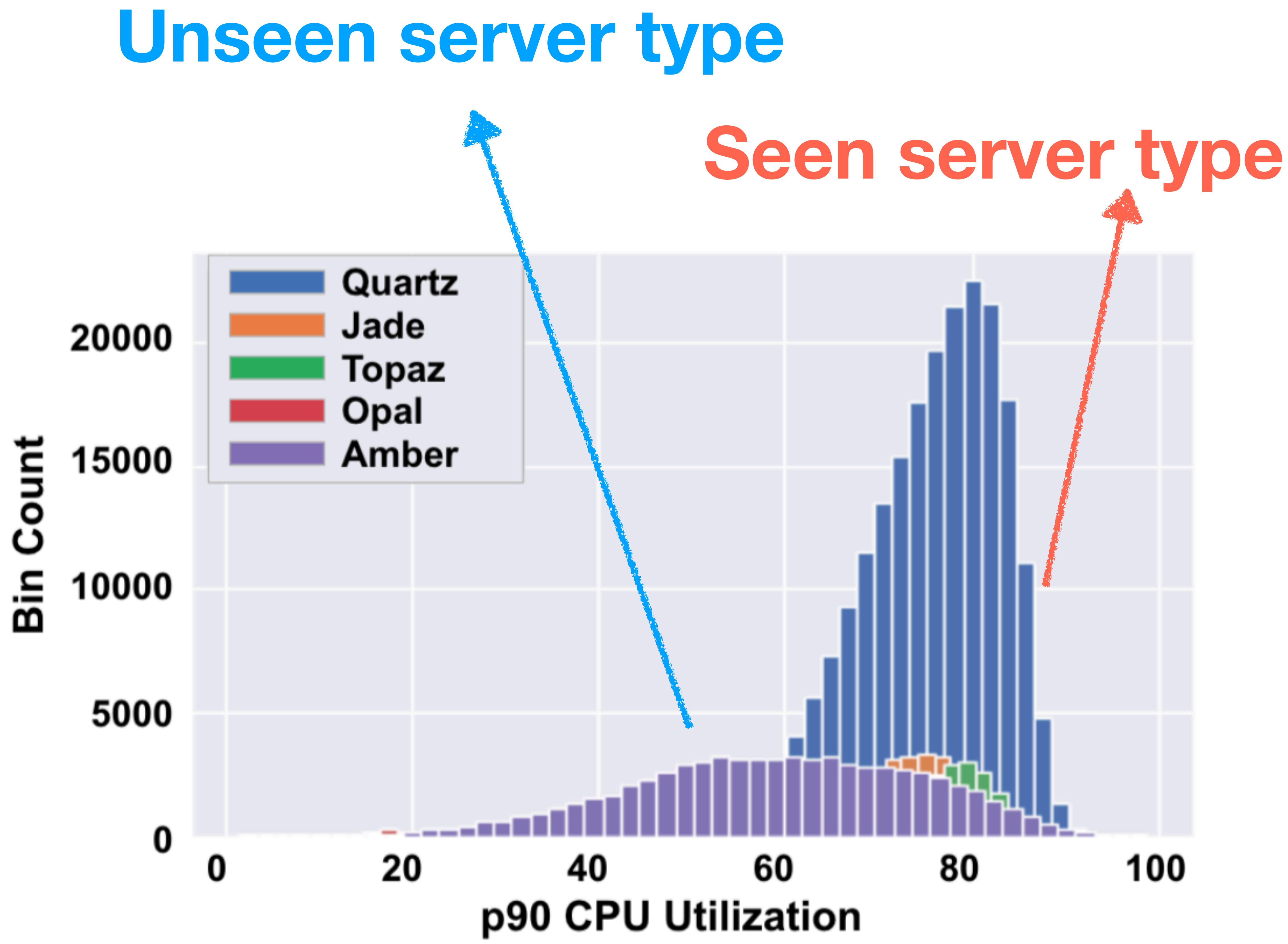
Because their data distributions differ

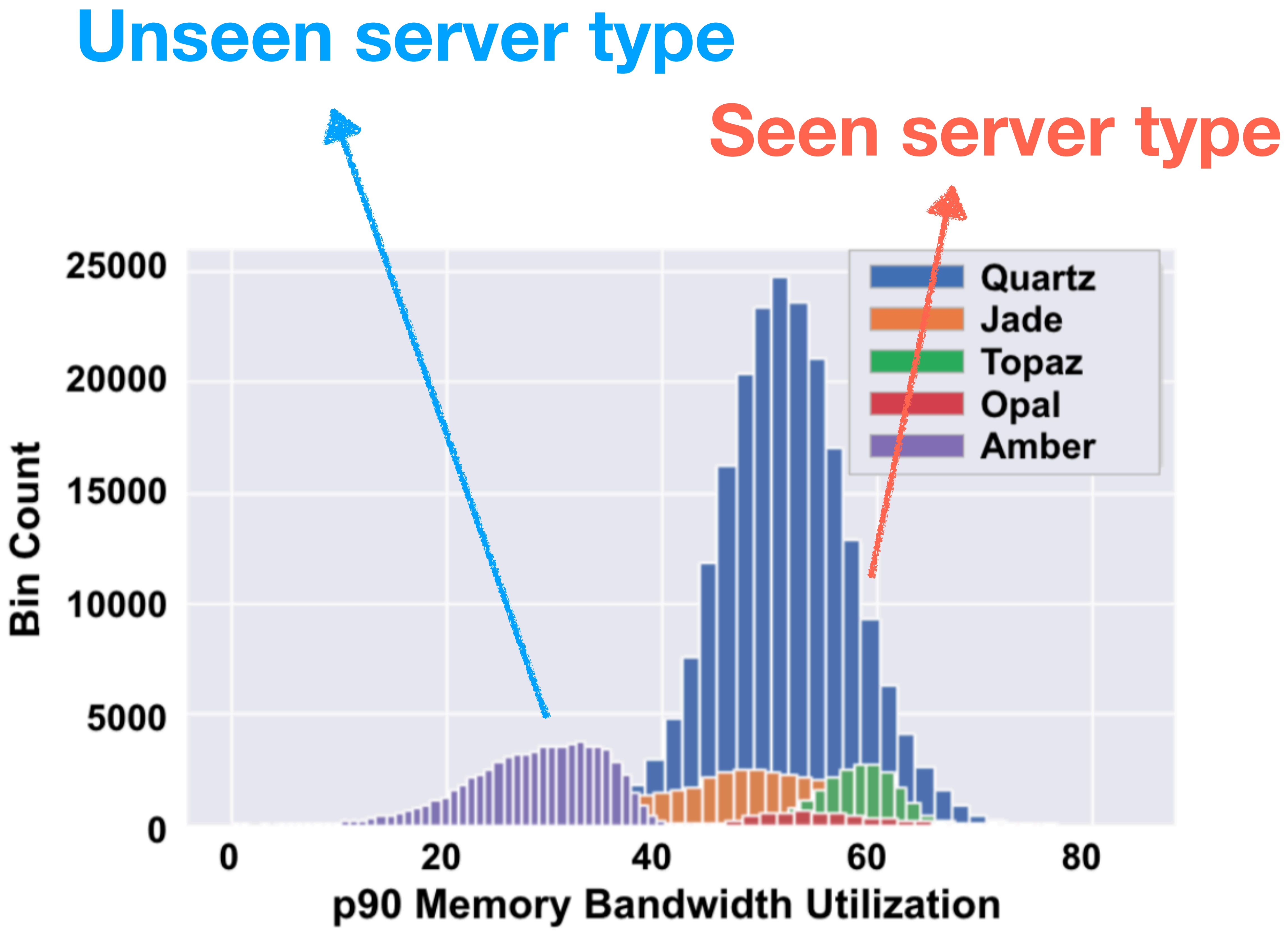




Why is the model not generalizing to unseen servers?

Because their data distributions differ





Models generalize poorly because of distribution changes

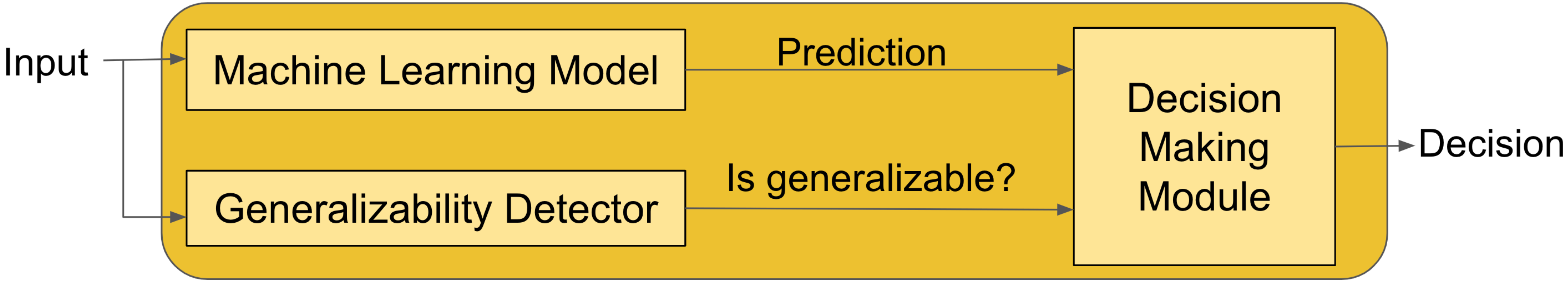
This can happen because of:
- unseen server types
- changing workload mixes
- and many more reasons …

Impossible to have a model with perfect generalizability

Always useful to have a mechanism to deal with cases when model does not generalize

Avoid mispredictions caused by poor generalizability

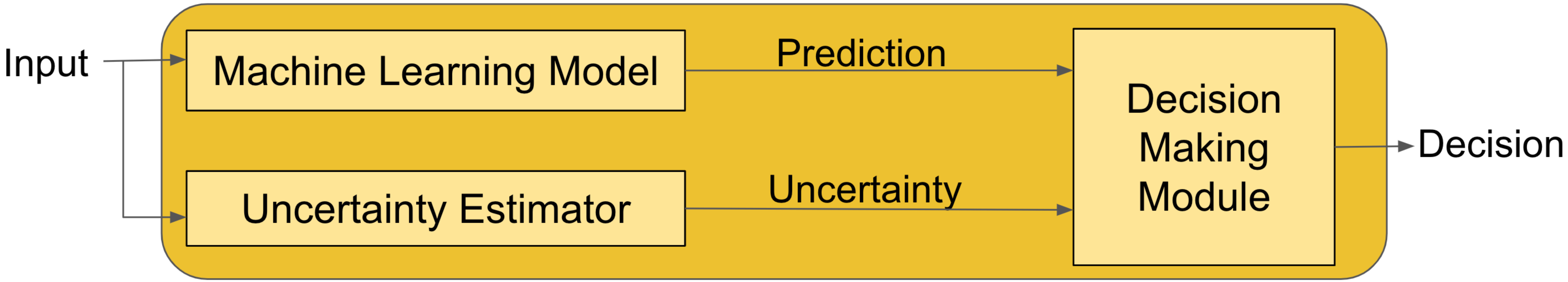
Proactively check the model's generalizability



Avoid mispredictions caused by poor generalizability

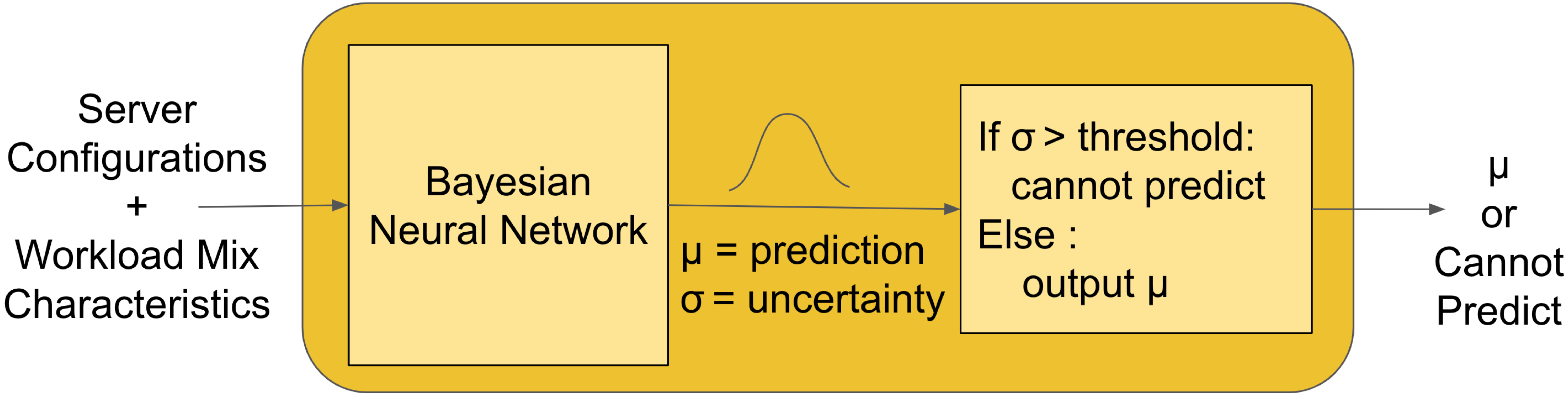
Proactively check the model's generalizability

Use proxies like uncertainty



"Uncertainty-aware" memory bandwidth prediction

- Uses Bayesian Neural Network for prediction and uncertainty estimation



“Uncertainty-aware” memory bandwidth prediction

Bayesian neural network trained on data from

Quartz, Jade, Topaz, and Opal

Test data from	MAPE	Uncertainty
Seen server types: Quartz, Jade, Topaz, Opal	8.7%	
Unseen server type: Amber	47.7%	

“Uncertainty-aware” memory bandwidth prediction

Bayesian neural network trained on data from

Quartz, Jade, Topaz, and Opal

Test data from	MAPE	Uncertainty
Seen server types: Quartz, Jade, Topaz, Opal	8.7%	1.2
Unseen server type: Amber	47.7%	

“Uncertainty-aware” memory bandwidth prediction

Bayesian neural network trained on data from

Quartz, Jade, Topaz, and Opal

Test data from	MAPE	Uncertainty
Seen server types: Quartz, Jade, Topaz, Opal	8.7%	1.2
Unseen server type: Amber	47.7%	15.6

Is uncertainty always high for unseen server types?

Is uncertainty always high for unseen server types?

Bayesian neural network trained on data from Quartz and Topaz

Is uncertainty always high for unseen server types?

Bayesian neural network trained on data from Quartz and Topaz

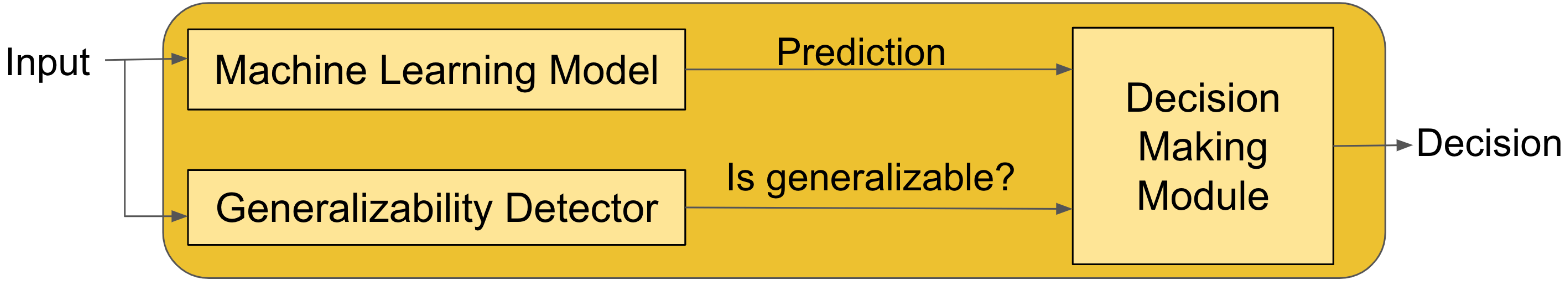
Test data from	MAPE	Uncertainty
Seen server types: Quartz, Topaz	8.0%	1.0
Unseen server type: Amber	56.0%	15.0

Is uncertainty always high for unseen server types?

Bayesian neural network trained on data from Quartz and Topaz

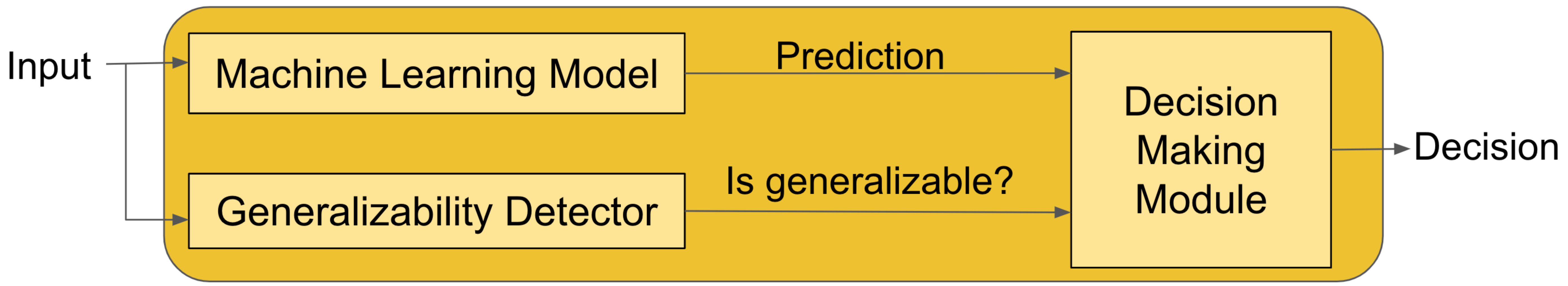
Test data from	MAPE	Uncertainty
Seen server types: Quartz, Topaz	8.0%	1.0
Unseen server type: Amber	56.0%	15.0
Unseen Server Type: Jade	12.8%	1.9
Unseen Server Type: Opal	11.8%	3.4

Workflow helps you proactively decide when to use / not use the prediction of machine learning model

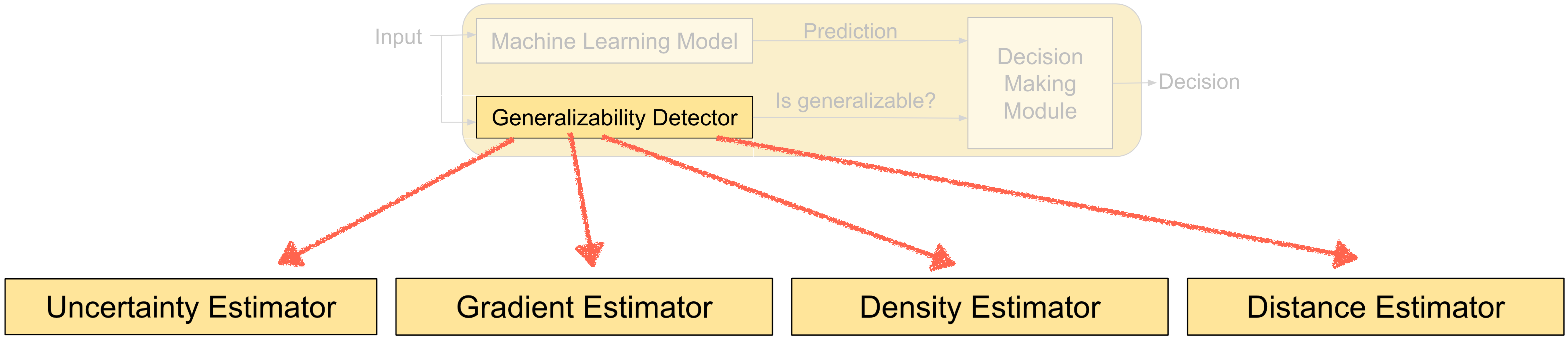


And improves reliability of machine learning for systems

Towards reliable machine learning for systems …

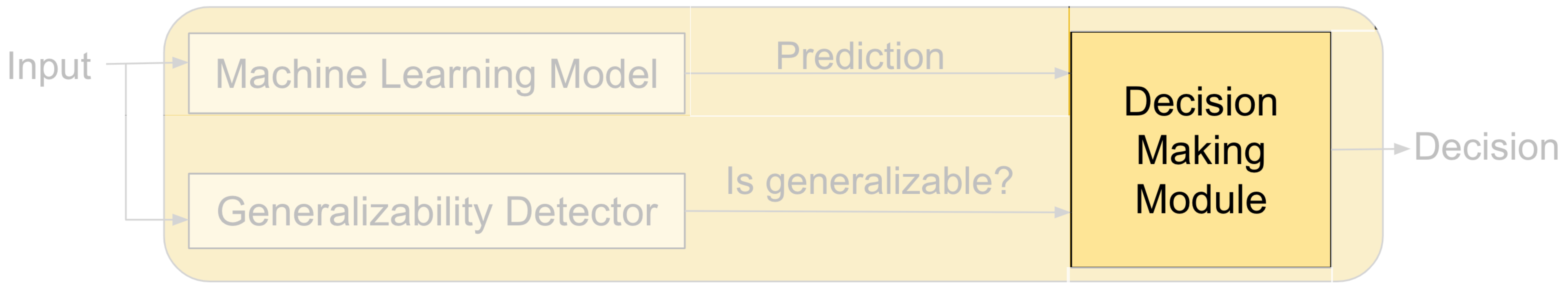


Towards reliable machine learning for systems …



Which generalizability detector to use?

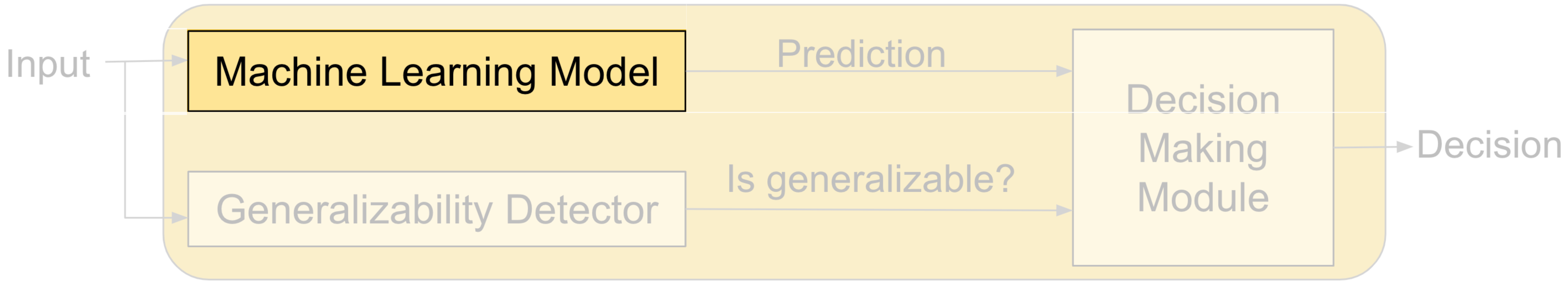
Towards reliable machine learning for systems …



What to do when not using the model's prediction?

Maybe fall back on traditional heuristics

Towards reliable machine learning for systems …



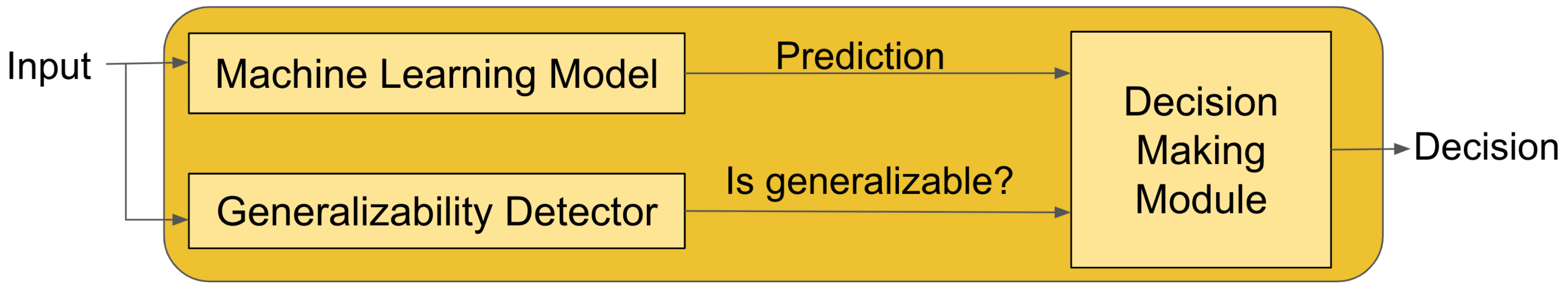
How to make the model more generalizable?

Causal Modelling

Meta Learning

Curriculum Learning

Towards reliable machine learning for systems …



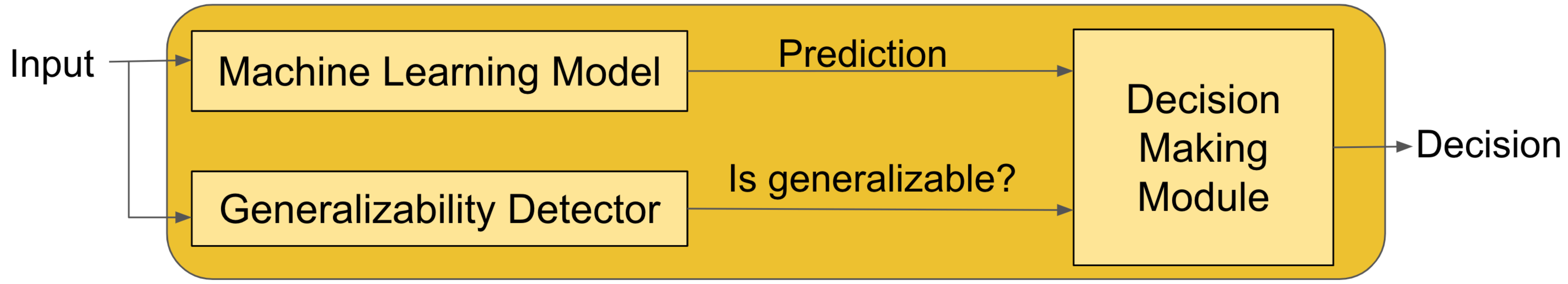
Helps us answer "***When*** a model mispredicts?"

But not "***Why*** a model mispredicts?"

Need to interpret machine learning models

Key Takeaways

- Poor generalizability of ML models makes them unreliable. This unreliability hinders the industrial adoption of ML for Systems proposals.
- To improve reliability, avoid mispredictions by proactively measuring proxies of generalizability to guide model usage. Example: Workflow helps decide on which unseen server we can use the model's predictions.



Test data from	MAPE	Uncertainty
Unseen server type: Amber	56.0%	15.0
Unseen Server Type: Jade	12.8%	1.9