ModelKeeper

Accelerating DNN Training via Automated Training Warmup

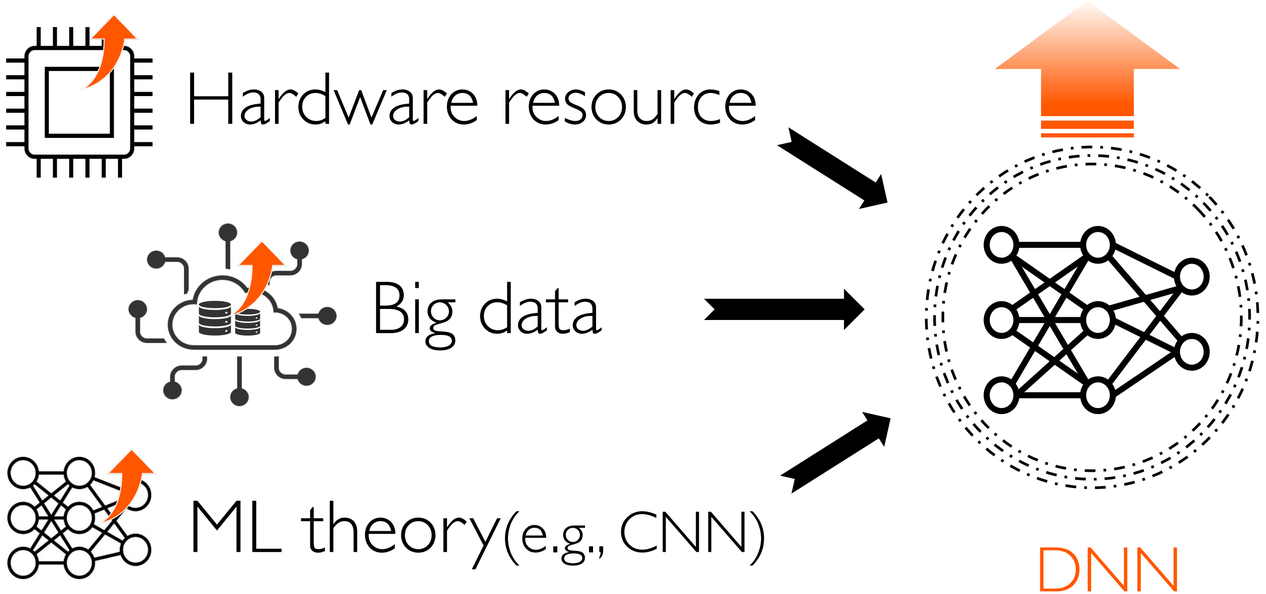
Fan Lai, Yinwei Dai,

Harsha V. Madhyastha, Mosharaf Chowdhury

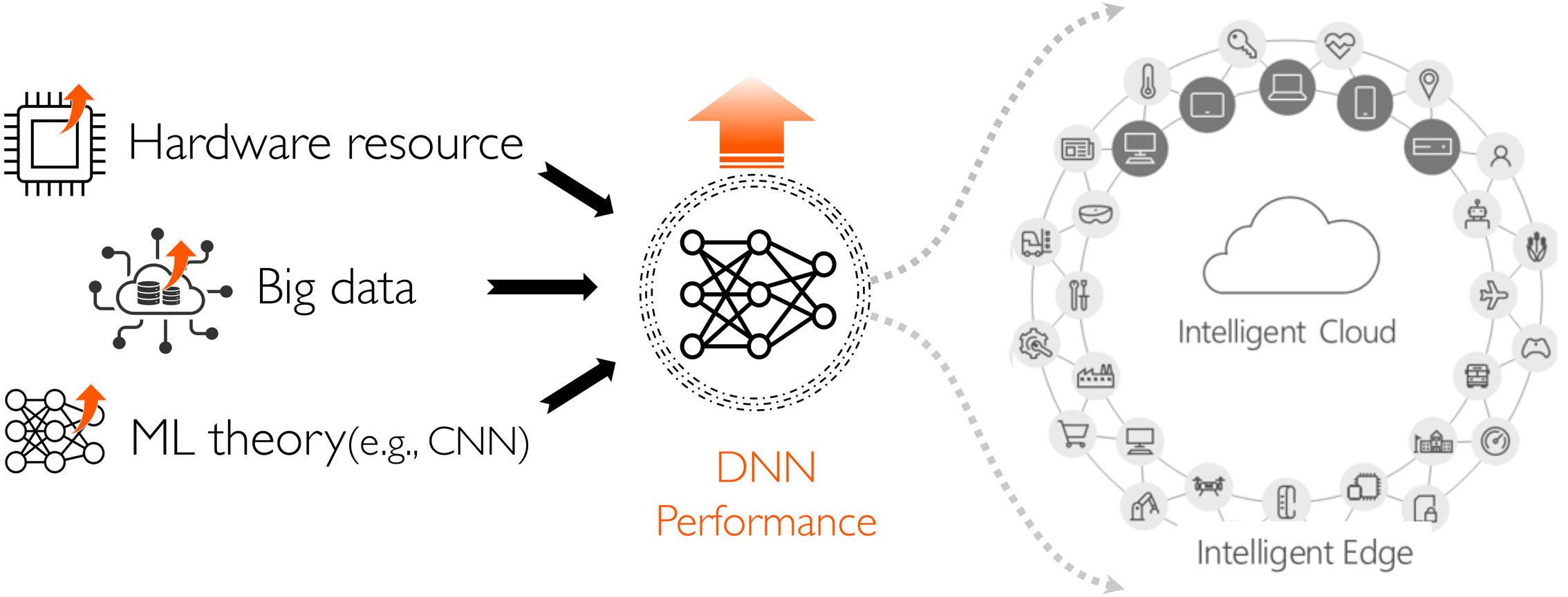
SymbioticLab

University of Michigan

Deep Neural Networks Become Prevalent



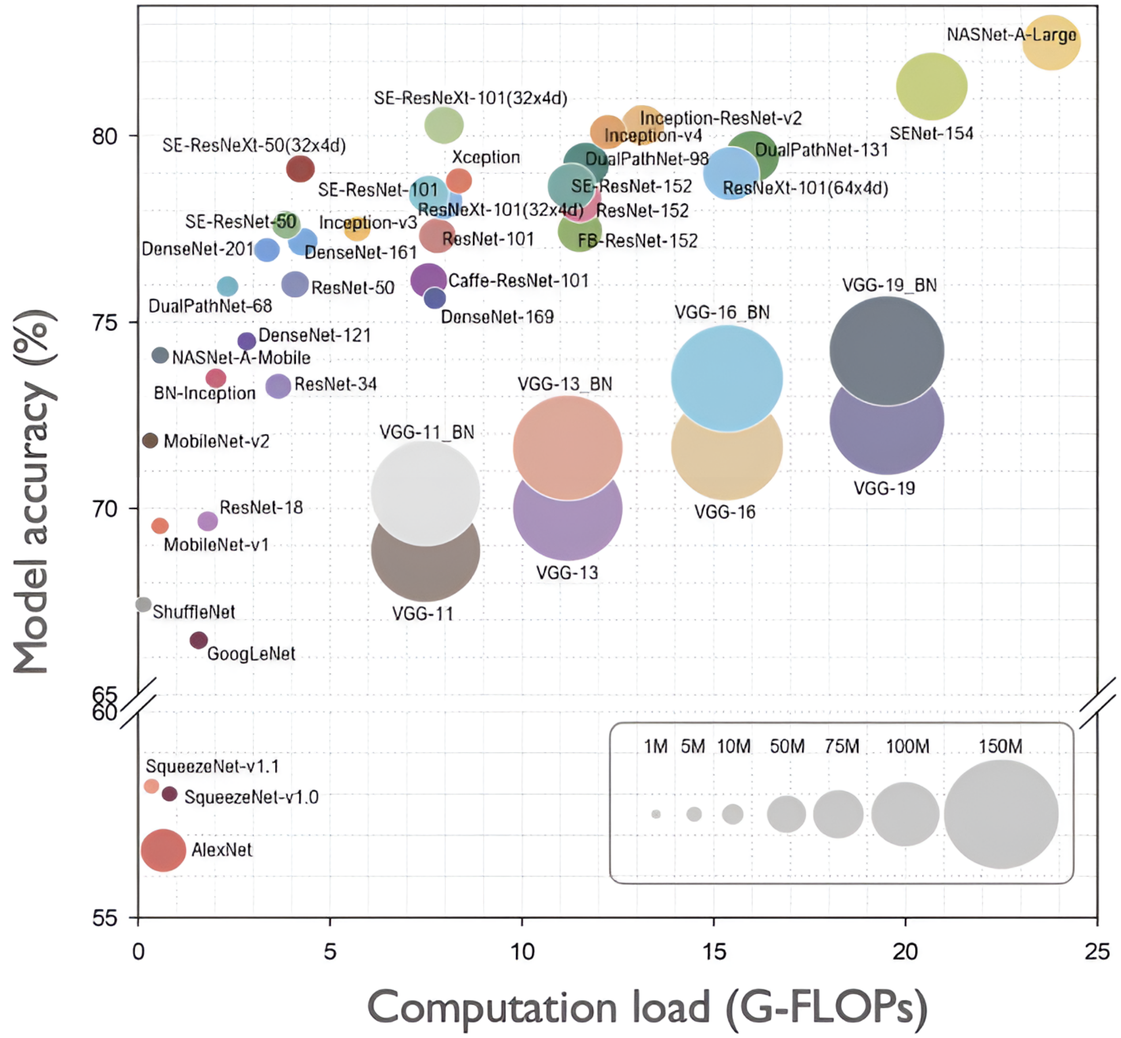
Deep Neural Networks Become Prevalent



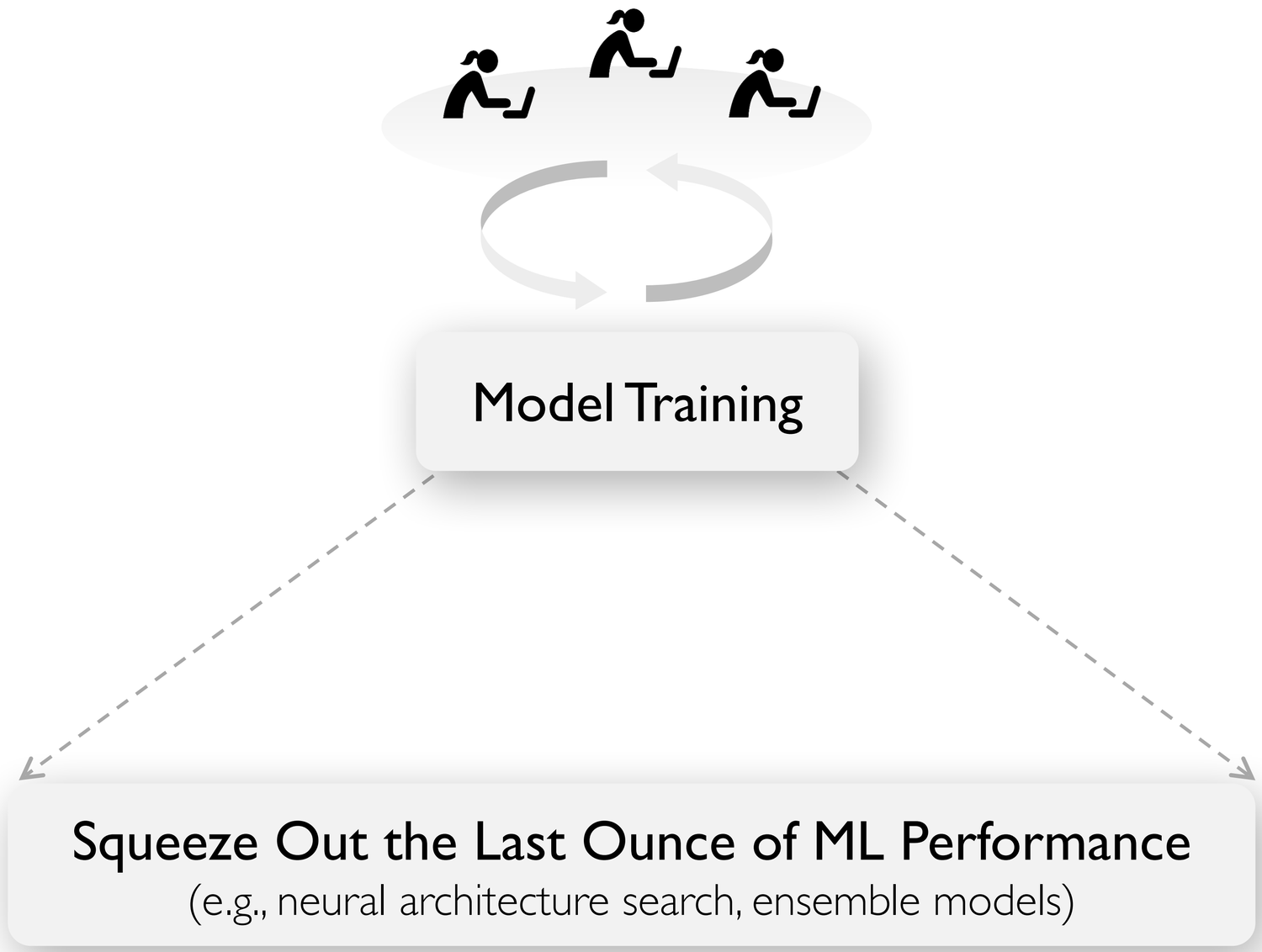
DNN Training Demand is Skyrocketing

DNN Training Demand is Skyrocketing

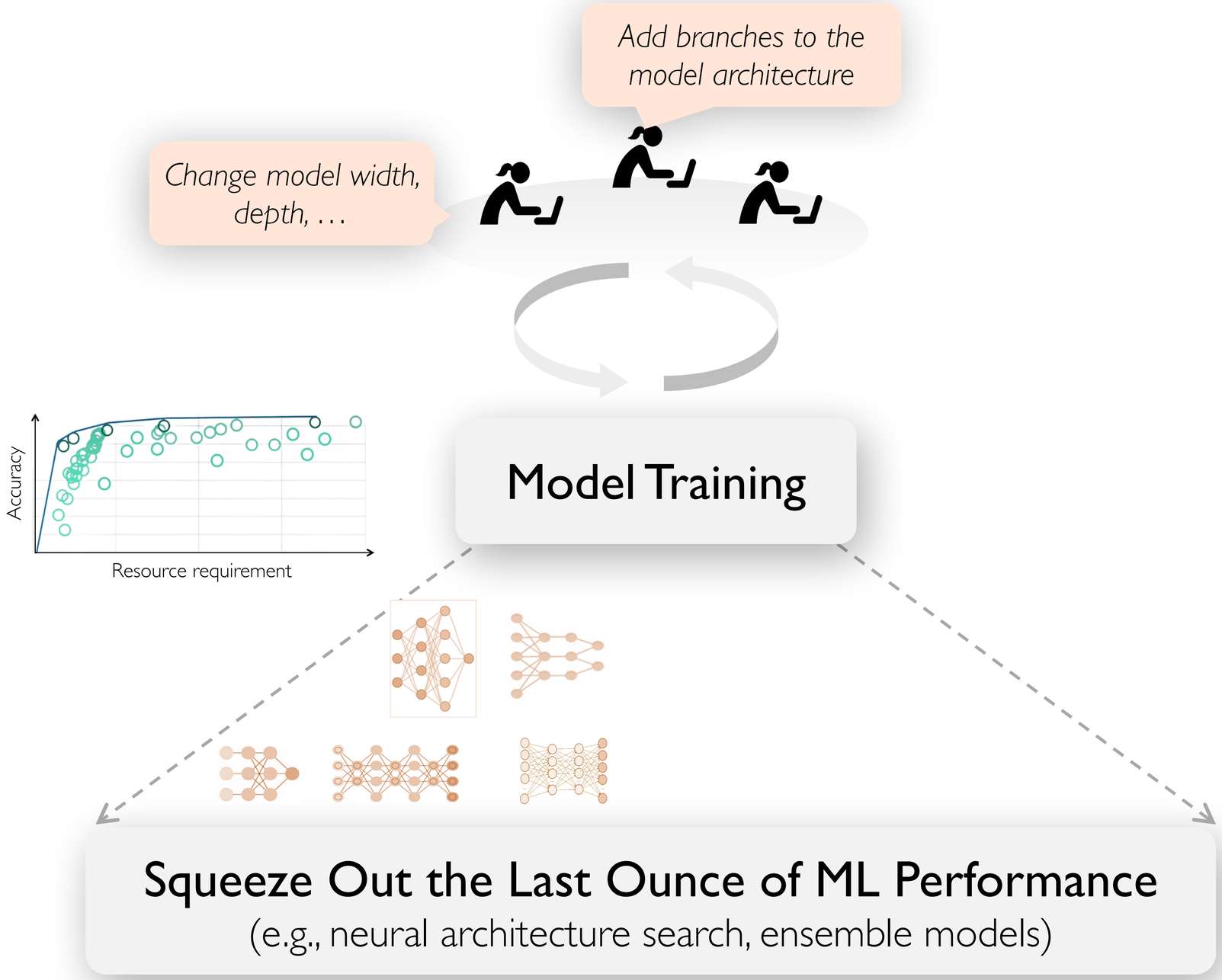
Vast design space of DNNs



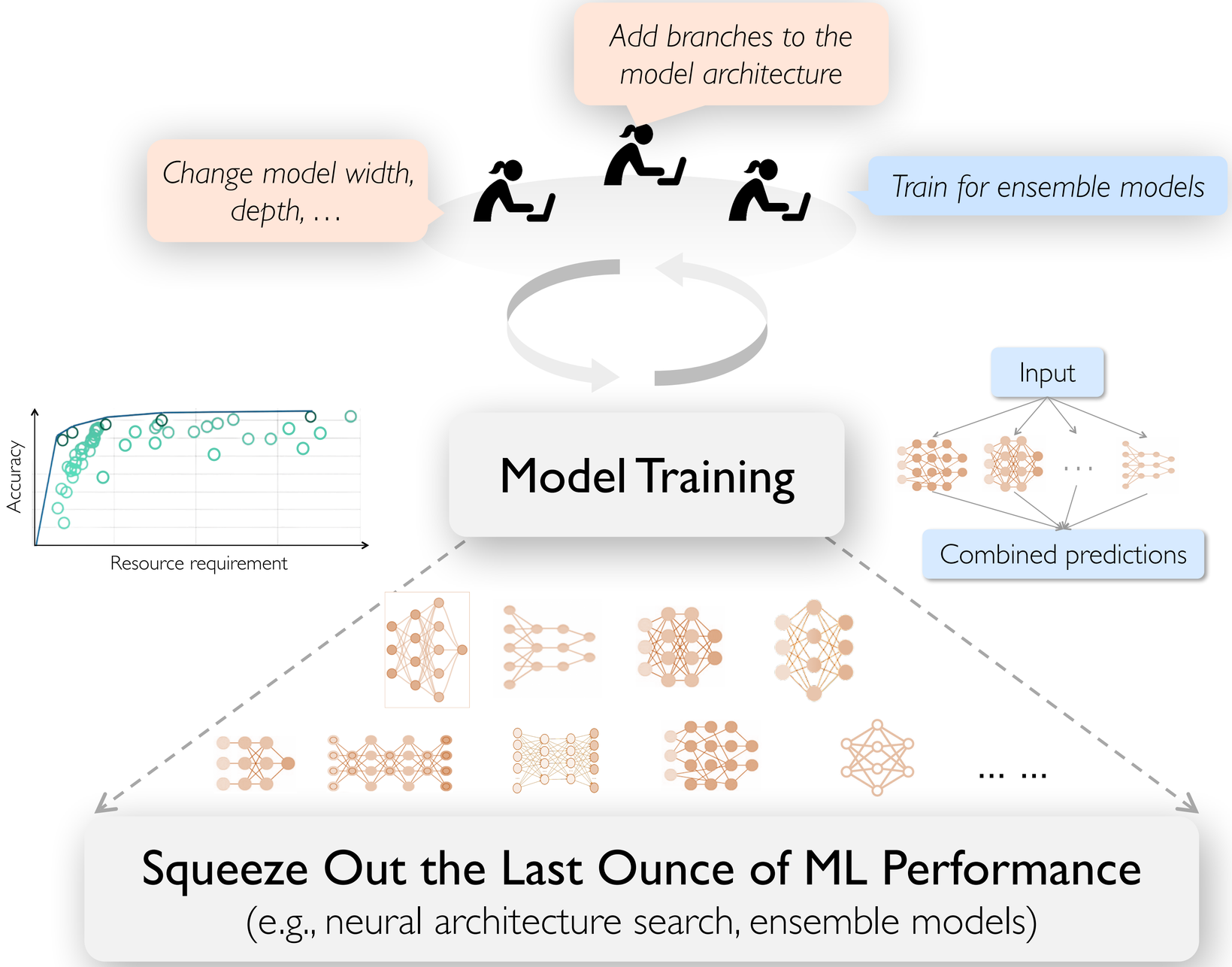
DNN Training Demand is Skyrocketing



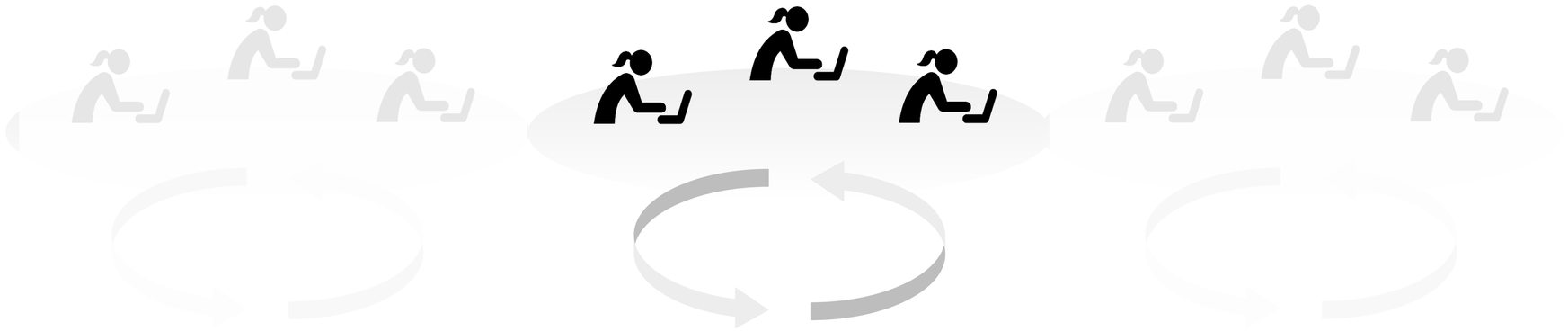
DNN Training Demand is Skyrocketing



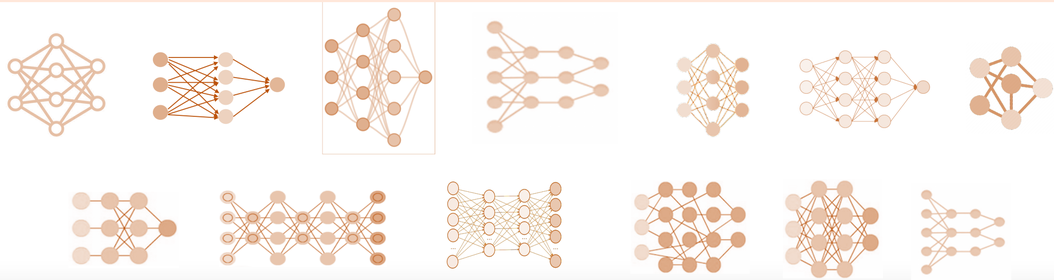
DNN Training Demand is Skyrocketing



DNN Training Demand is Skyrocketing



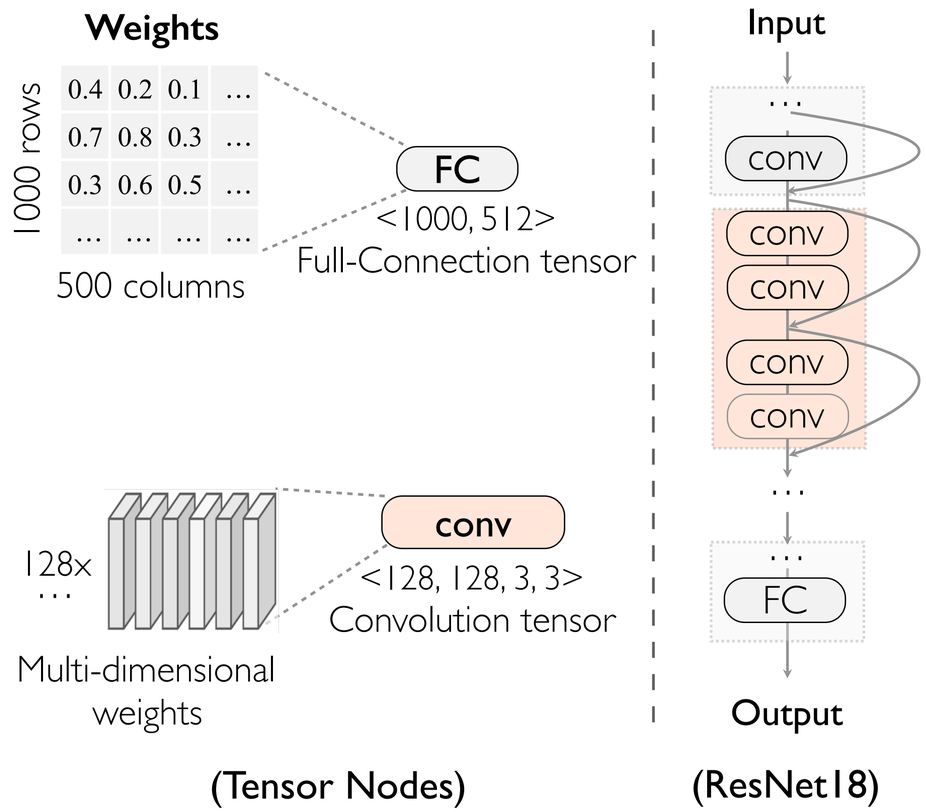
Large shared clusters run thousands of training jobs every day.
Can we *automatically* repurpose trained models for new jobs?



Squeeze Out the Last Ounce of ML Performance
(e.g., neural architecture search, ensemble models)

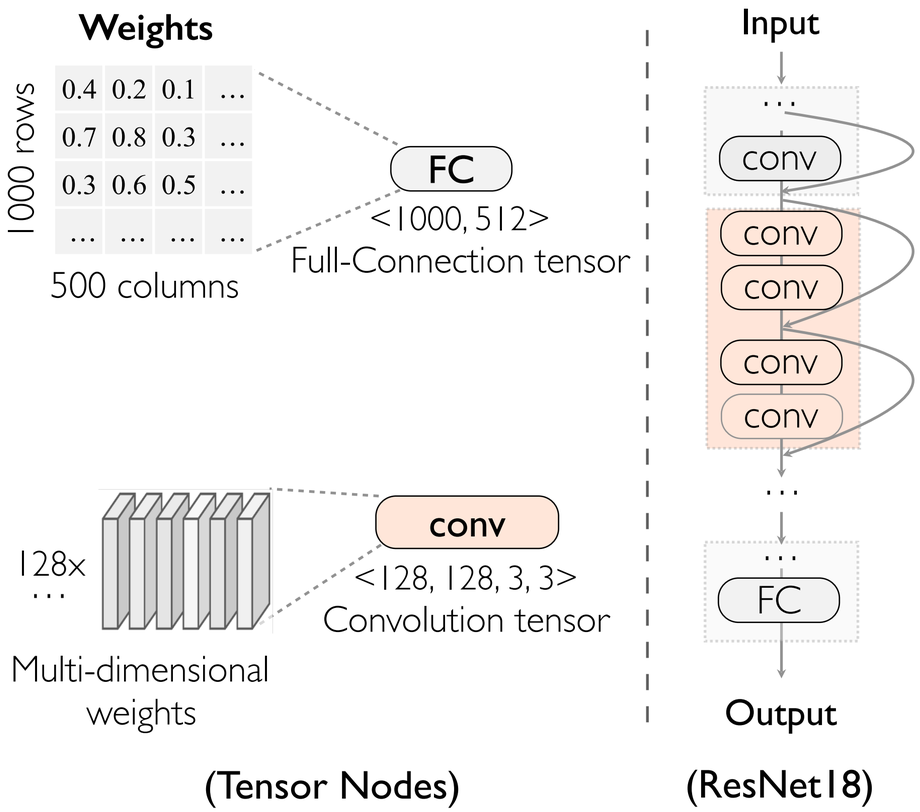
Opportunity for Saving Training Execution

- **ML model is a graph of tensors**



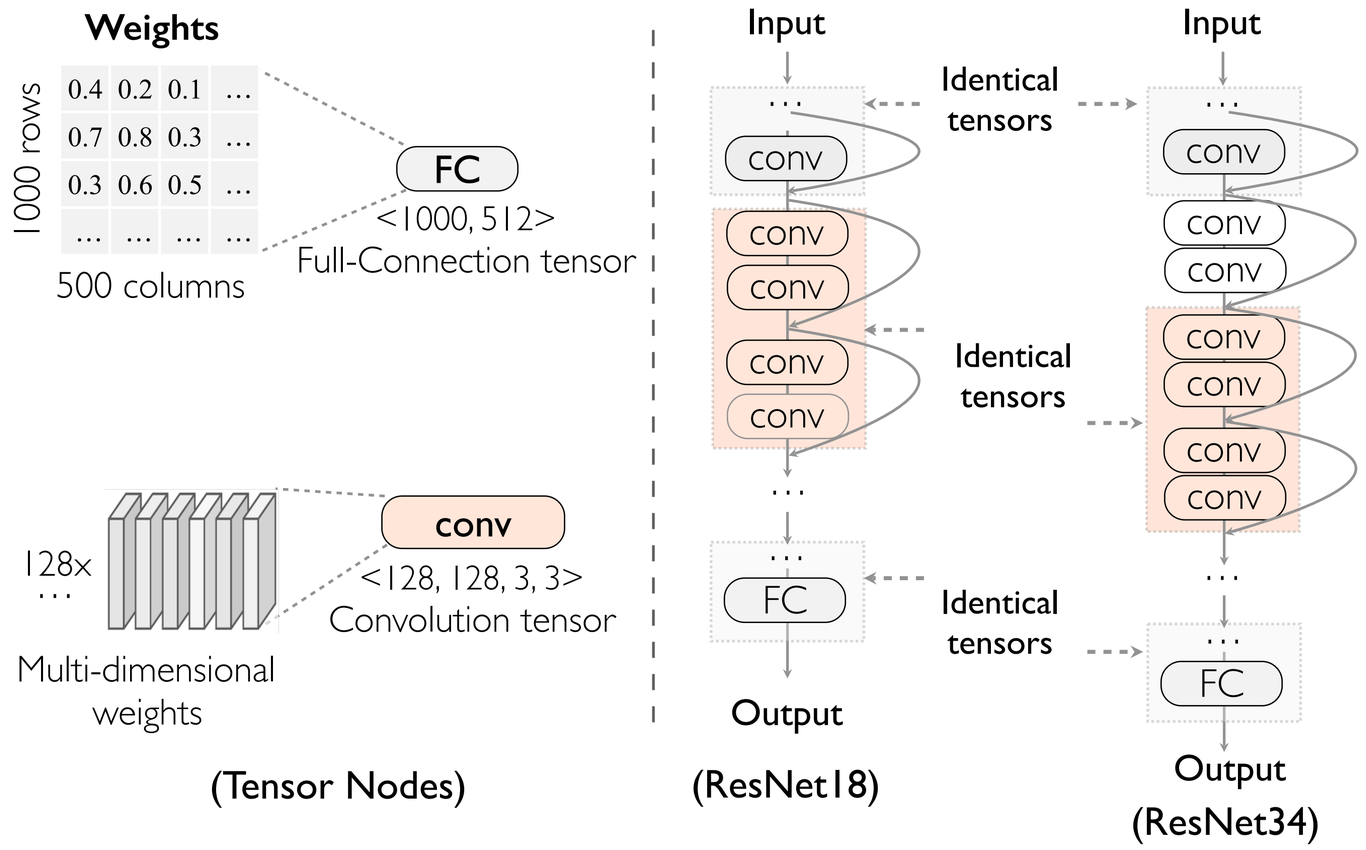
Opportunity for Saving Training Execution

- **ML model is a graph of tensors; training searches for best weight values**



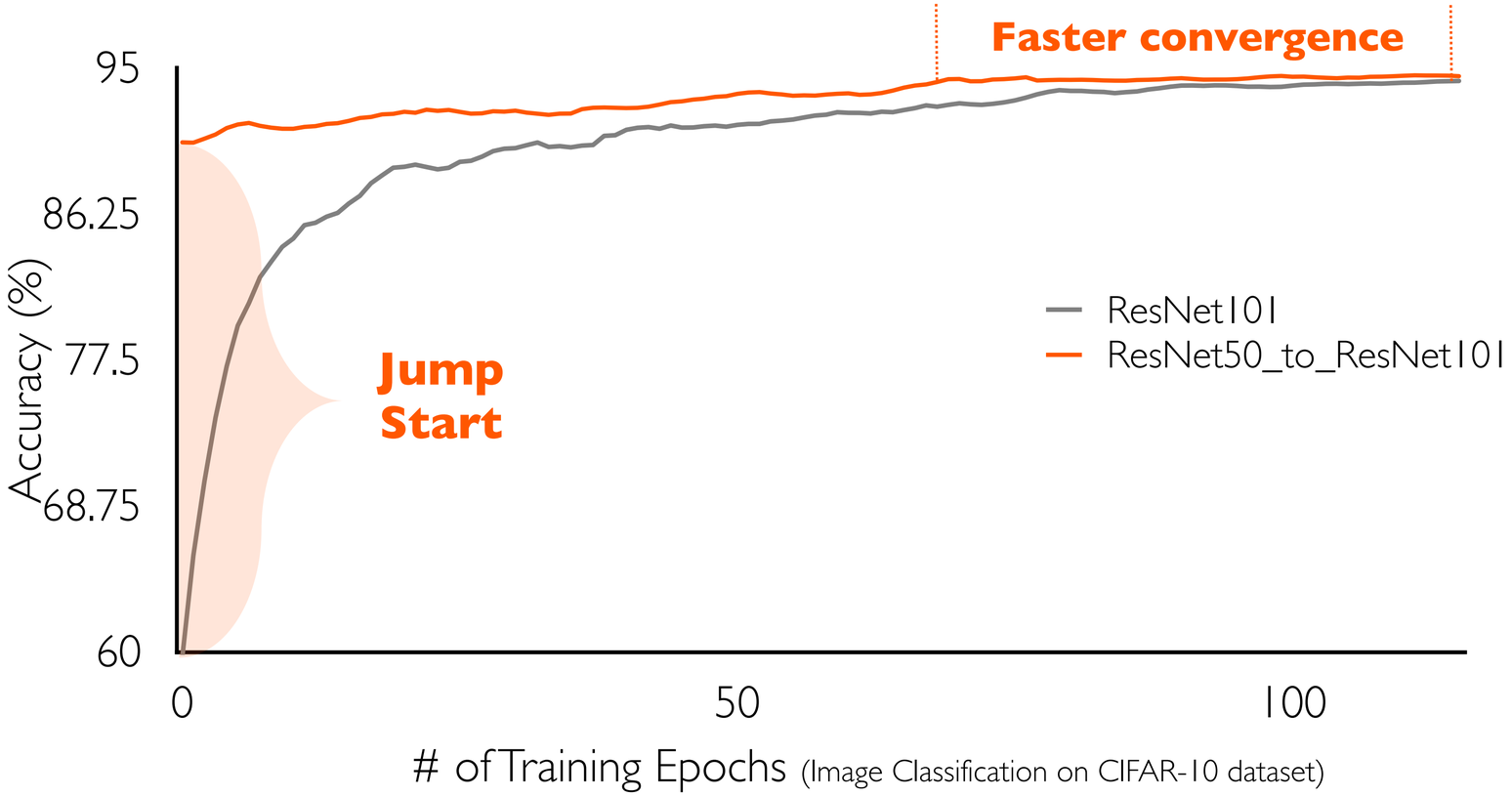
Opportunity for Saving Training Execution

- **ML model is a graph of tensors; training searches for best weight values**



Opportunity for Saving Training Execution

- **ML model is a graph of tensors; training searches for best weight values**
  - Weight transformation can jump start training



Opportunity for Saving Training Execution

- **ML model is a graph of tensors; training searches for best weight values**
  - Weight transformation can jump start training

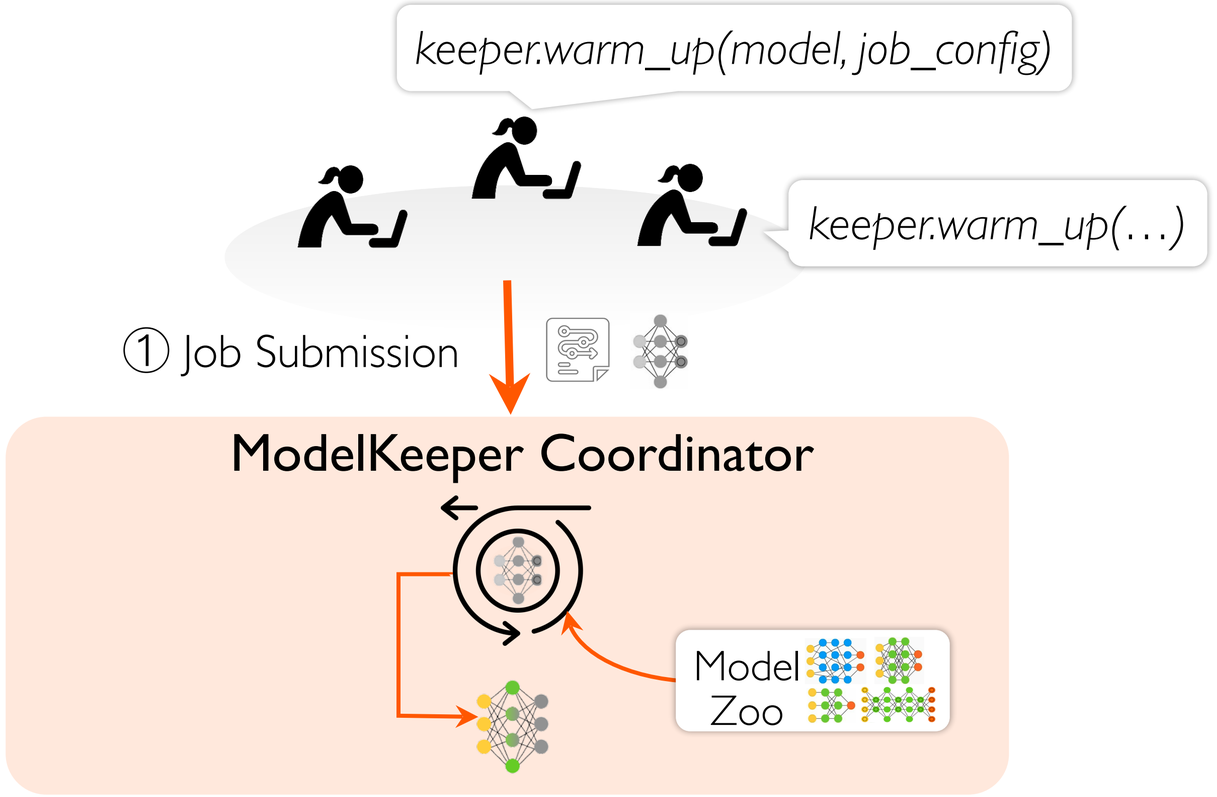
- **Large clusters often contain jobs w/ similar architectures**[1,2]
  - ~40% models have architecturally similar counterparts

- **Automated warmup w/o overhead is a must**
  - Due to too many jobs, varying user expertise, architectures, etc.

[1] "MLaaS in the Wild: Workload Analysis and Scheduling in Large-Scale Heterogeneous GPU Clusters", HKUST and Alibaba, NSDI'22
[2] "Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective", Facebook, HPCA'18

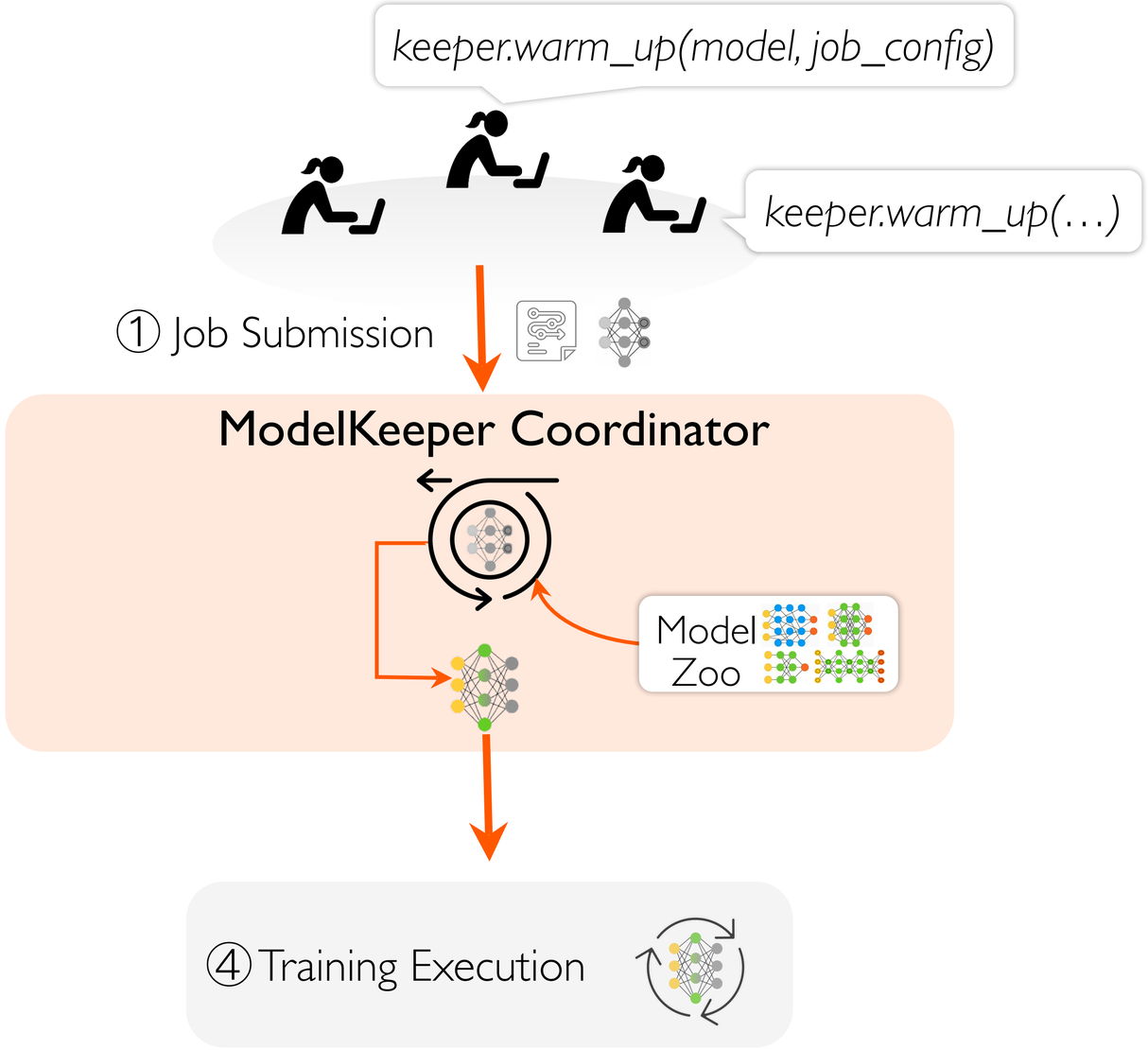
ModelKeeper

Automated Training Warmup System



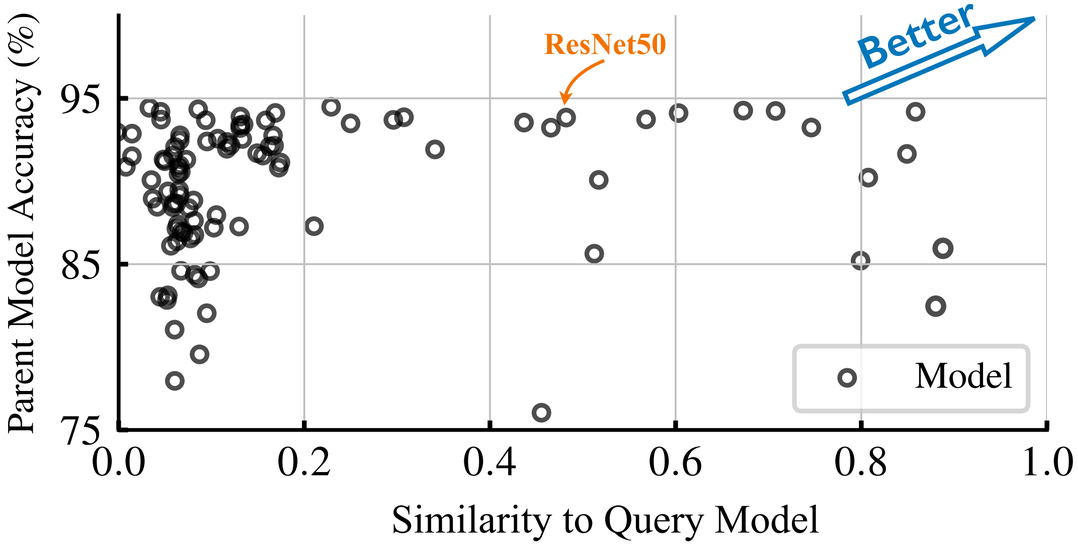
ModelKeeper

Automated Training Warmup System



Effectiveness of Training Warmup

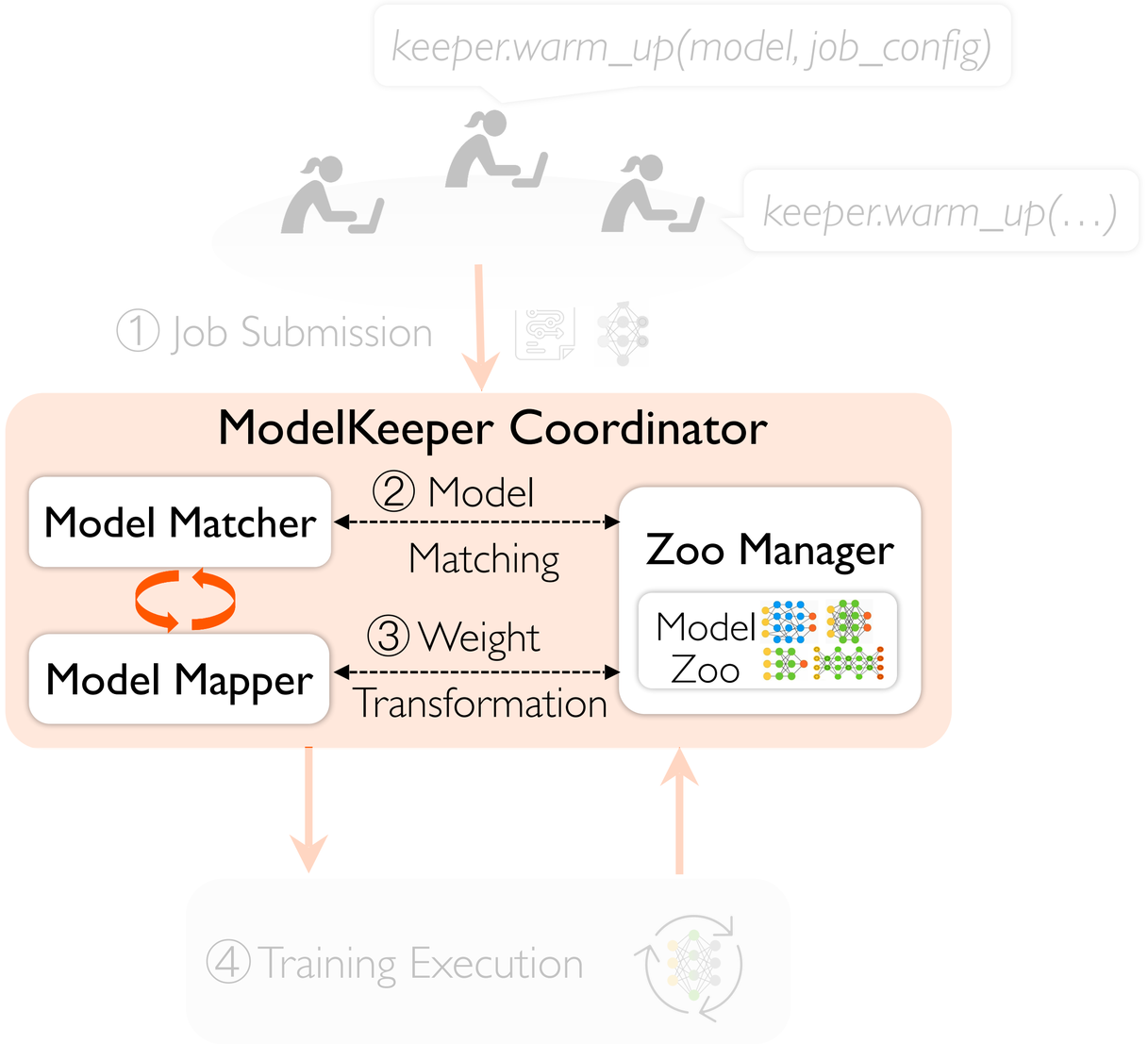
- **Model architectural similarity**
  - Determine how many can transform
- **Parent model accuracy**
  - Determine how beneficial to transform



Models are heterogeneous (ResNet-101 Example).

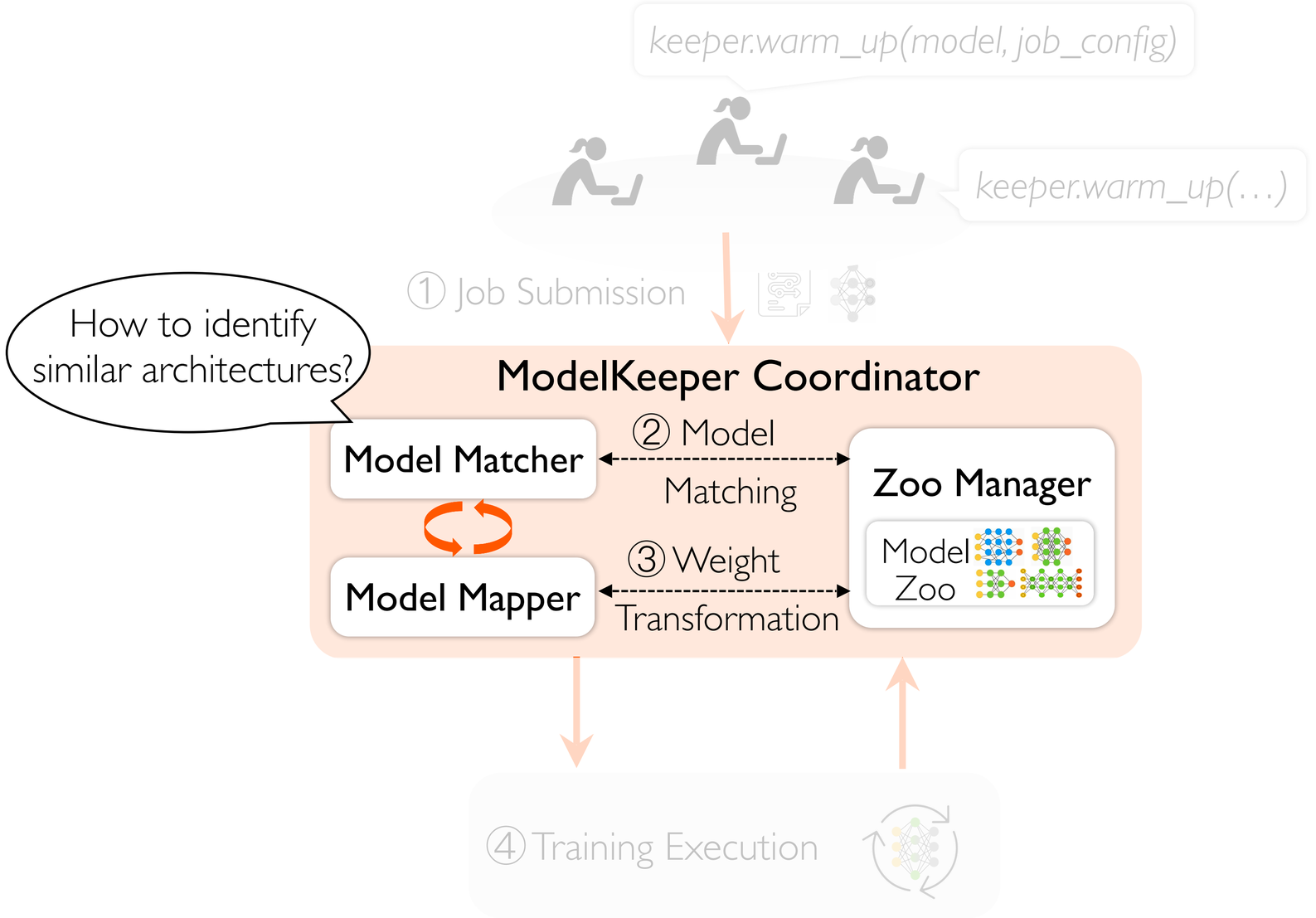
ModelKeeper

Automated Training Warmup System



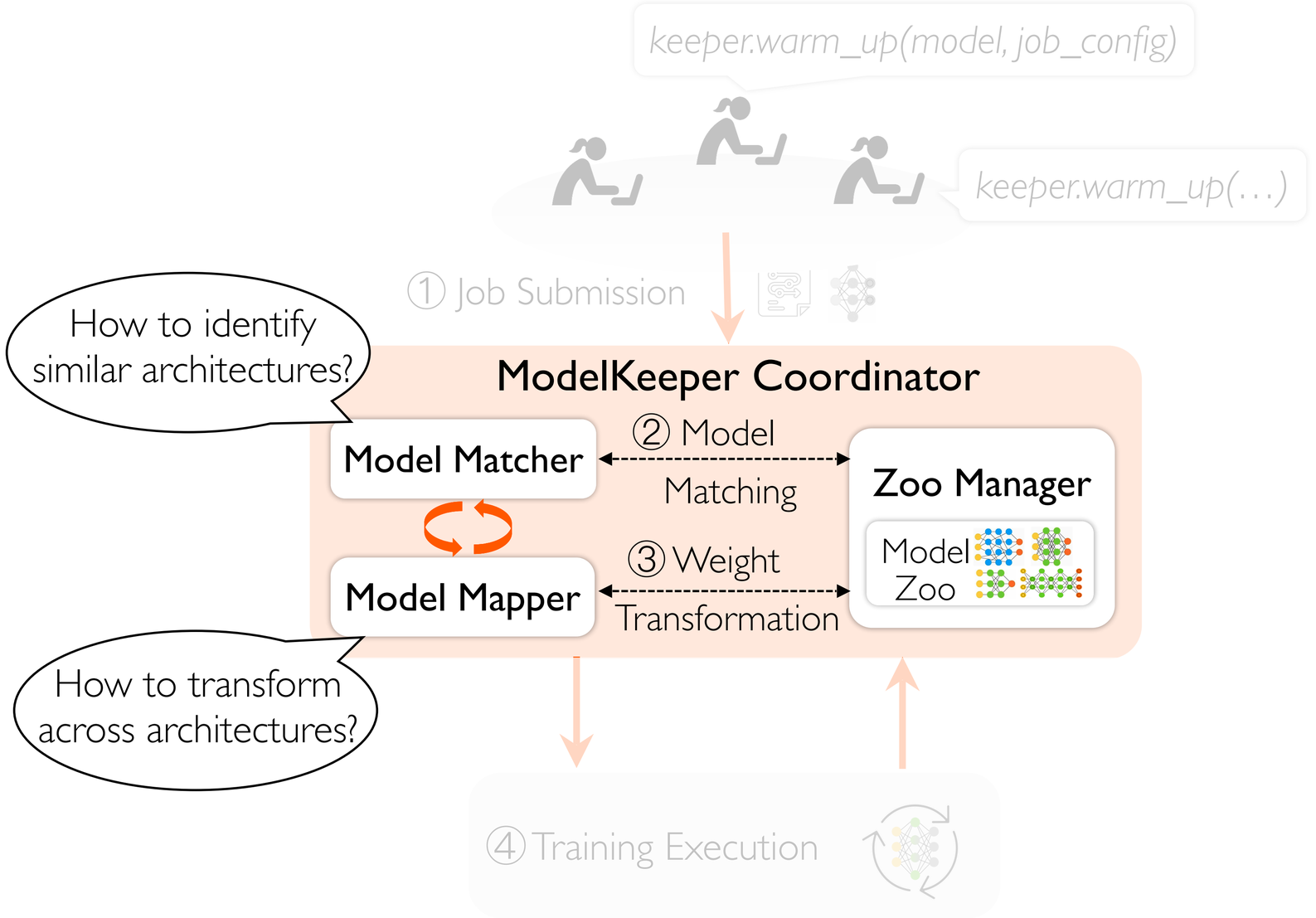
ModelKeeper

Automated Training Warmup System



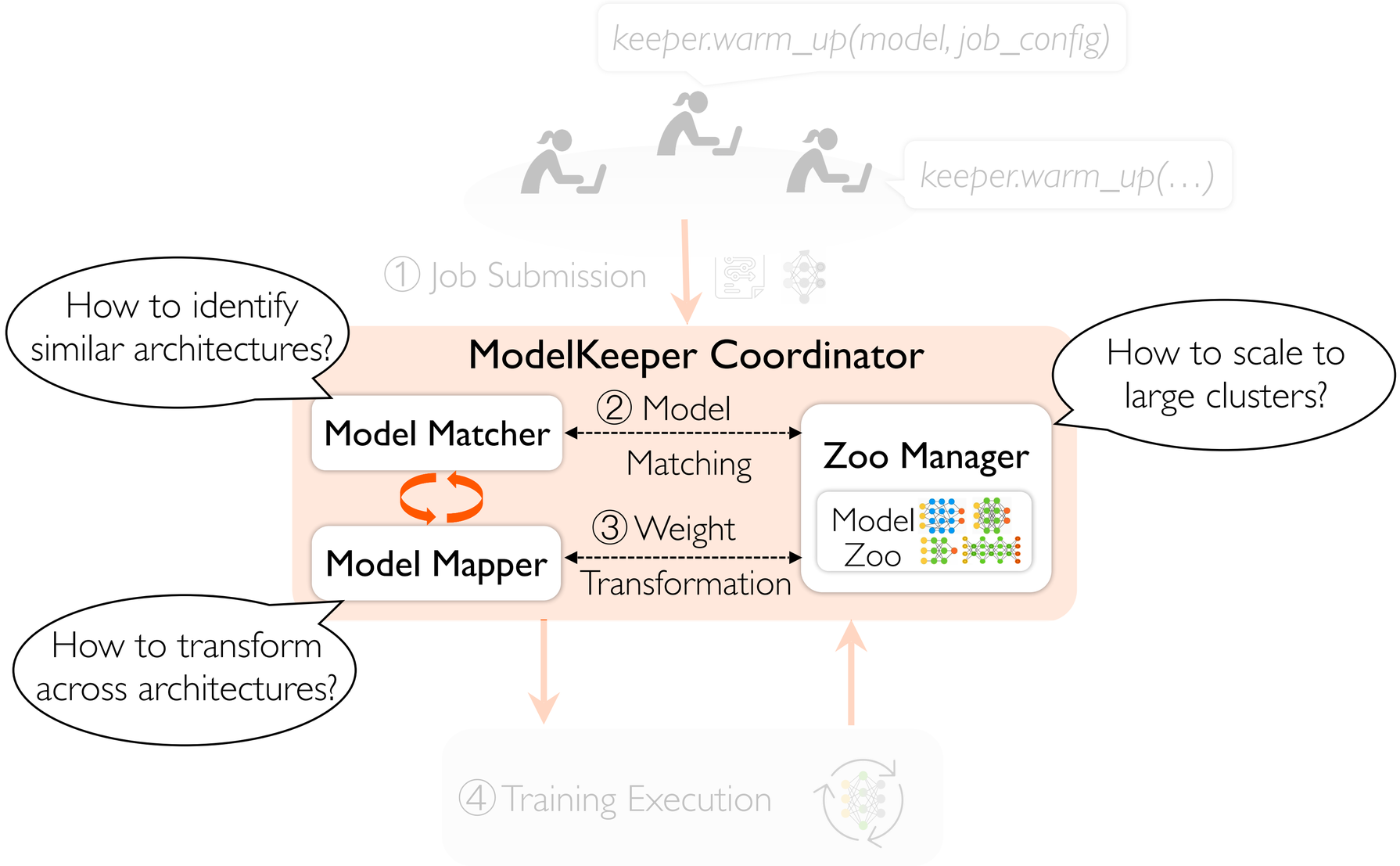
ModelKeeper

Automated Training Warmup System

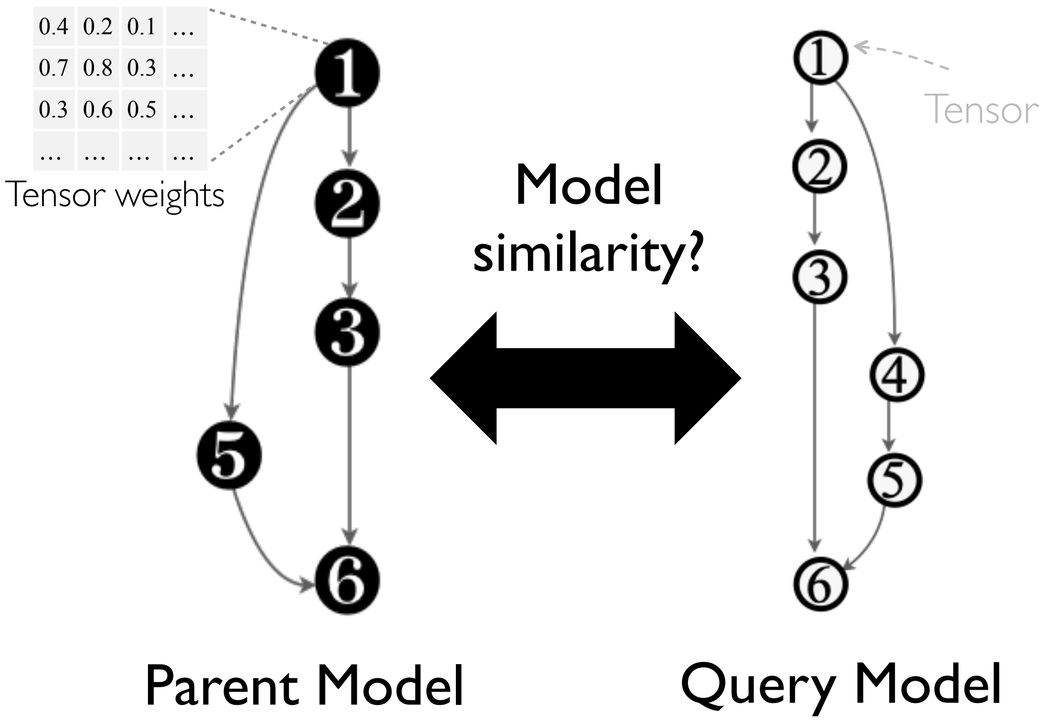


ModelKeeper

Automated Training Warmup System

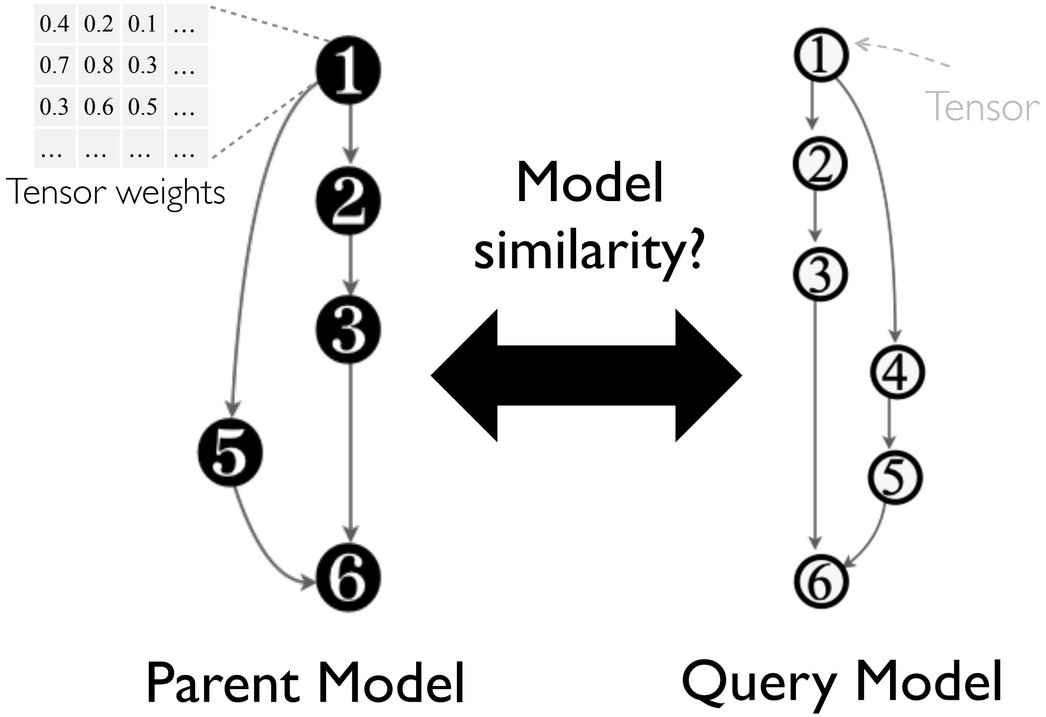


Challenge 1: Identify Architecturally Similar Models



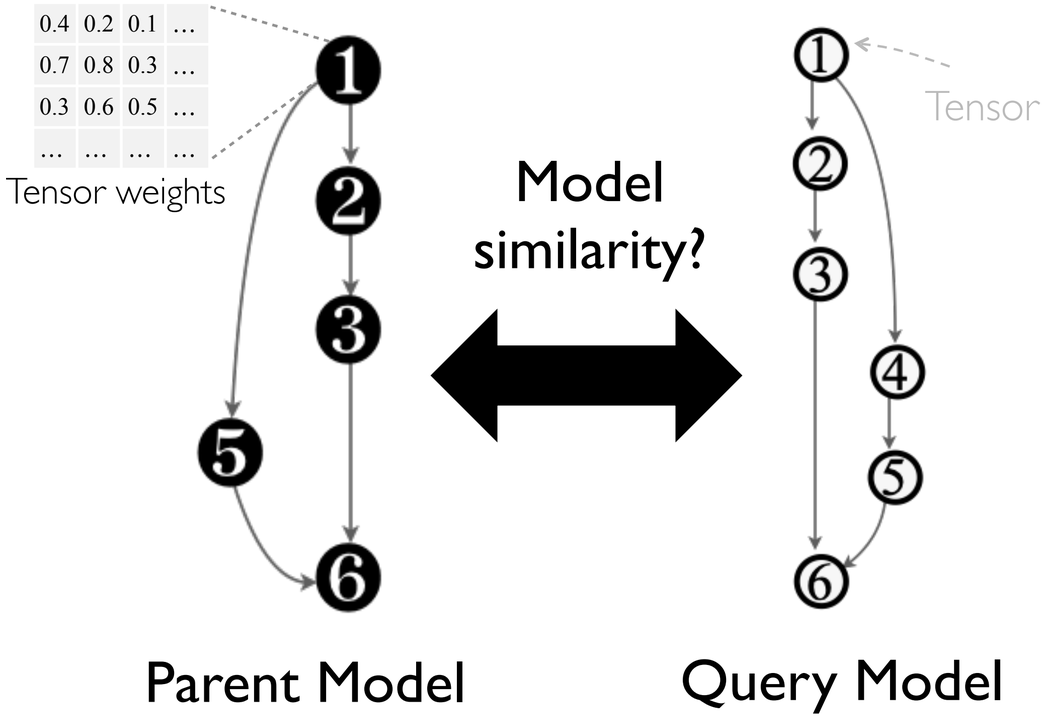
Challenge 1: Identify Architecturally Similar Models

- **How to quantify # of transformable weights between two models?**
  - Match the structure of tensors at the graph level



Challenge 1: Identify Architecturally Similar Models

- **How to quantify # of transformable weights between two models?**
  - Match the structure of tensors at the graph level



Why challenging?

- Models prefer matching prefix tensors
- Model matching can be partial
- Graph matching is NP-Hard

Challenge 1: Identify Architecturally Similar Models

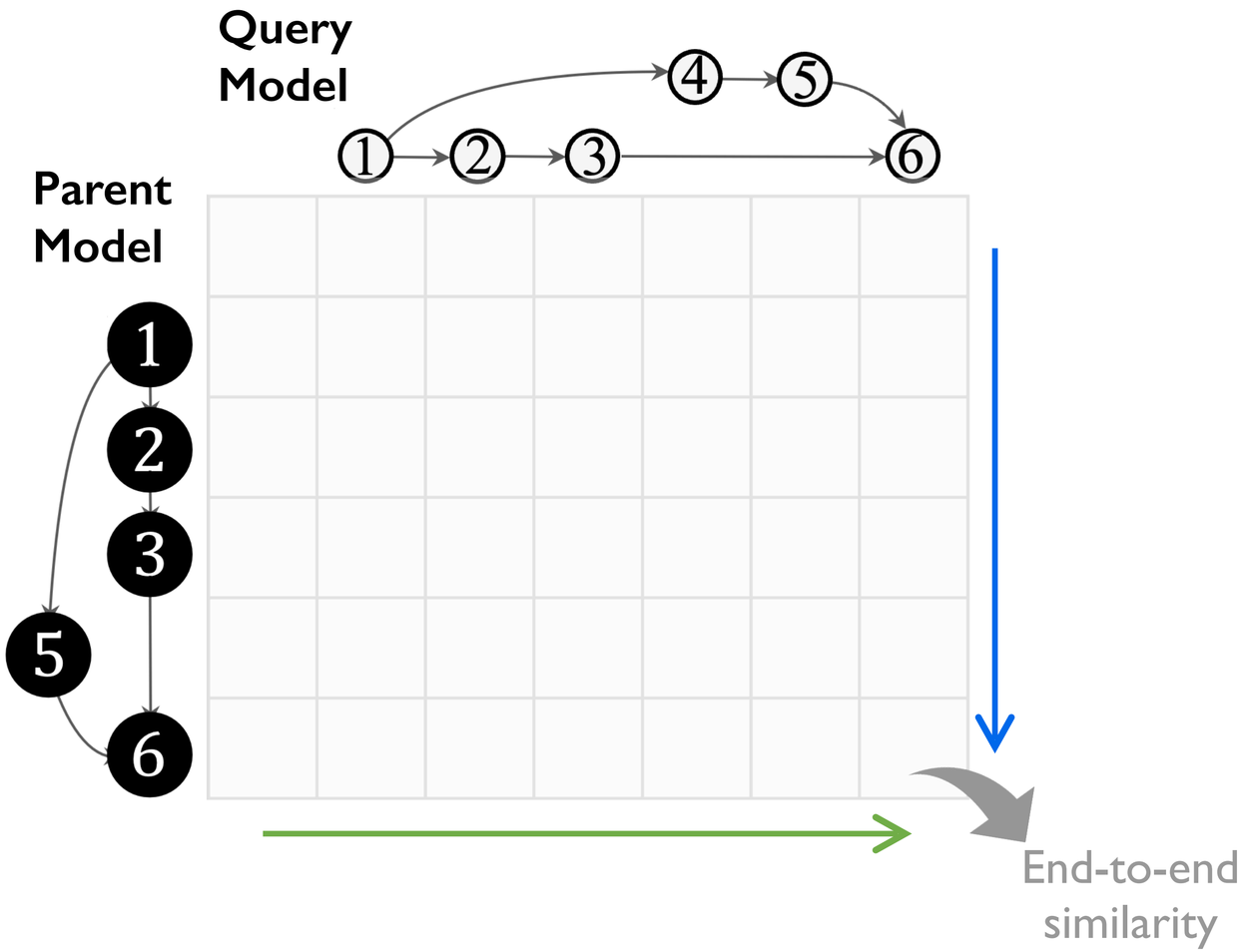
Goal

Identify the maximum number of transformable weights

Challenge 1: Identify Architecturally Similar Models

Goal

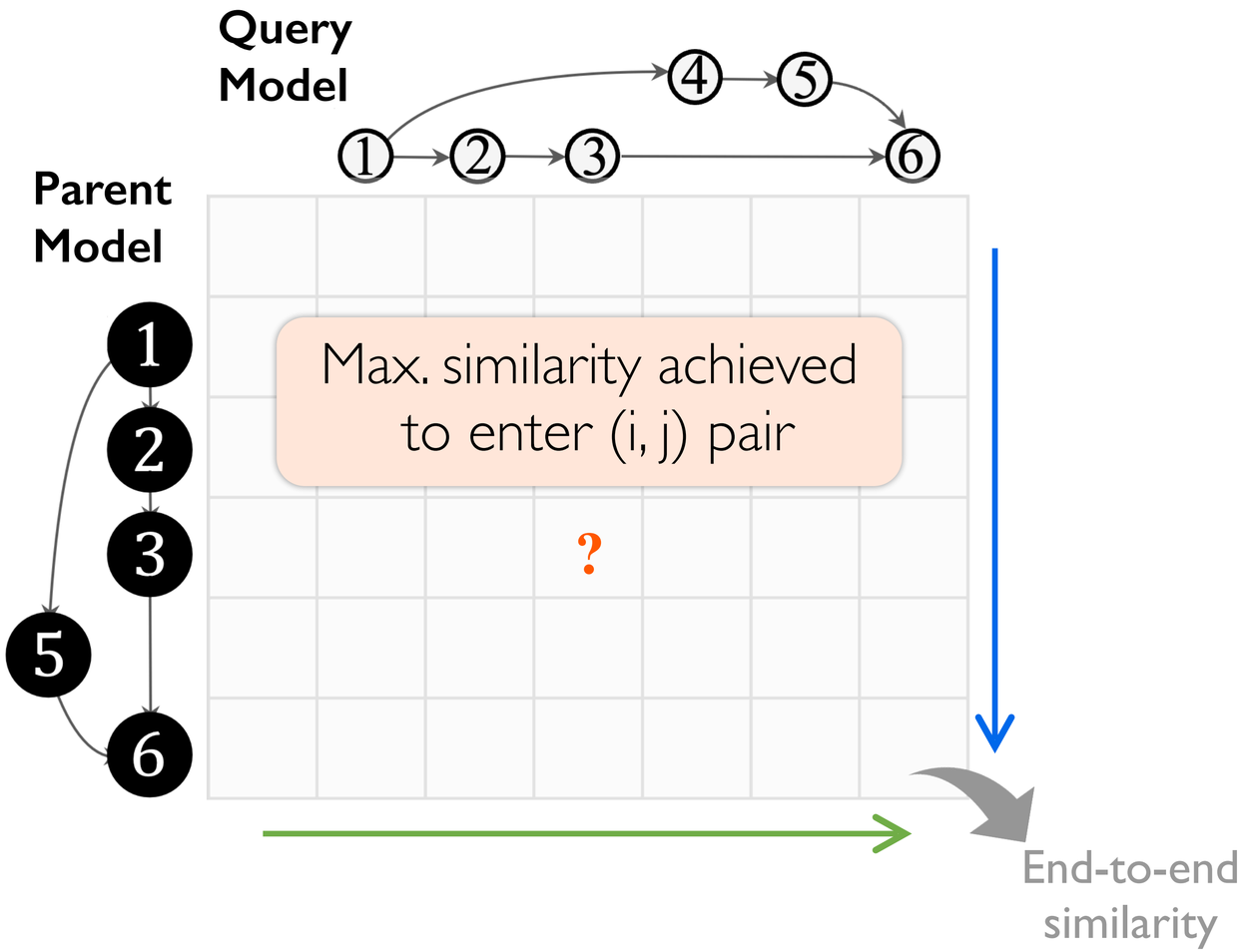
Identify the maximum number of transformable weights



Challenge 1: Identify Architecturally Similar Models

Goal

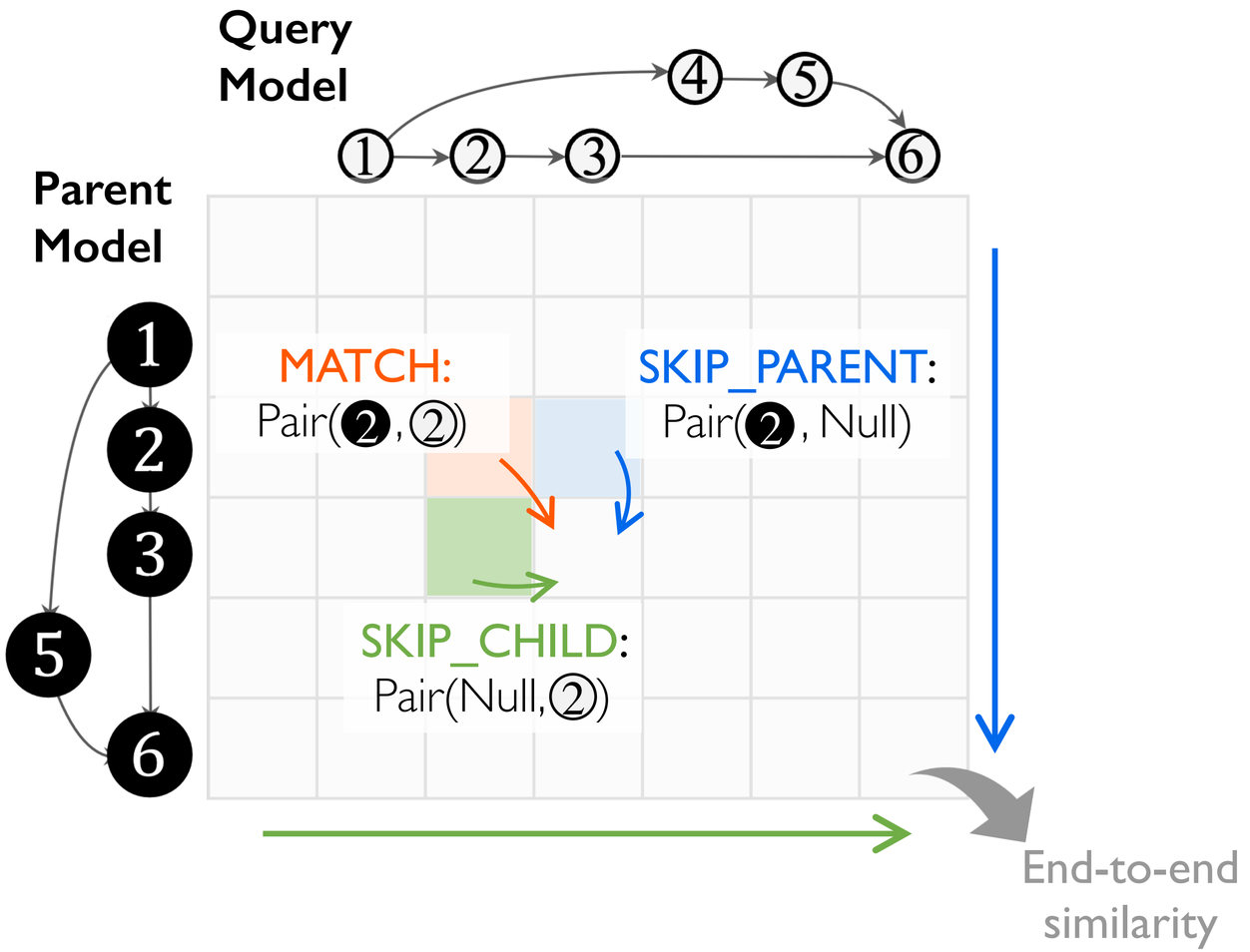
Identify the maximum number of transformable weights



Challenge 1: Identify Architecturally Similar Models

Goal

Identify the maximum number of transformable weights



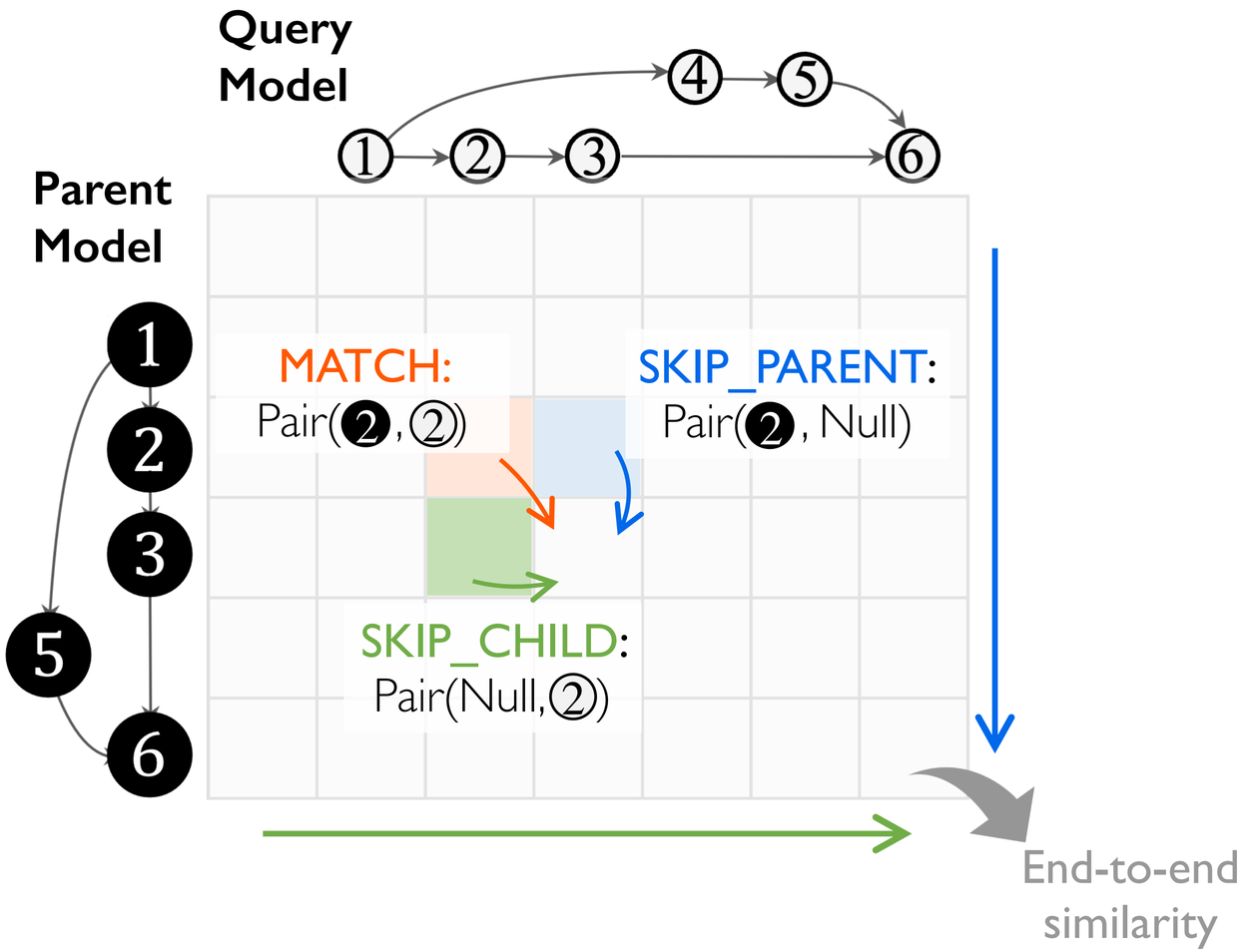
Challenge 1: Identify Architecturally Similar Models

Goal

Identify the maximum number of transformable weights

Operation score

- MATCH: % of transformable weights
- SKIP_CHILD: 0 (transfer 0% weights)



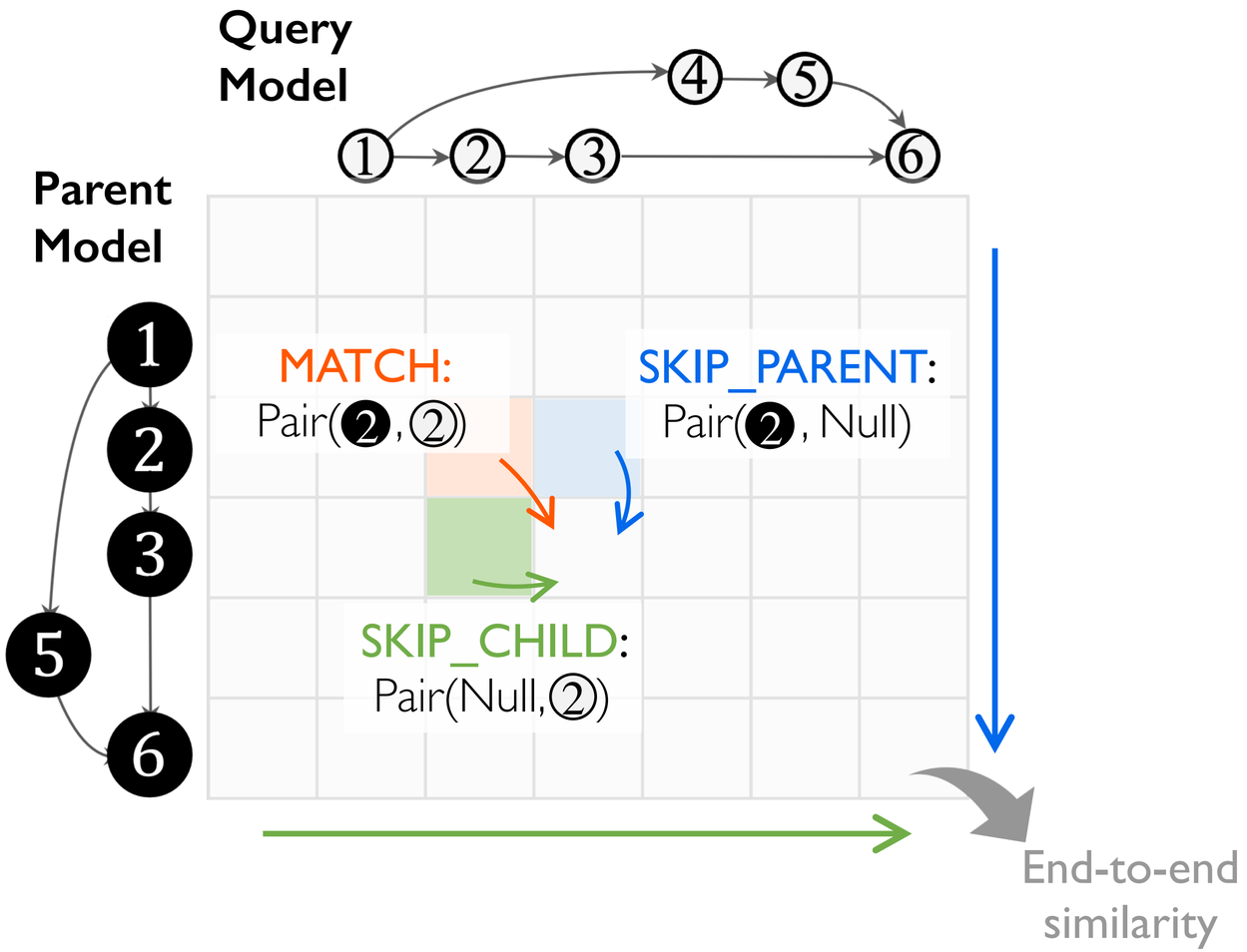
Challenge 1: Identify Architecturally Similar Models

Goal

Identify the maximum number of transformable weights

Operation score

- MATCH: % of transformable weights
- SKIP_CHILD: 0 (transfer 0% weights)
- SKIP_PARENT: -1 (lose 100% weights)



Challenge 1: Identify Architecturally Similar Models

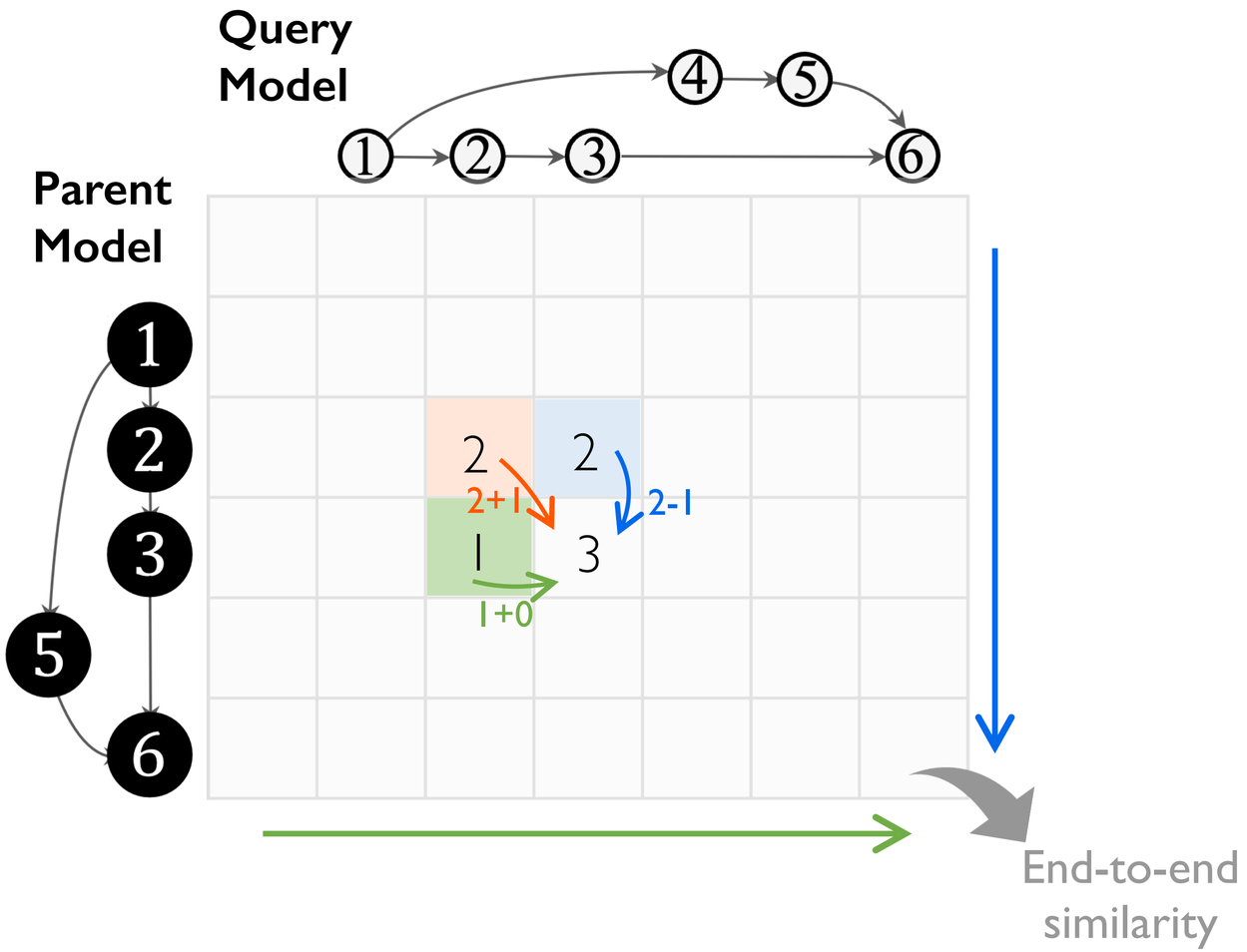
Goal

Identify the maximum number of transformable weights

Max{last_state + operation score}

Operation score

- MATCH: % of transformable weights
- SKIP_CHILD: 0 (transfer 0% weights)
- SKIP_PARENT: -1 (lose 100% weights)

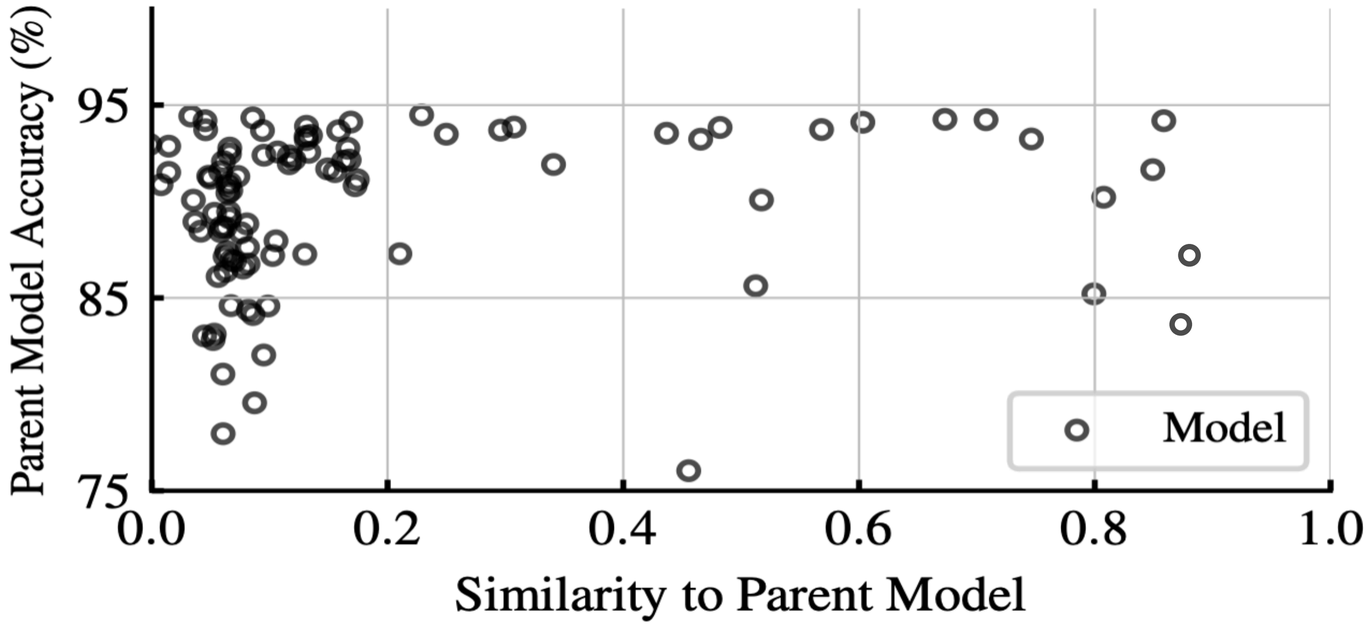


Challenge 2: Transform Max. Parent Information

- **How to achieve similarity-accuracy frontier?**
  - Bucketing by similarity, then select the high-accuracy model

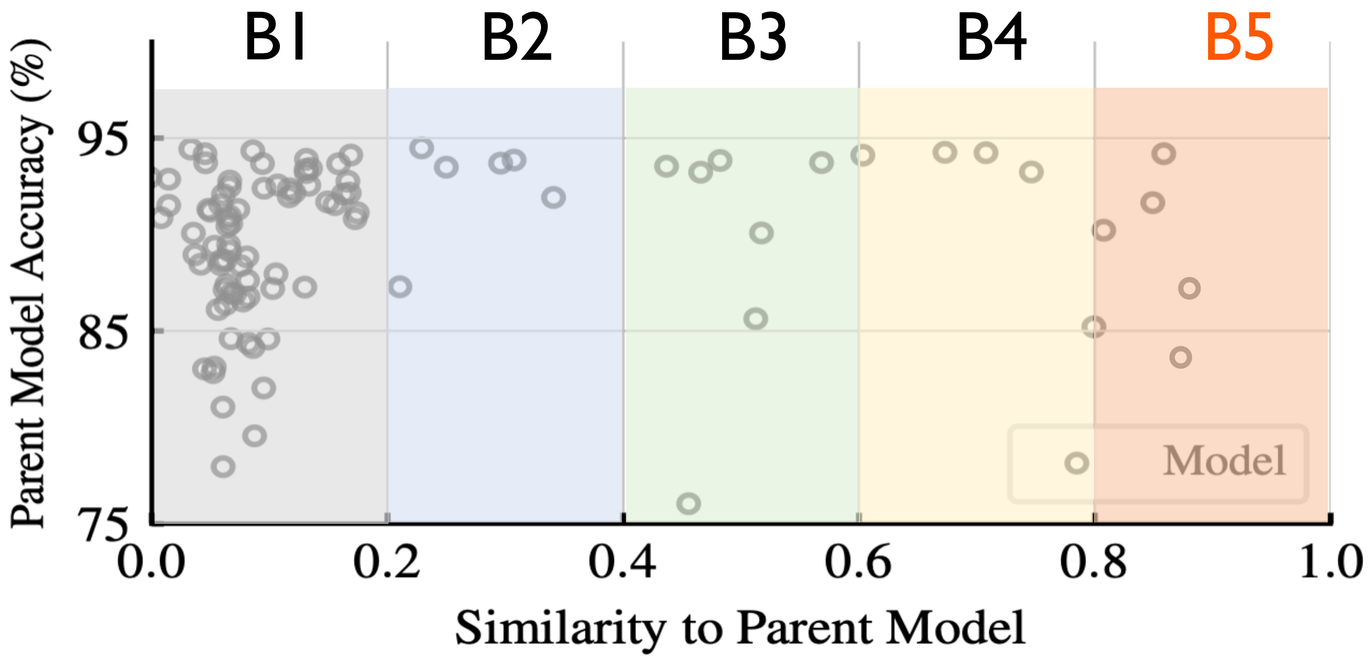
Challenge 2: Transform Max. Parent Information

- **How to achieve similarity-accuracy frontier?**
  - Bucketing by similarity, then select the high-accuracy model



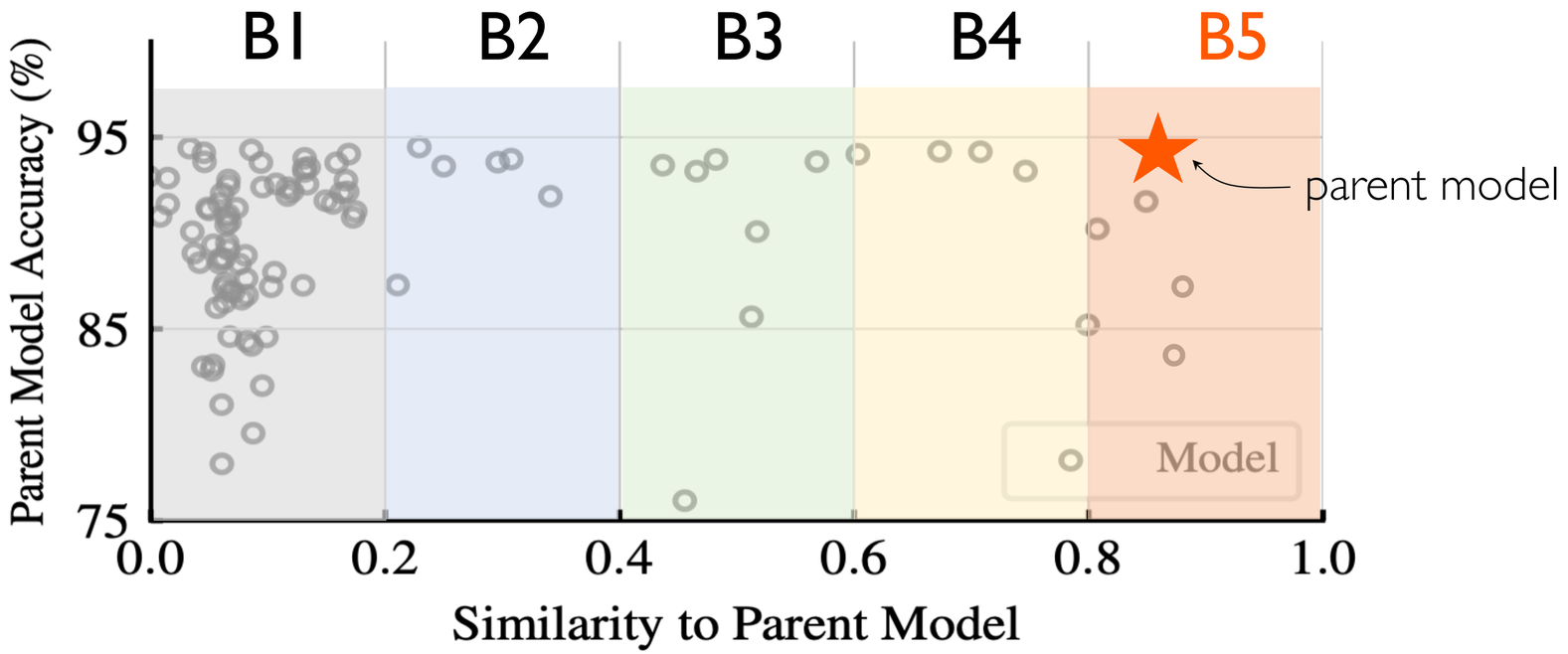
Challenge 2: Transform Max. Parent Information

- **How to achieve similarity-accuracy frontier?**
  - Bucketing by similarity, then select the high-accuracy model



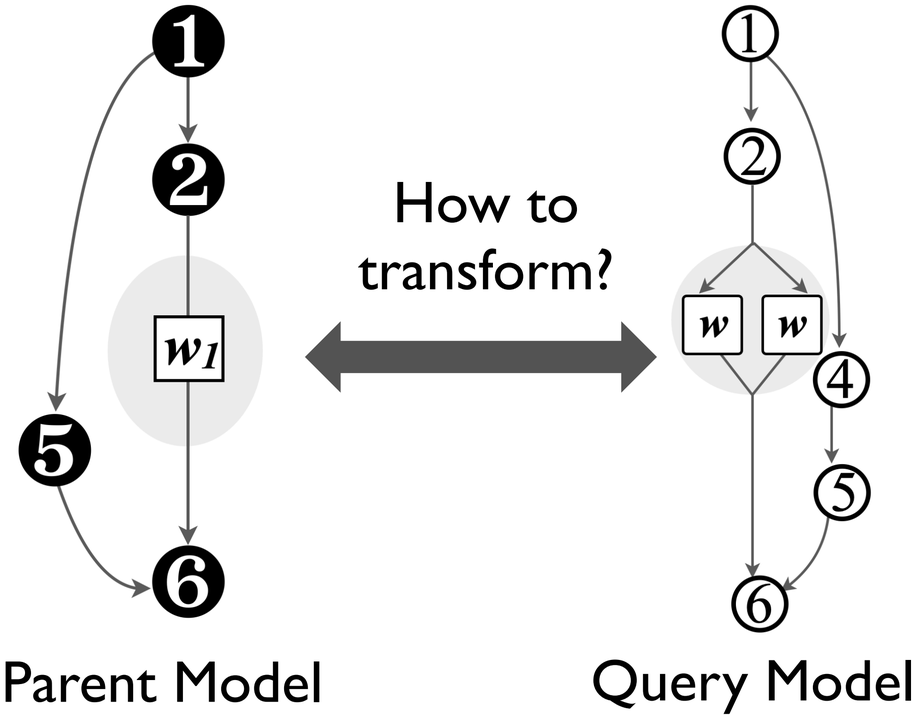
Challenge 2: Transform Max. Parent Information

- **How to achieve similarity-accuracy frontier?**
  - Bucketing by similarity, then select the high-accuracy model



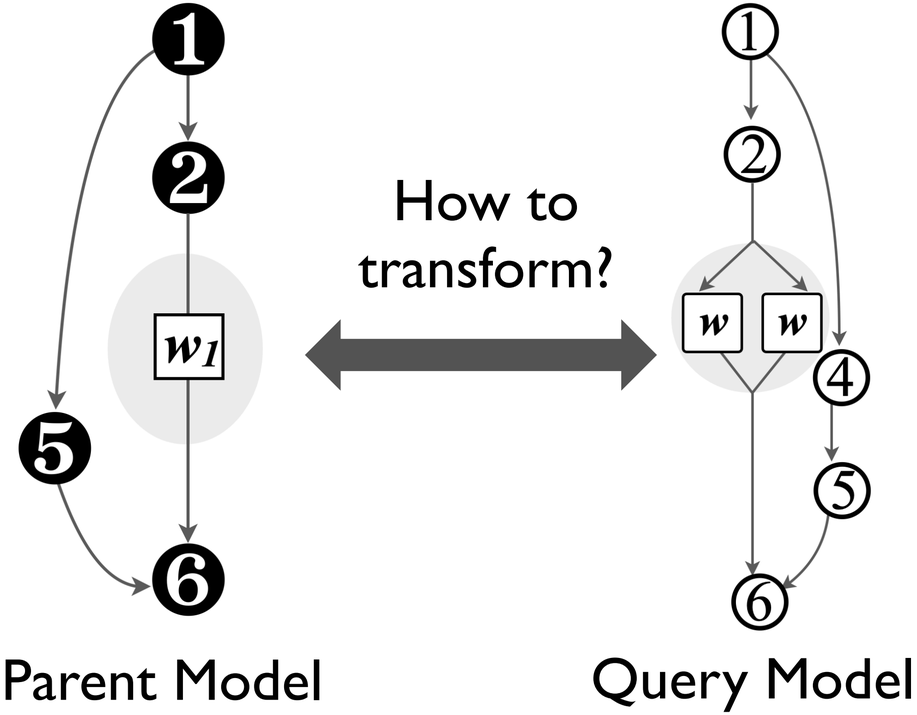
Challenge 2: Transform Max. Parent Information

- **How to transform in the presence of non-identical architectures?**



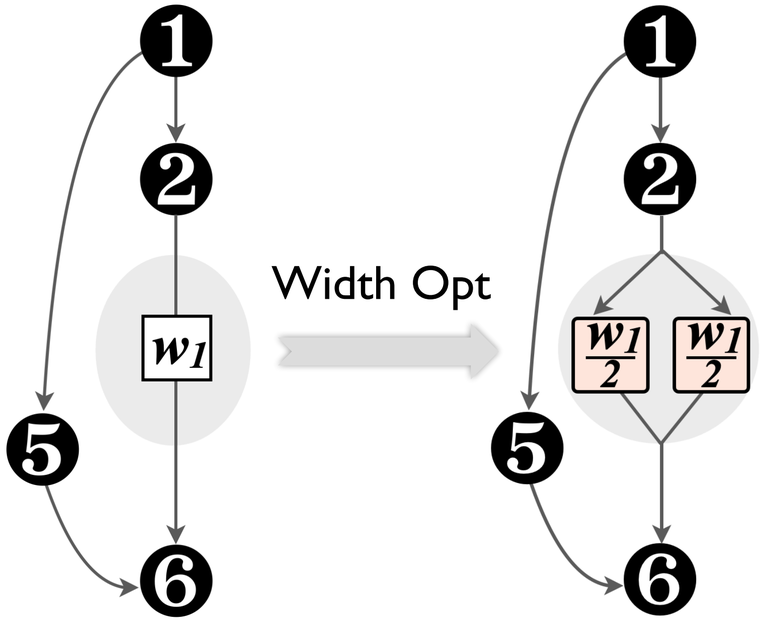
Challenge 2: Transform Max. Parent Information

- **How to transform in the presence of non-identical architectures?**
  - Function-preserving width operator, depth operator



Challenge 2: Transform Max. Parent Information

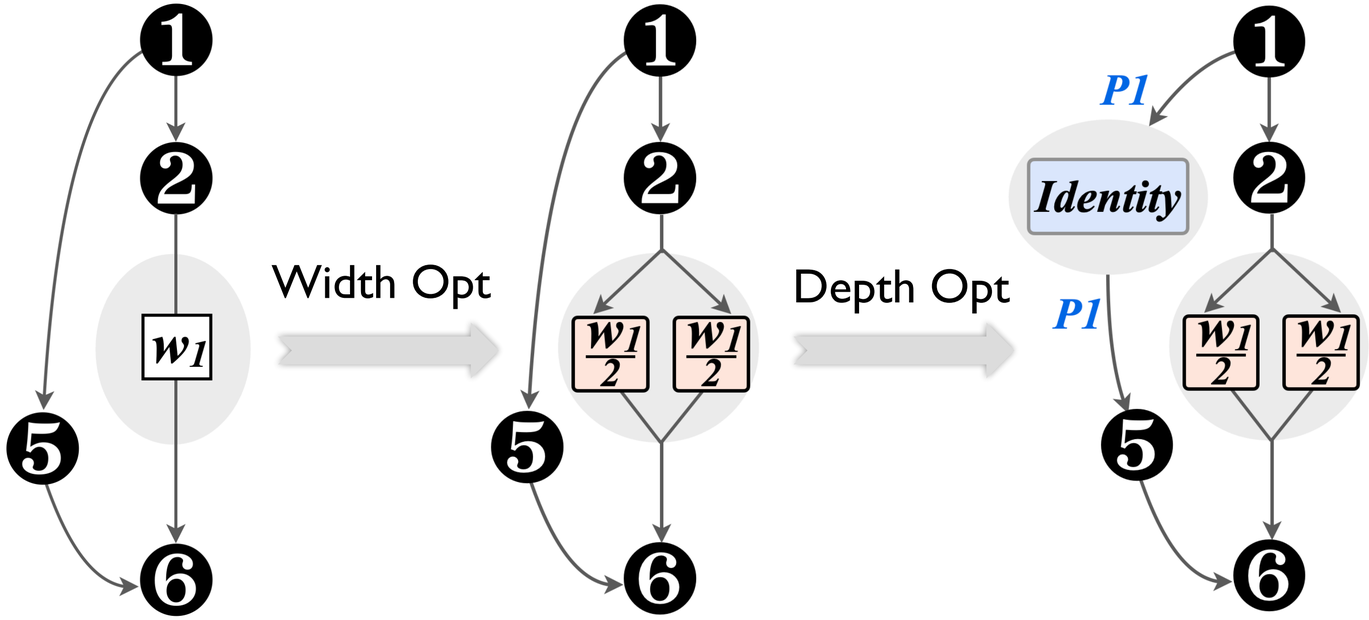
- **How to transform in the presence of non-identical architectures?**
  - Function-preserving width operator, depth operator



Parent Model

Challenge 2: Transform Max. Parent Information

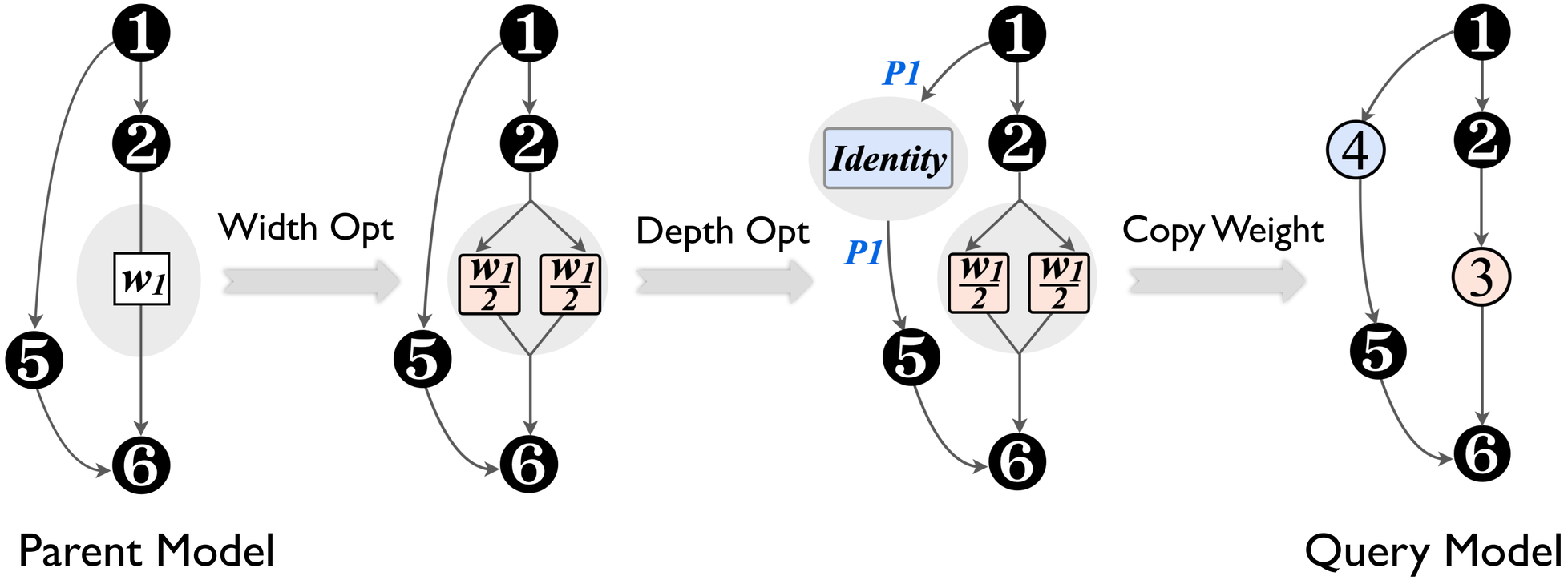
- **How to transform in the presence of non-identical architectures?**
  - Function-preserving width operator, depth operator



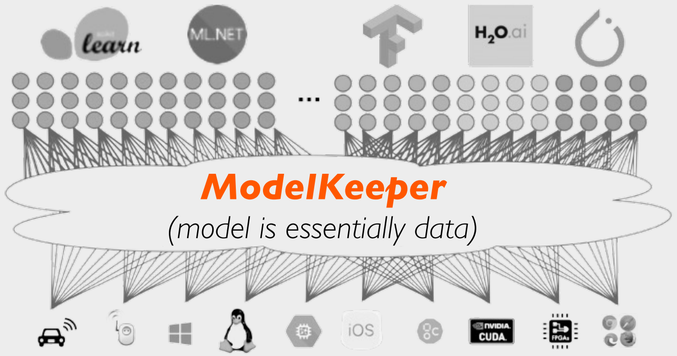
Parent Model

Challenge 2: Transform Max. Parent Information

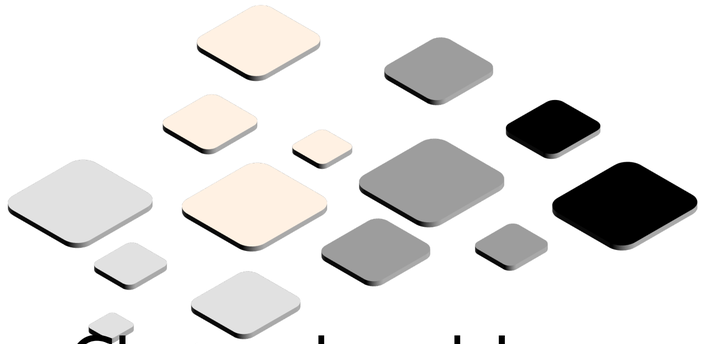
- **How to transform in the presence of non-identical architectures?**
  - Function-preserving width operator, depth operator



Transform Effectively At Scale



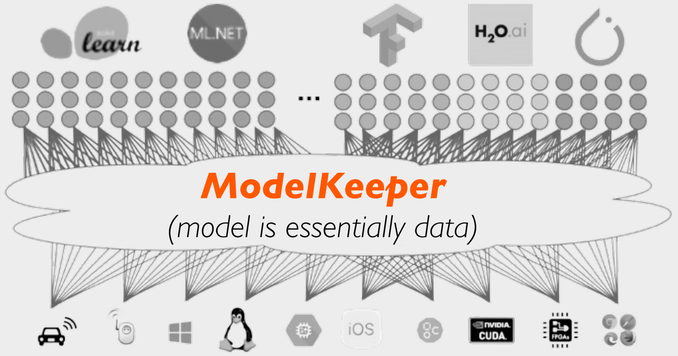
- **How to identify similar models on the cloud scale?**
  - Thousands of daily jobs



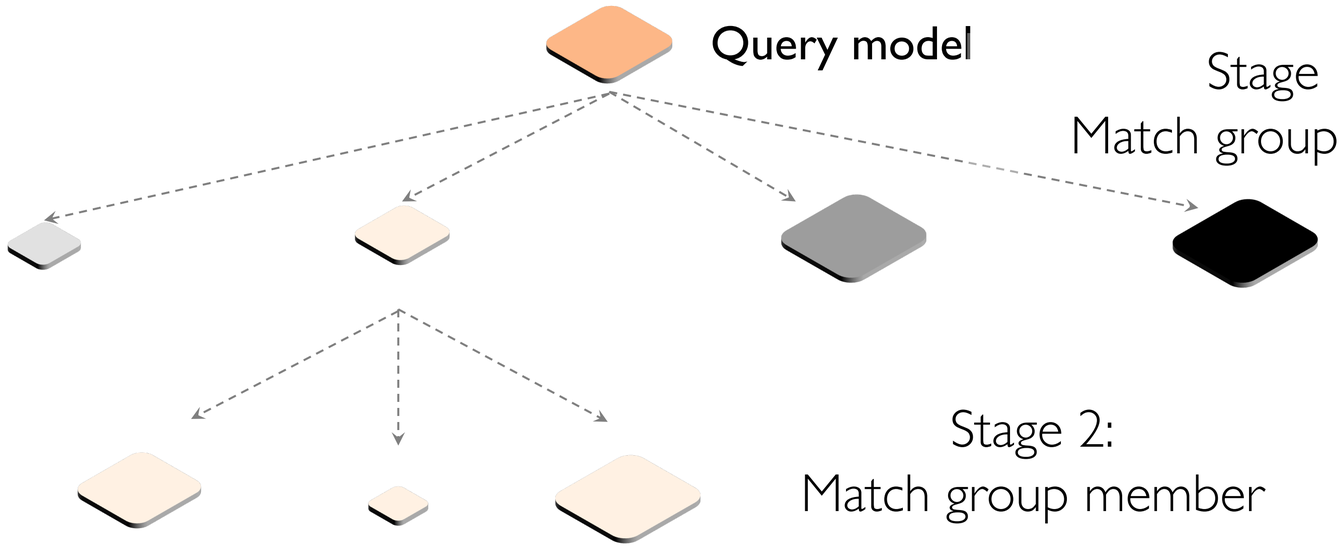
Clustered model zoo

Numerous models in the zoo

Transform Effectively At Scale

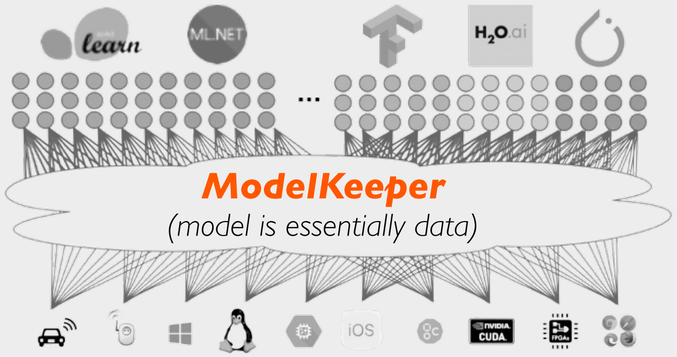


- **How to identify similar models on the cloud scale?**
  - Two-stage matching using K-medoids clustering



ModelKeeper responds to user requests in **8s** at cluster scale (**2.5k** HuggingFace zoo models).

Transform Effectively At Scale



- **How to identify similar models on the cloud scale?**
  - Two-stage matching using K-medoids clustering
- **How to manage the model zoo s.t. storage capacity?**
  - Admit and evict models on the fly
- **How to avoid low-accuracy models in the zoo?**

Please refer to our paper for details

Evaluation

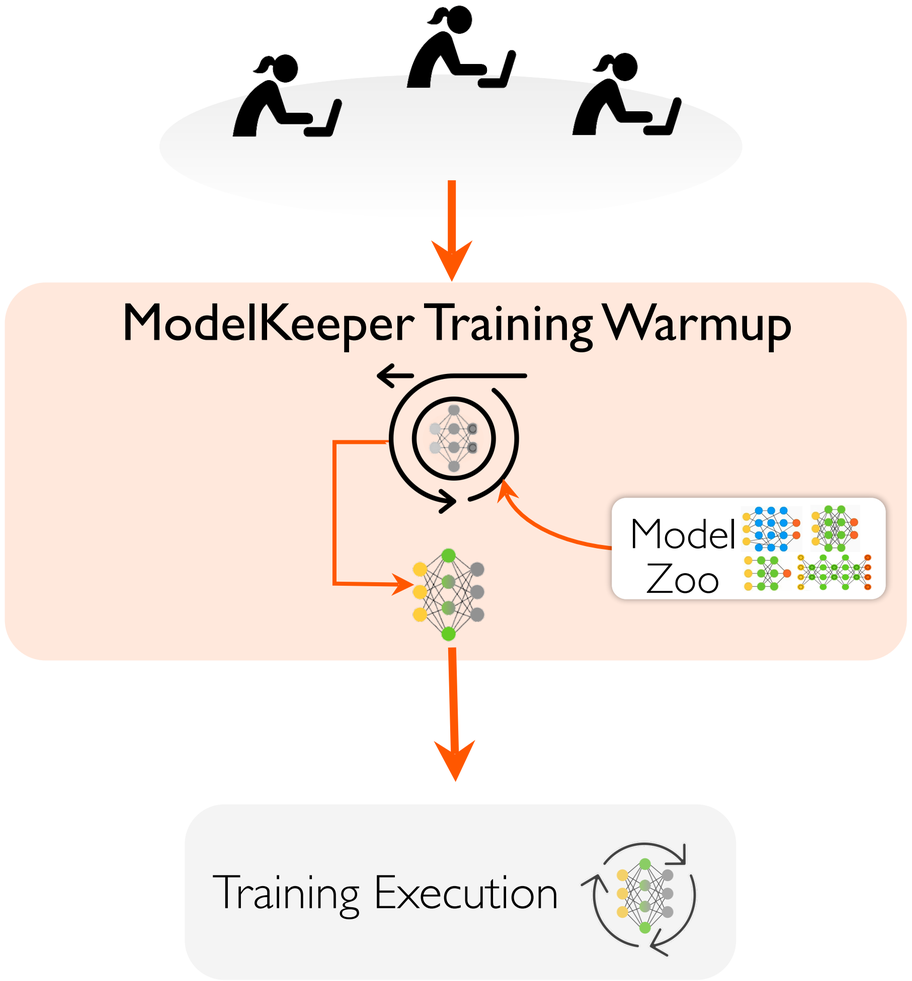
First automated training warmup system supporting

Neural Network Intelligence

mlflow

RAY

AUTO KERAS



Evaluation

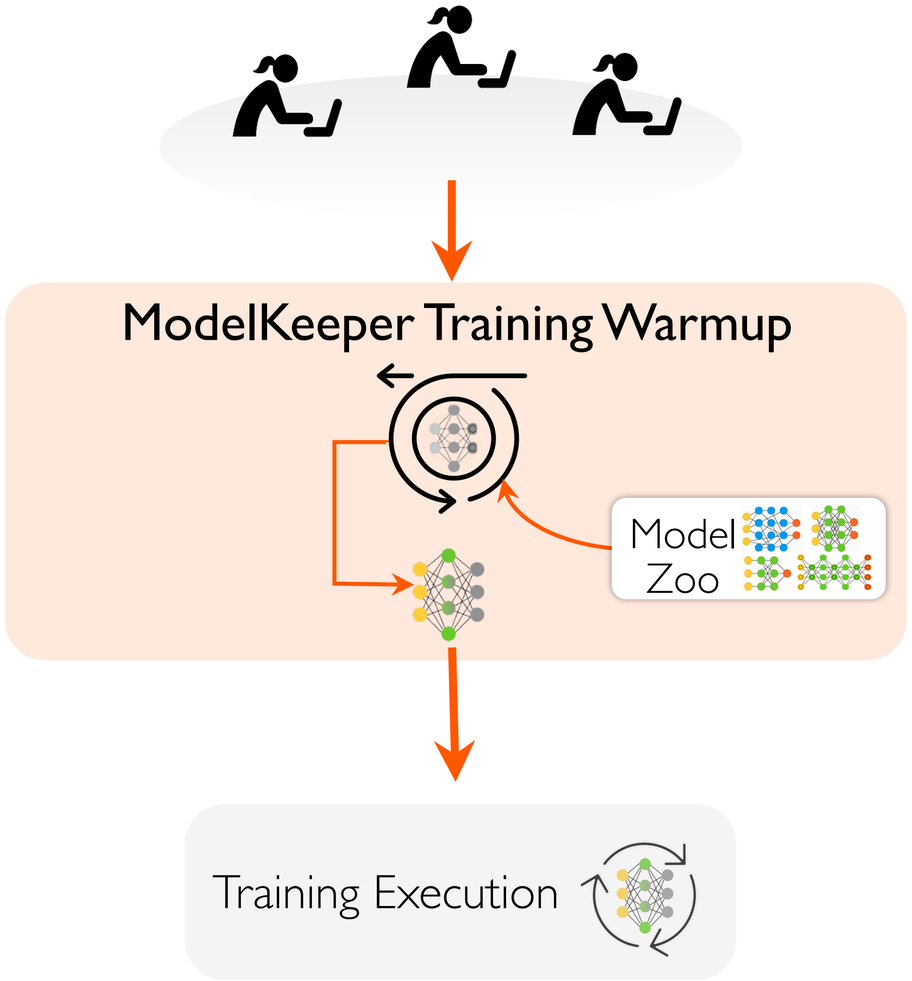
First automated training warmup system supporting

Neural Network Intelligence

mlflow

RAY

AUTO KERAS



Experiment setting:

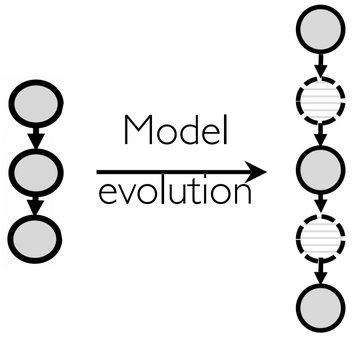
- 80-machine cluster
- 1k+ CV, NLP models
- Months of training

Large GPU Saving & Faster Training Completion (TC)

Large GPU Saving & Faster Training Completion (TC)

Task	Baseline
Neural Architecture Search	Retiarii[1]
	AutoKeras[2]

Rely on the lineage of model mutation to transfer weights

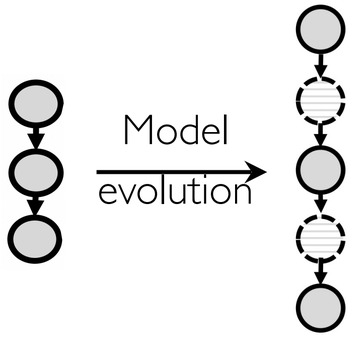


[1] Retiarii: A Deep Learning Exploratory-Training Framework, OSDI'20
[2] Auto-Keras: An Efficient Neural Architecture Search System, KDD'19

Large GPU Saving & Faster Training Completion (TC)

Task	Baseline	Workload	# of Models	GPU Saving
Neural Architecture Search	Retiarii[1]	NASBench	1,000	65.5%
	AutoKeras[2]	Bayesian-AutoKeras	500	76.7%

Rely on the lineage of model mutation to transfer weights

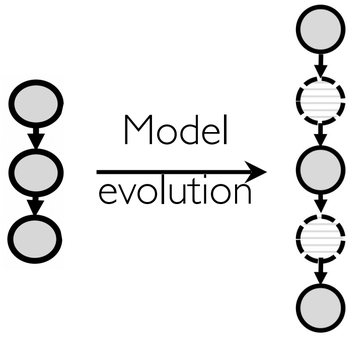


[1] Retiarii: A Deep Learning Exploratory-Training Framework, OSDI'20
[2] Auto-Keras: An Efficient Neural Architecture Search System, KDD'19

Large GPU Saving & Faster Training Completion (TC)

Task	Baseline	Workload	# of Models	GPU Saving	TC Improvement
Neural Architecture Search	Retiarii[1]	NASBench	1,000	65.5%	2.9X
	AutoKeras[2]	Bayesian-AutoKeras	500	76.7%	4.3X

Rely on the lineage of model mutation to transfer weights



[1] Retiarii: A Deep Learning Exploratory-Training Framework, OSDI'20
[2] Auto-Keras: An Efficient Neural Architecture Search System, KDD'19

Large GPU Saving & Faster Training Completion (TC)

Task	Baseline	Workload	# of Models	GPU Saving	TC Improvement
Neural Architecture Search	Retiarii[1]	NASBench	1,000	65.5%	2.9X
	AutoKeras[2]	Bayesian-AutoKeras	500	76.7%	4.3X
Ensemble Learning	MotherNet[3]	V-Ensemble	104	41.2%	1.7X
Image Classification	Ray w/o ModelKeeper	Imgclsmob	389	64.2%	2.8X
Language Modeling		HuggingFace	69	44.7%	1.8X

[1] Retiarii: A Deep Learning Exploratory-Training Framework, OSDI'20
[2] Auto-Keras: An Efficient Neural Architecture Search System, KDD'19
[3] MotherNets: Rapid Deep Ensemble Learning, MLSys'20

Large GPU Saving & Faster Training Completion (TC)

Task	Baseline	Workload	# of Models	GPU Saving	TC Improvement
Neural Architecture Search	Retiarii[1]	NASBench	1,000	65.5%	2.9X
	AutoKeras[2]	Bayesian-AutoKeras	500	76.7%	4.3X
Ensemble Learning	MotherNet[3]	V-Ensemble	104	41.2%	1.7X
Image Classification	Ray w/o ModelKeeper	Imgclsmob	389	64.2%	2.8X
Language Modeling		HuggingFace	69	44.7%	1.8X

[1] Retiarii: A Deep Learning Exploratory-Training Framework, OSDI'20
[2] Auto-Keras: An Efficient Neural Architecture Search System, KDD'19
[3] MotherNets: Rapid Deep Ensemble Learning, MLSys'20

ModelKeeper

Reduce training execution via automated training warmup

https://github.com/SymbioticLab/ModelKeeper

Save training execution {
- *by identifying a trained model with similar architectures*
- *by transforming model weights across architectures*

Thank you!

UNIVERSITY OF MICHIGAN