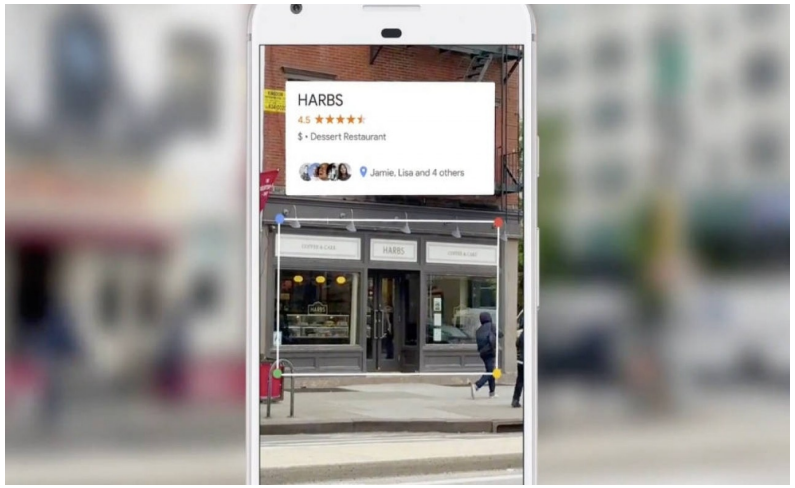
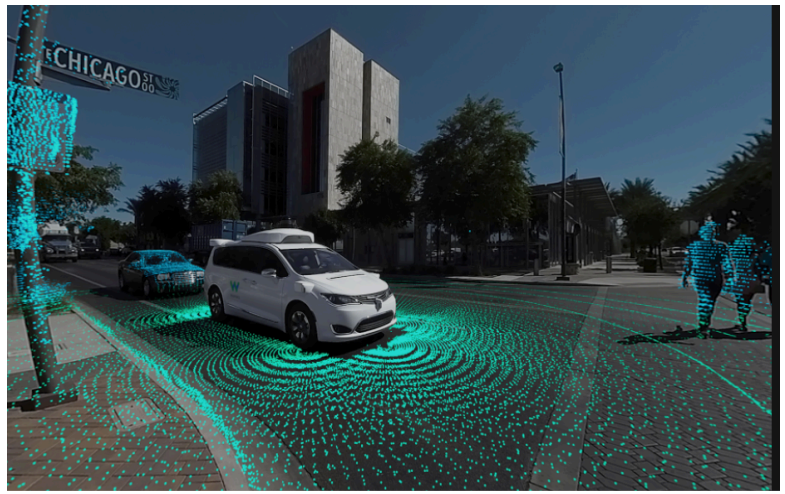
Mistify: Automating DNN Model Porting for On-Device Inference at the Edge

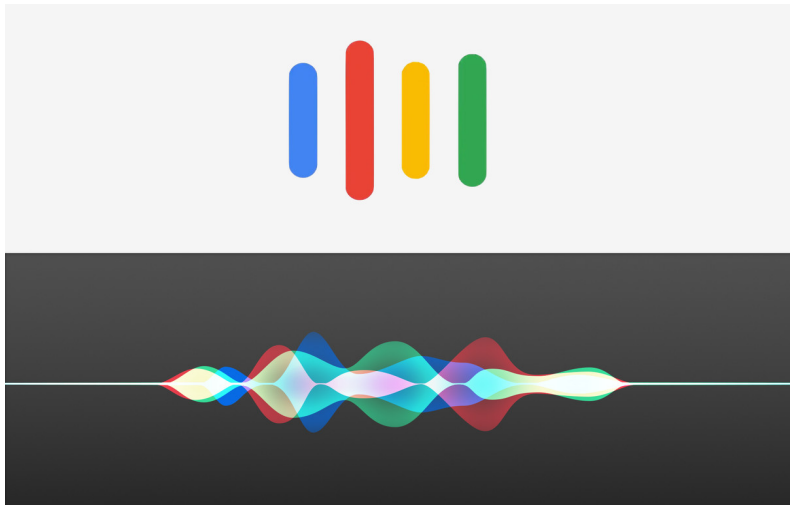
Peizhen Guo, Bo Hu, Wenjun Hu

Yale University

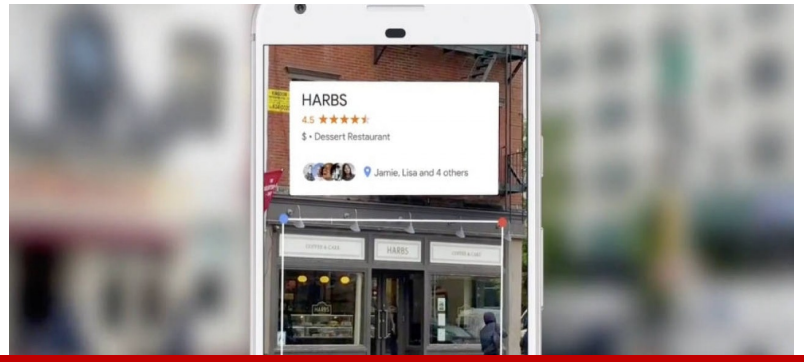
On-device deep learning inference

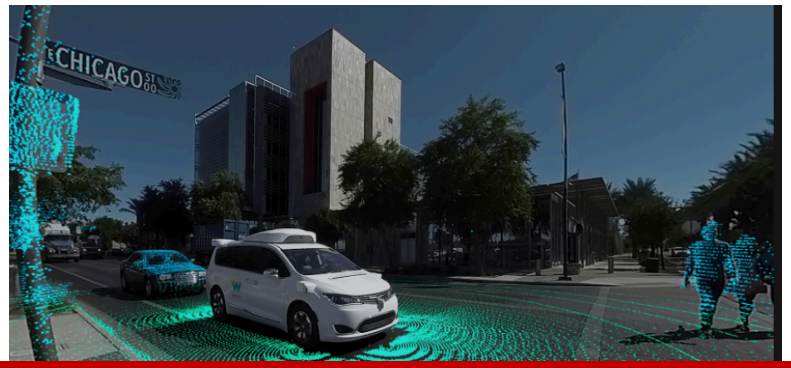






On-device deep learning inference



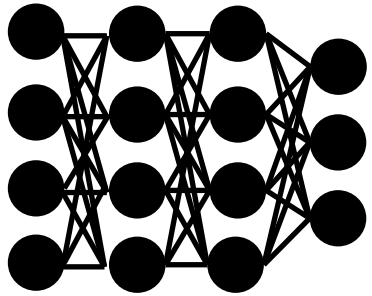


Need compact and accurate DNN models

Where do the models come from?

Where do the models come from?

Pre-trained model



TensorFlow Hub

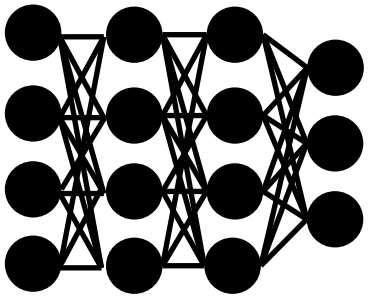
PYTORCH HUB

GLUON

Keras

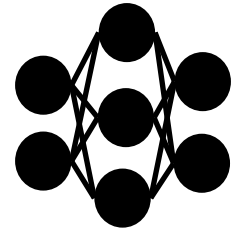
Where do the models come from?

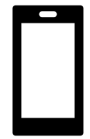
Pre-trained model

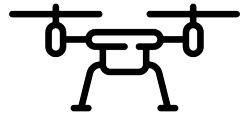


Tailor to the deployment setting

Deployed model







TensorFlow Hub

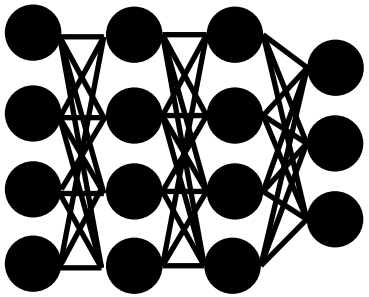
PYTORCH HUB

GLUON

Keras

Tons of DNN tailoring algorithms

Pre-trained model



MorphNet: Fast & Simple Resource-Constrained Structure Learning of Deep Networks

PROXYLESSNAS: DIRECT NEURAL ARCHITECTURE SEARCH ON TARGET TASK AND HARDWARE

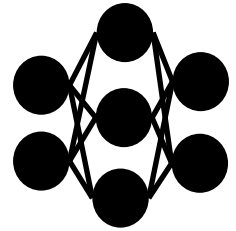
AdaNet: Adaptive Structural Learning of Artificial Neural Networks

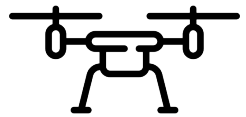
ONCE-FOR-ALL: TRAIN ONE NETWORK AND SPECIALIZE IT FOR EFFICIENT DEPLOYMENT

ChamNet: Towards Efficient Network Design through Platform-Aware Model Adaptation

SLIMMABLE NEURAL NETWORKS

Deployed model





TensorFlow Hub

PYTORCH HUB

GLUON

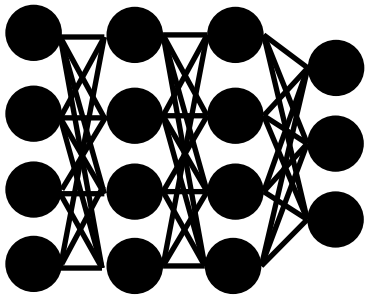
Keras

Many others……

However, tailoring a DNN is still not trivial!

However, tailoring a DNN is still not trivial!

Original model



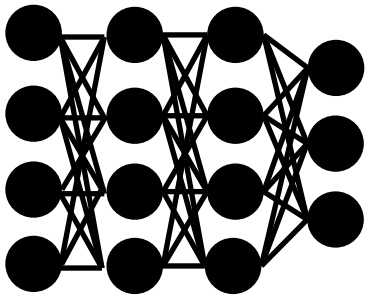
Edge deployment goal:
- 10 GFLOPs
- 300 MB size

Implement the tailoring algorithm

Laborious manual efforts
- Annotate the DNNs
- Configure parameters

However, tailoring a DNN is still not trivial!

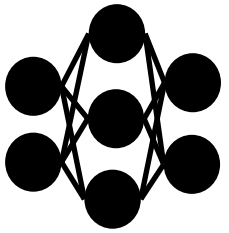
Original model



Edge deployment goal:
- 10 GFLOPs
- 300 MB size

Implement the tailoring algorithm

Execute the DNN adaptation

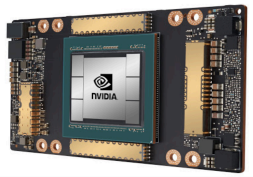


Tailored model

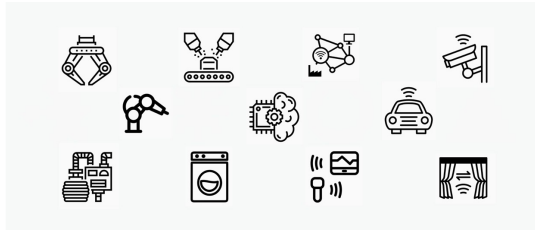
Expensive search process
- Hundreds of GPU hours
- 10x Gig memory usage

Laborious manual efforts
- Annotate the DNNs
- Configure parameters

Even worse in practice - Heterogeneous hardware targets





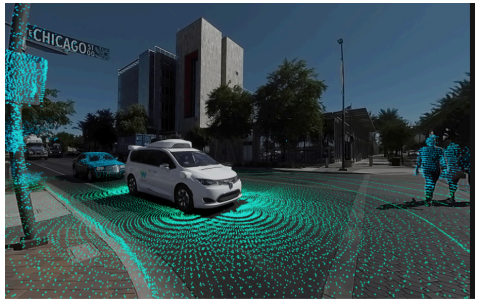


Hardware

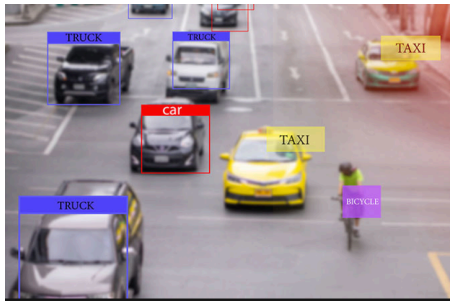
TFLOP/s GFLOP/s <MFLOP/s

Even worse in practice - Heterogeneous performance requirements

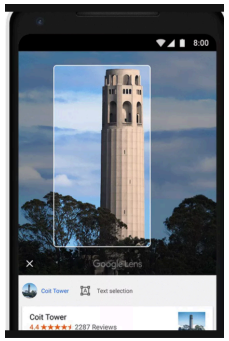
Autonomous driving



Traffic monitoring



Google Lens



Performance requirements

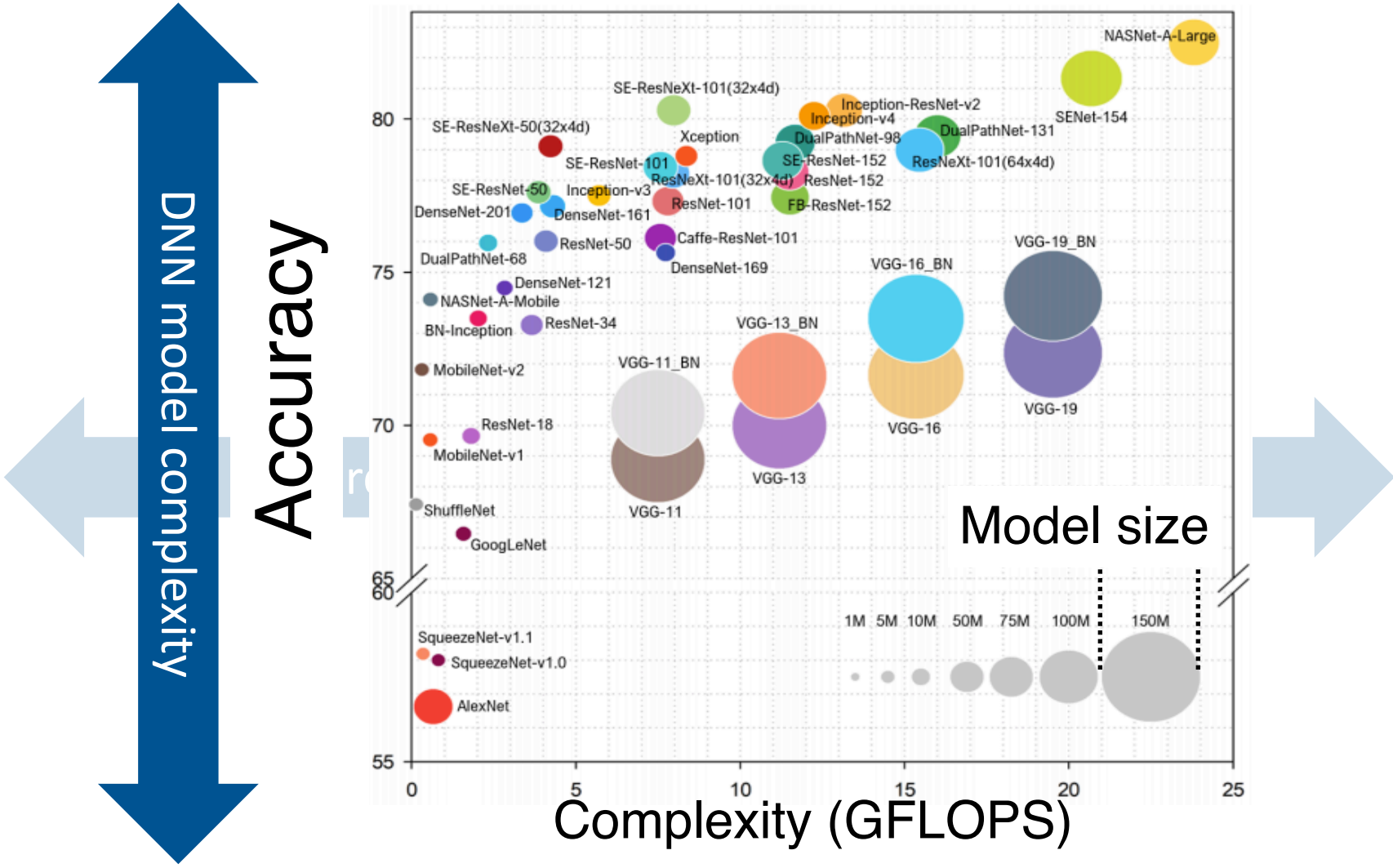
~1ms

~30ms

~1s

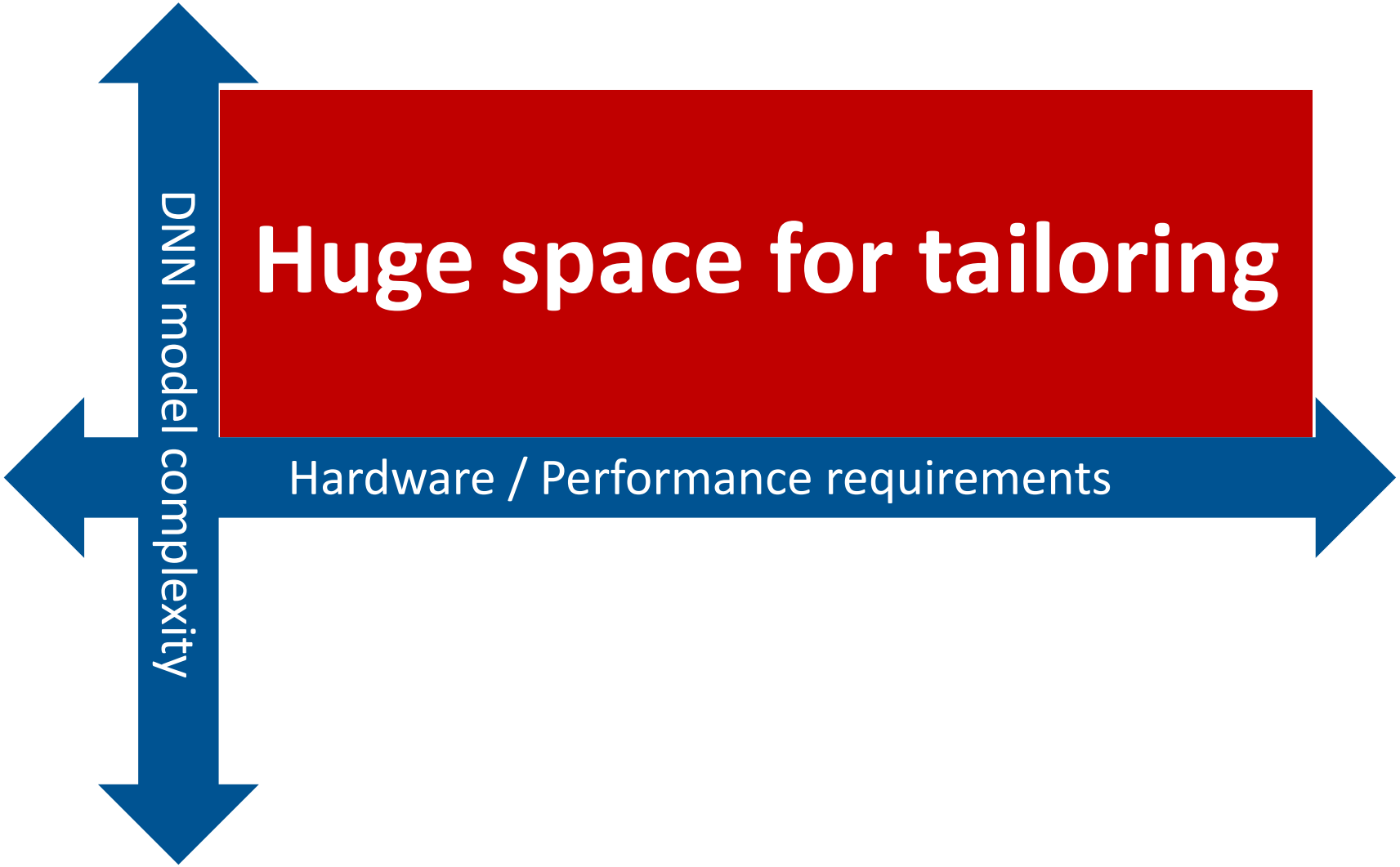
Even worse in practice - Model Diversity

Even worse in practice - Model Diversity

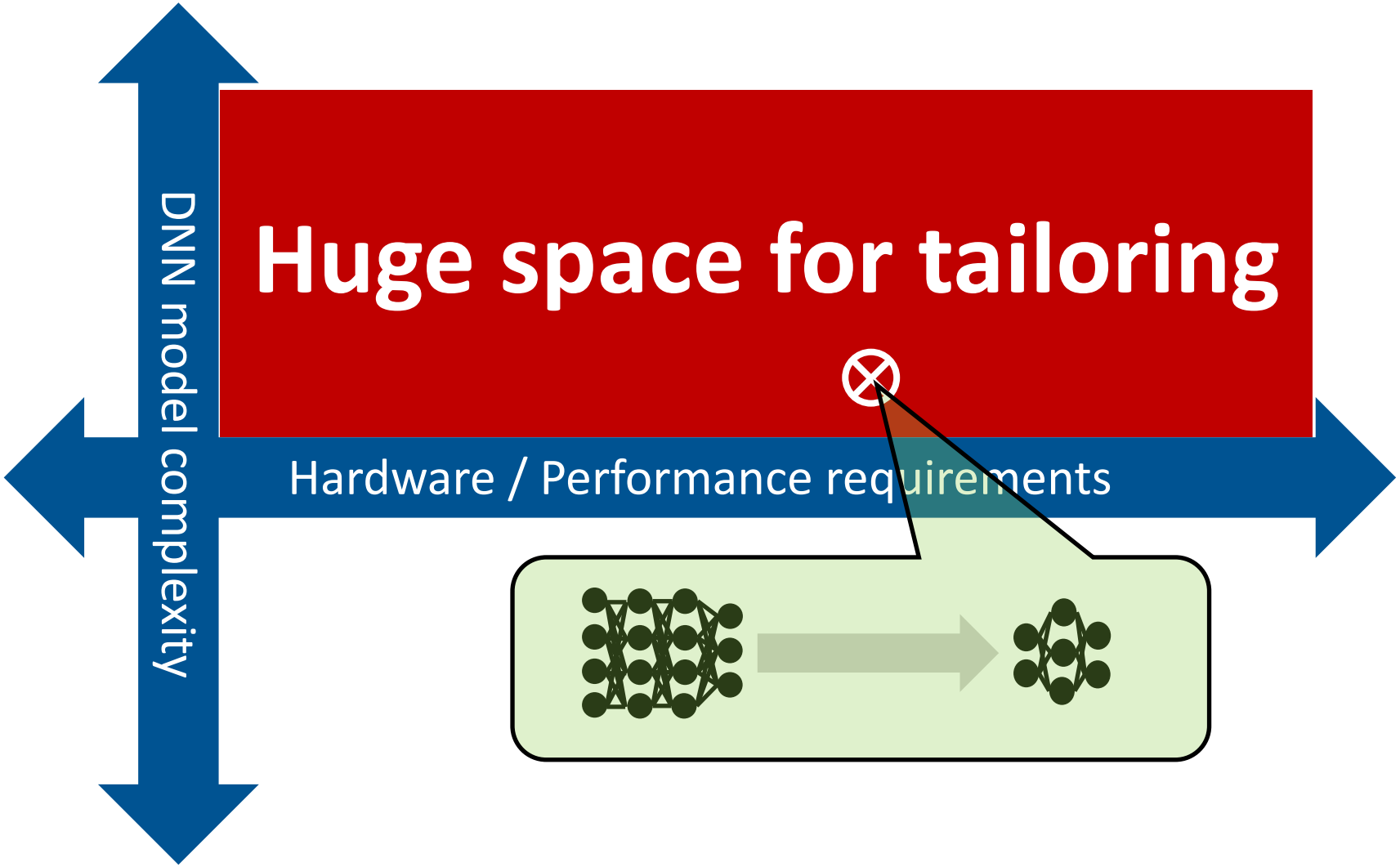


BianCo, Simone, et al. "Benchmark Analysis of Representative Deep Neural Network Architectures" *arXiv:1810.00763* (2018).

Even worse in practice -
Huge tailoring space



Even worse in practice -
Huge tailoring space



Even worse in practice - Runtime dynamics

App requirement dynamics
- Accuracy (critical vs. idle)
- Latency (day vs. night)
- Power (battery vs. charged)
- …

Device resource dynamics
- Memory space
- CPU quota
- Accelerator availability
- Queuing time

Summary: practical challenges

Unscalable DNN tailoring needs

Runtime dynamics

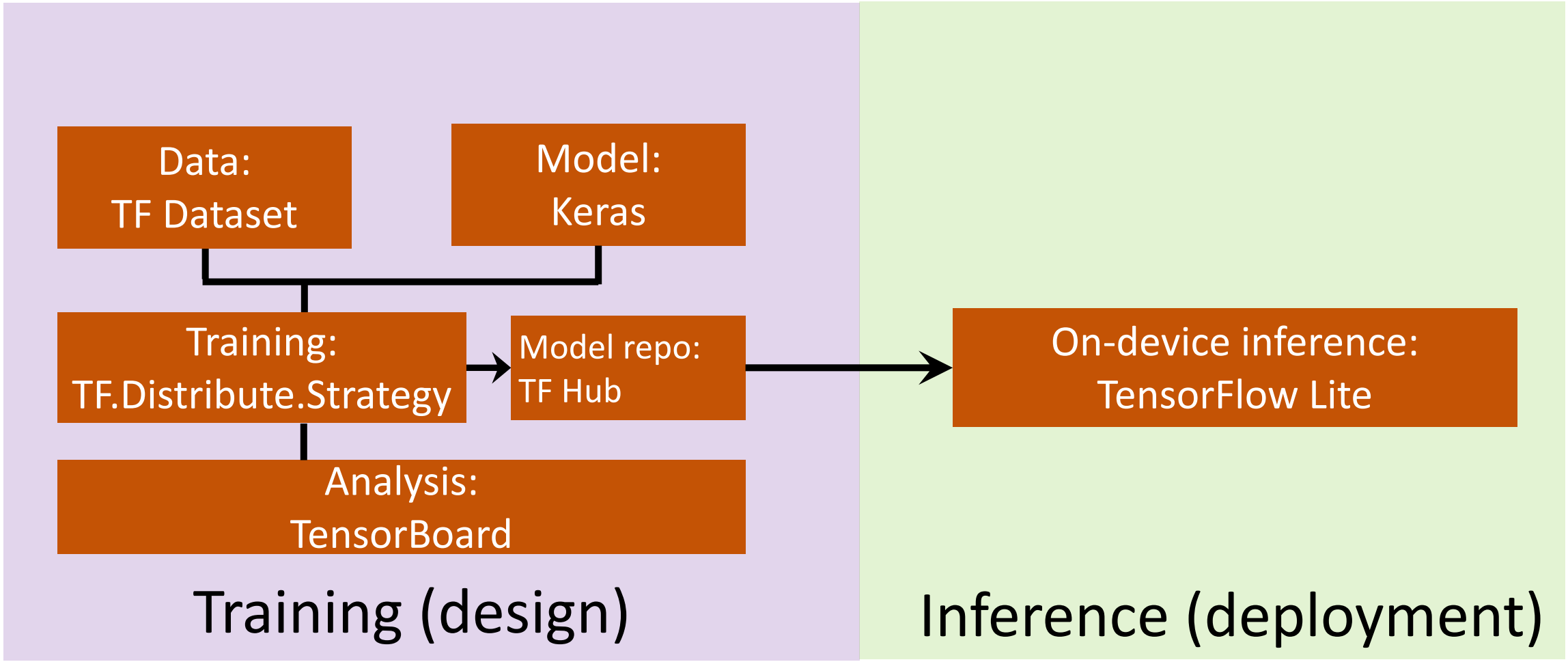
Summary: practical challenges

Unscalable DNN tailoring needs

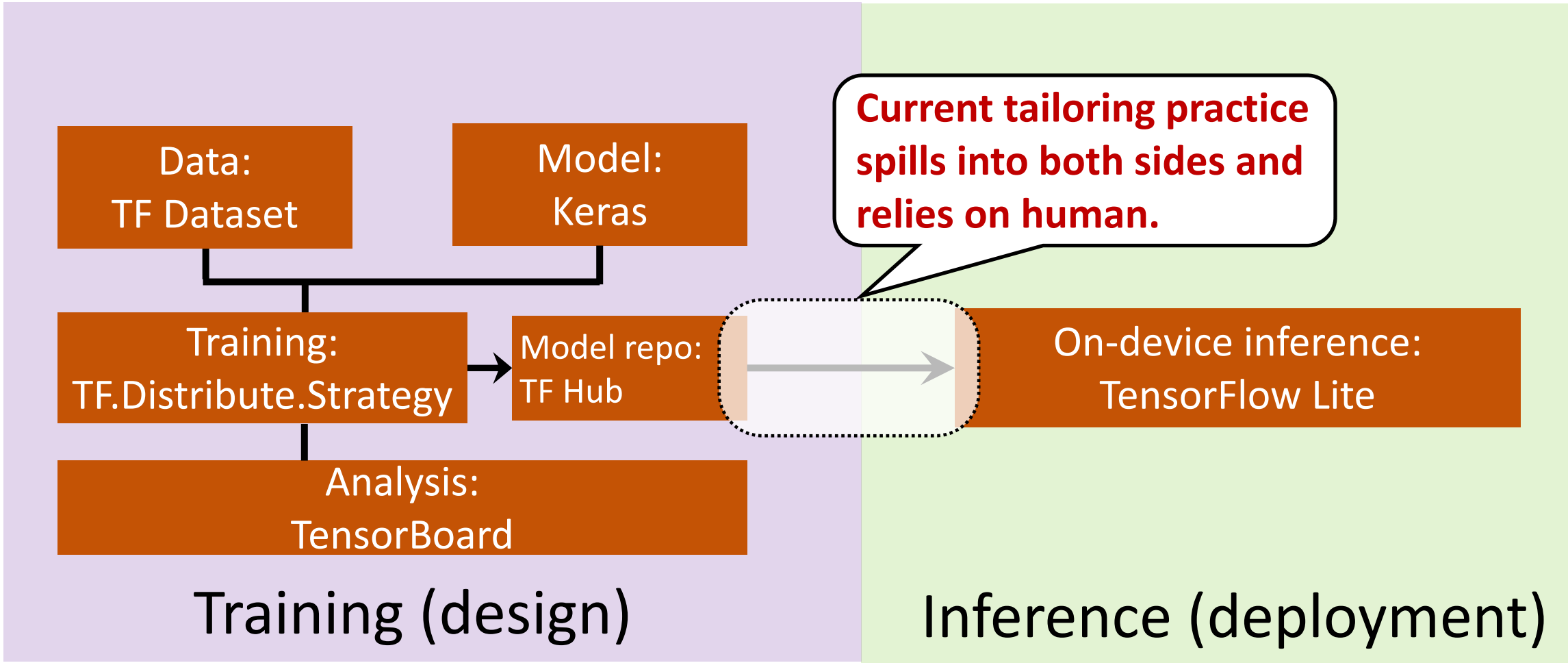
Runtime dynamics

Need *system support*

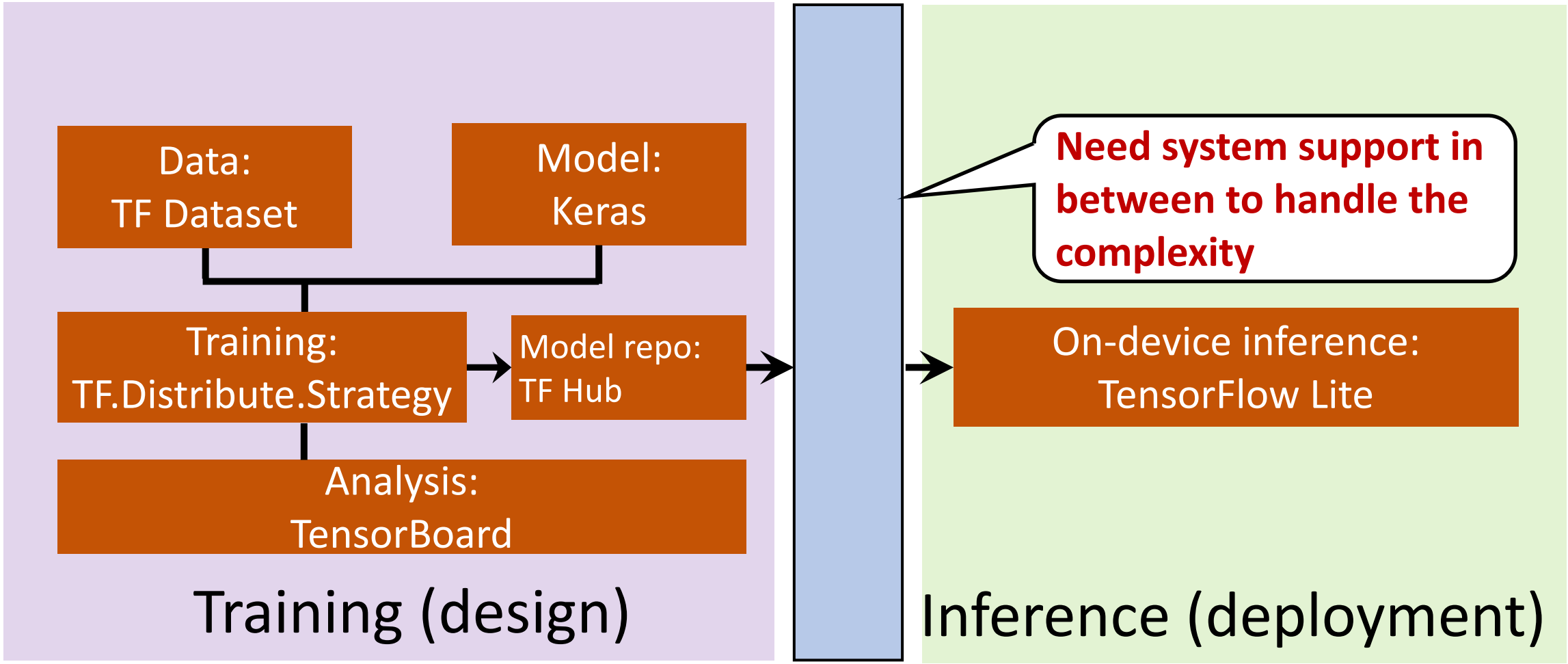
Existing DL ecosystems



Current DNN tailoring practice



Ideal DNN tailoring practice



Our solution - *Mistify*

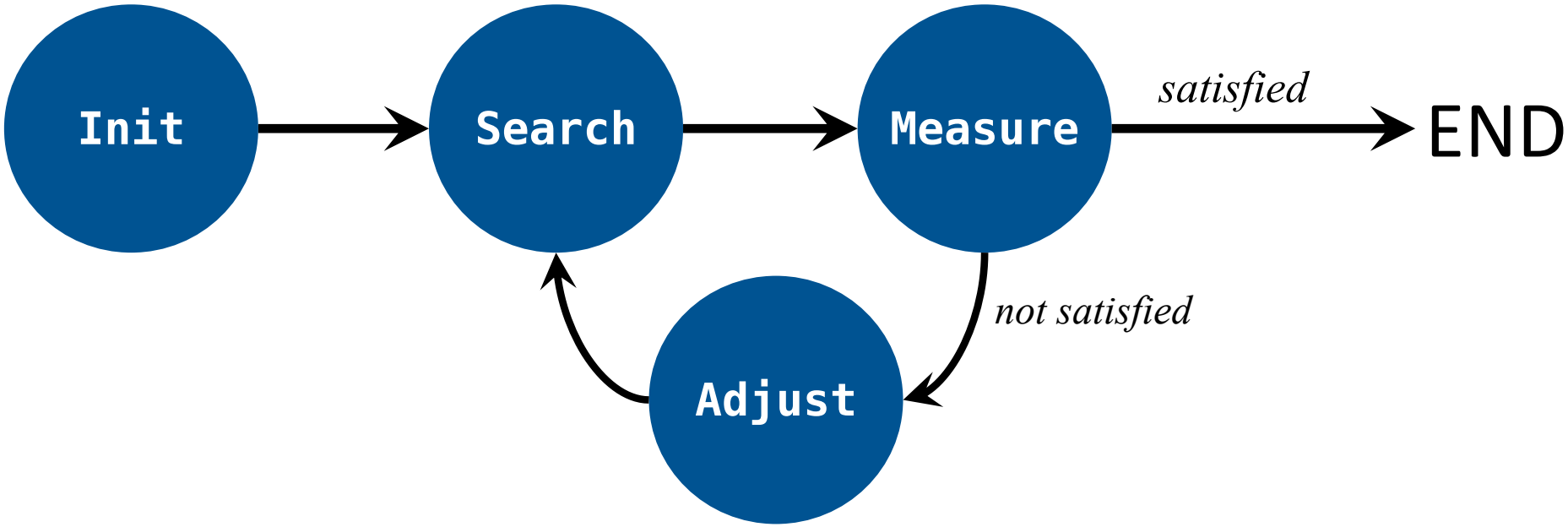
- ***Mistify* – framework for automated DNN model porting**
- **Decoupling and bridging DNN design and deployment**
- **Reducing manual efforts and computation overhead**

Mistify design

How *Mistify* addresses the challenges

- **Unscalable DNN tailoring needs**
  - Adaptation executor abstraction
  - Collective adaptation
- **Runtime dynamics**
  - Switching on multi-branch models
  - Model re-adaptation

Adaptation executor



Adaptation executor

Init

Embed adaptation logic, configure execution parameters, etc.

Search

Measure

Adjust

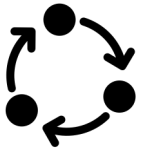
Adaptation executor

Init

Embed adaptation logic, configure execution parameters, etc.

Search

Core state: Run the actual DNN structure searching process for ~ iterations



Measure

Adjust

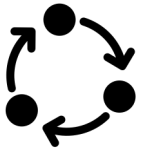
Adaptation executor

Init

Embed adaptation logic, configure execution parameters, etc.

Search

Core state: Run the actual DNN structure searching process for ~ iterations



Measure

Measuring the cost of the current DNN, and decide if ready to terminate.

Adjust

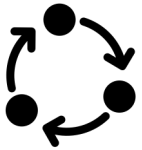
Adaptation executor

Init

Embed adaptation logic, configure execution parameters, etc.

Search

Core state: Run the actual DNN structure searching process for ~ iterations

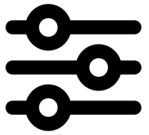


Measure

Measuring the cost of the current DNN, and decide if ready to terminate.

Adjust

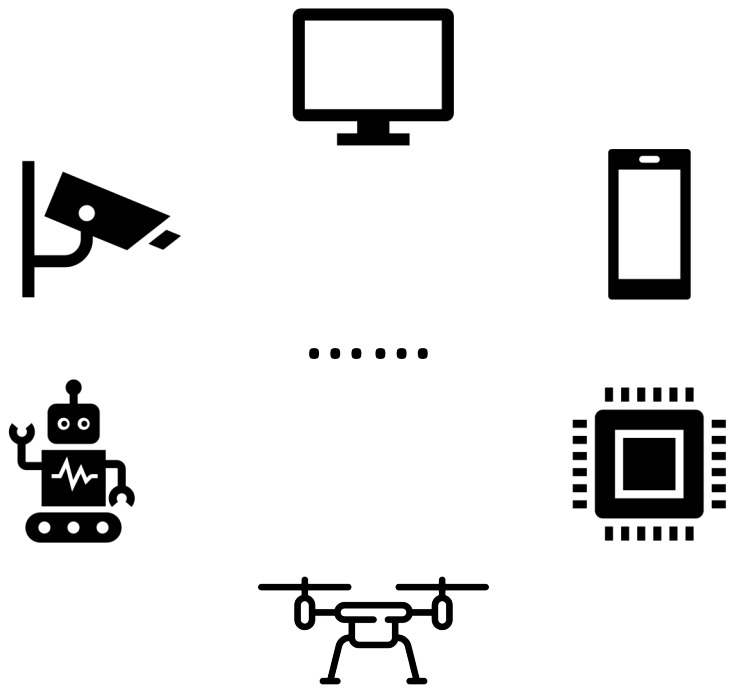
Adjust the parameters to control the searching algorithm behaviors



How *Mistify* addresses the challenges

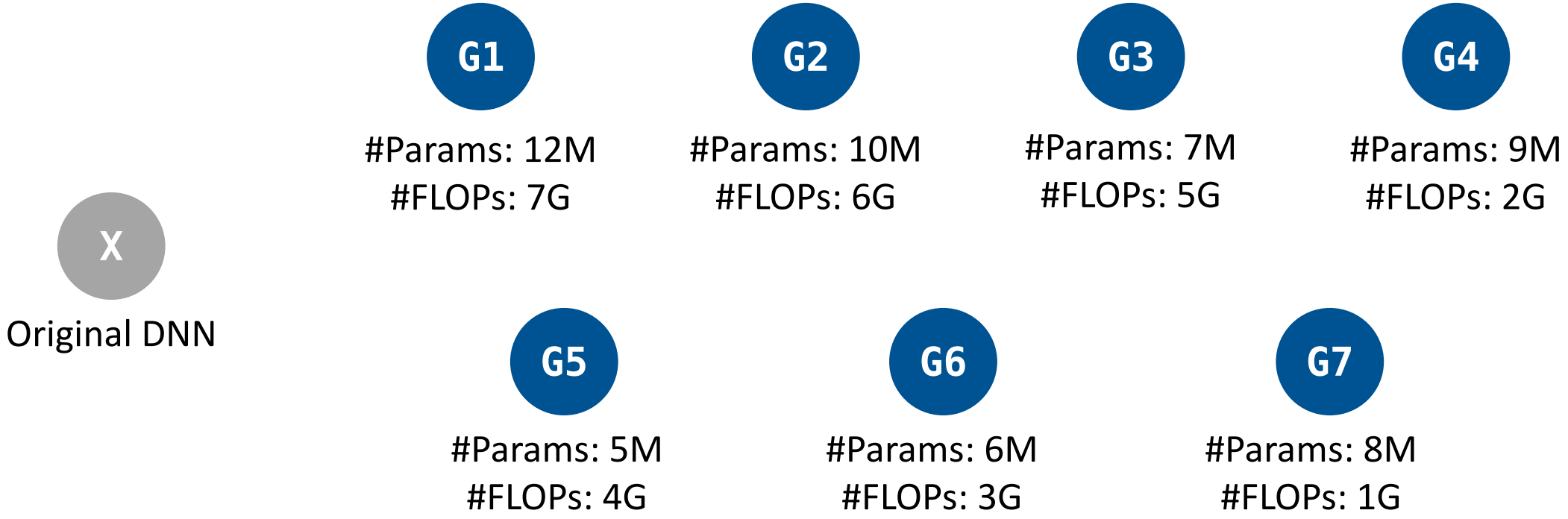
- **Unscalable DNN tailoring needs**
  - Adaptation executor abstraction – *minimizes manual efforts* 
  - Collective adaptation
- **Runtime dynamics**
  - Switching on multi-branch models
  - Model re-adaptation

Multiple adaptation goals

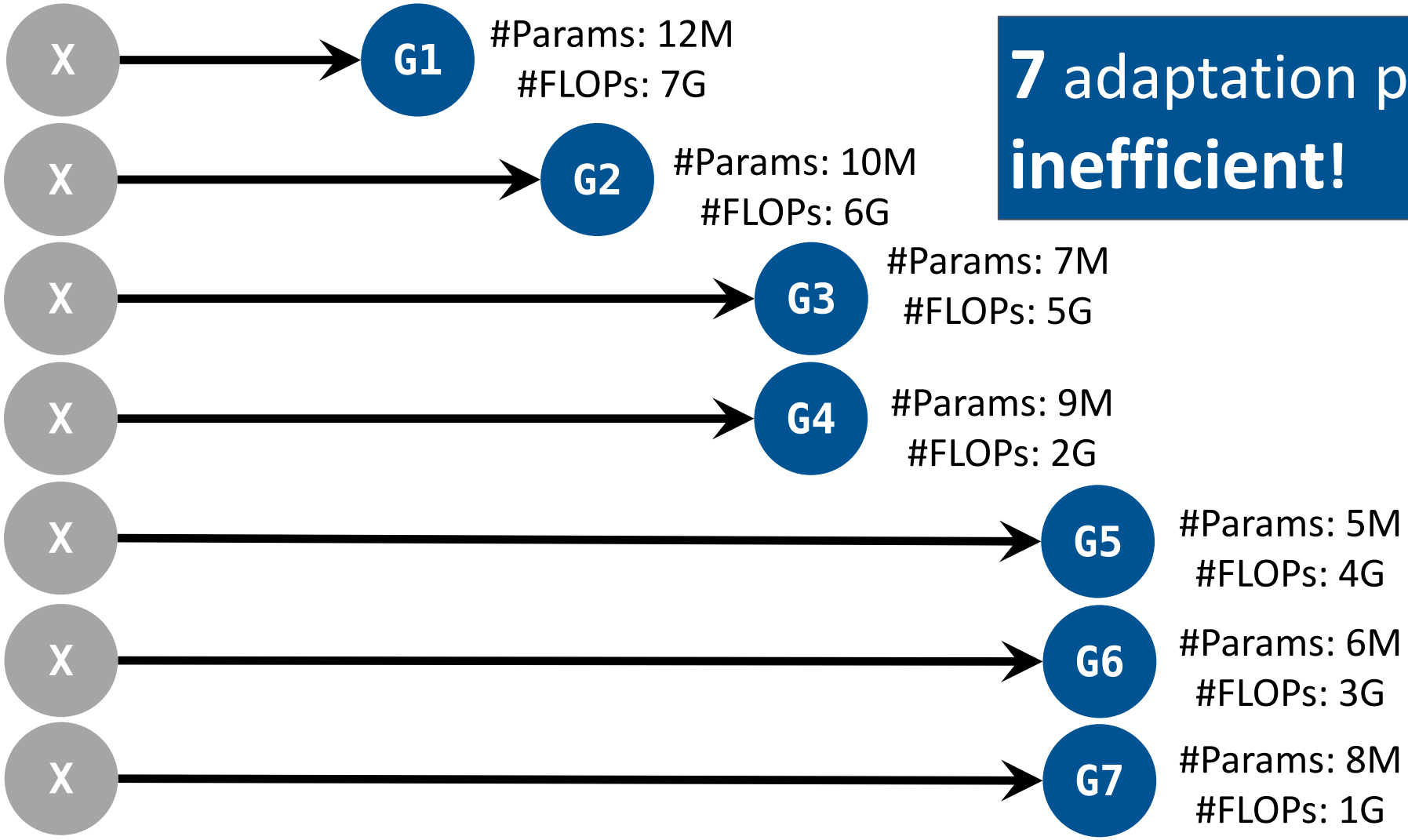


How to scale to a batch of simultaneous adaptations?

Multiple adaptation goals

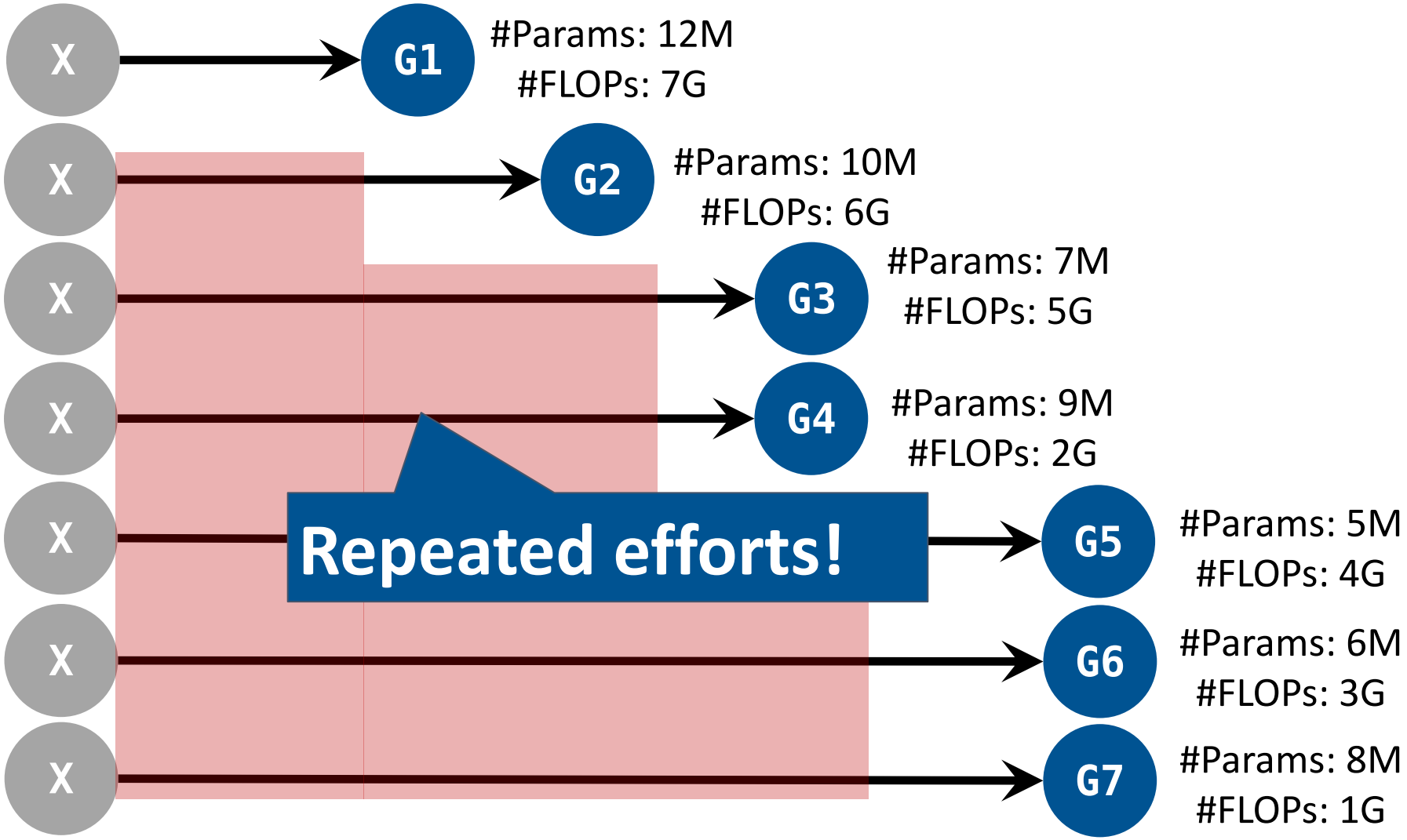


Adapt independently

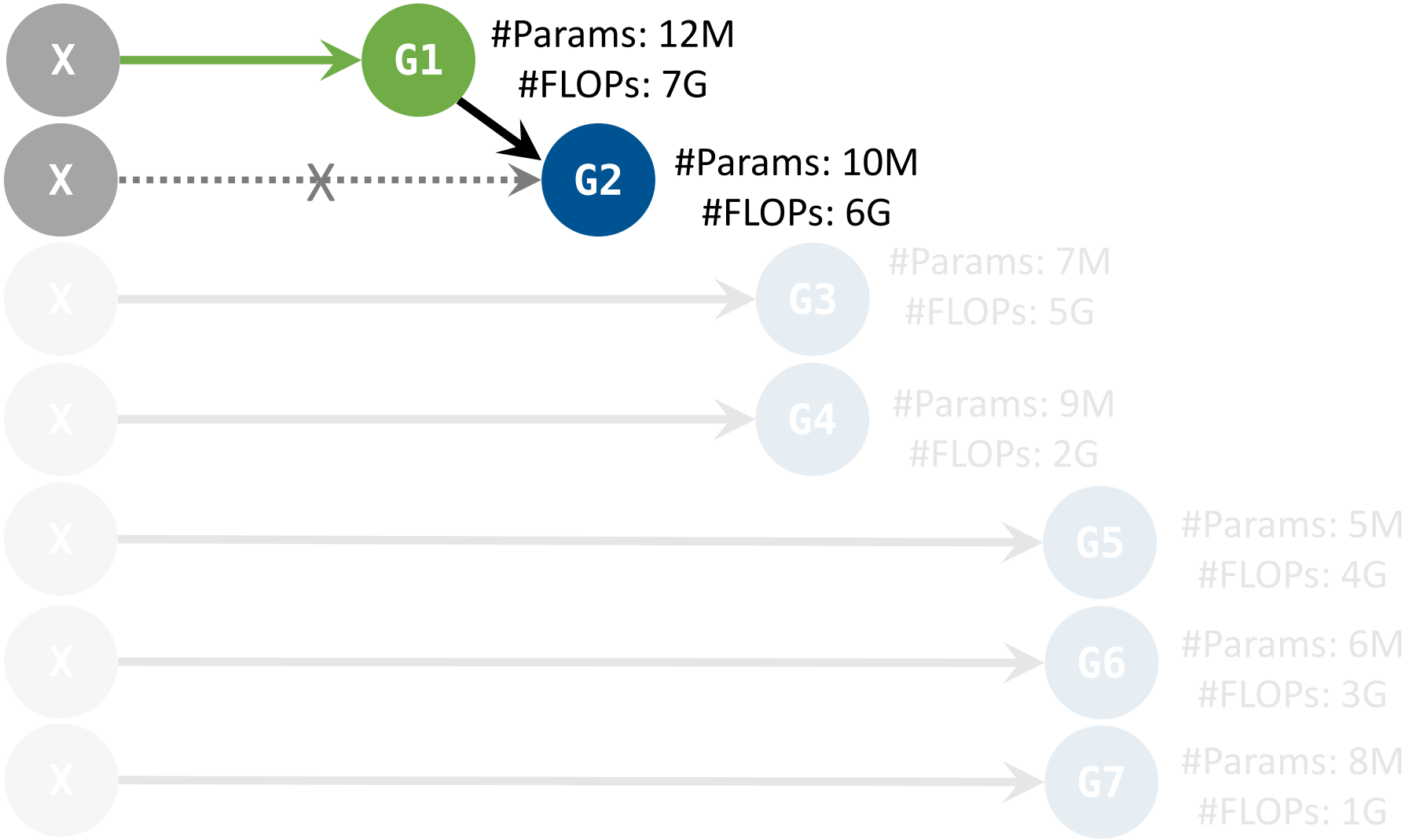


7 adaptation processes, **inefficient!**

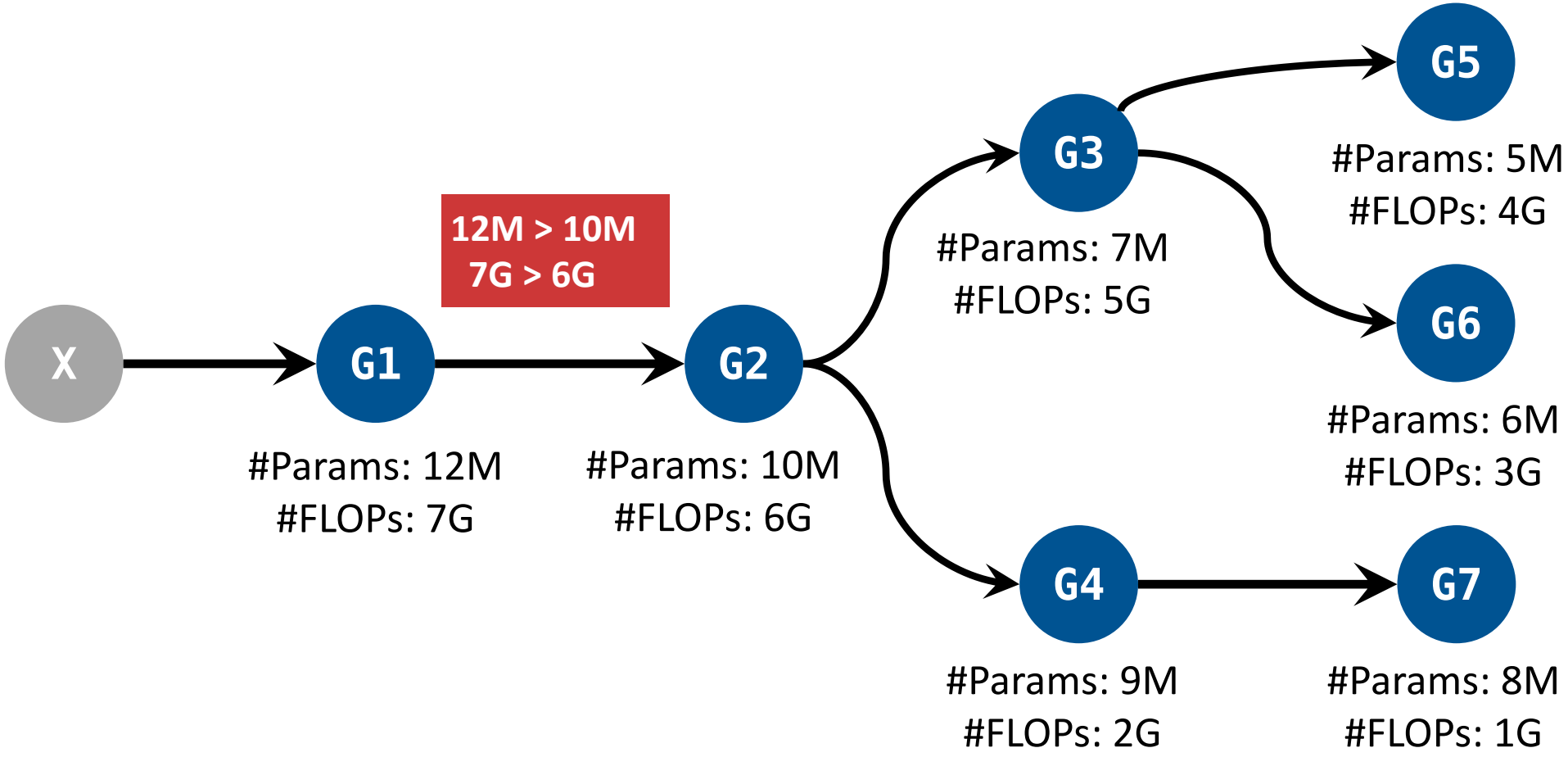
Adapt independently



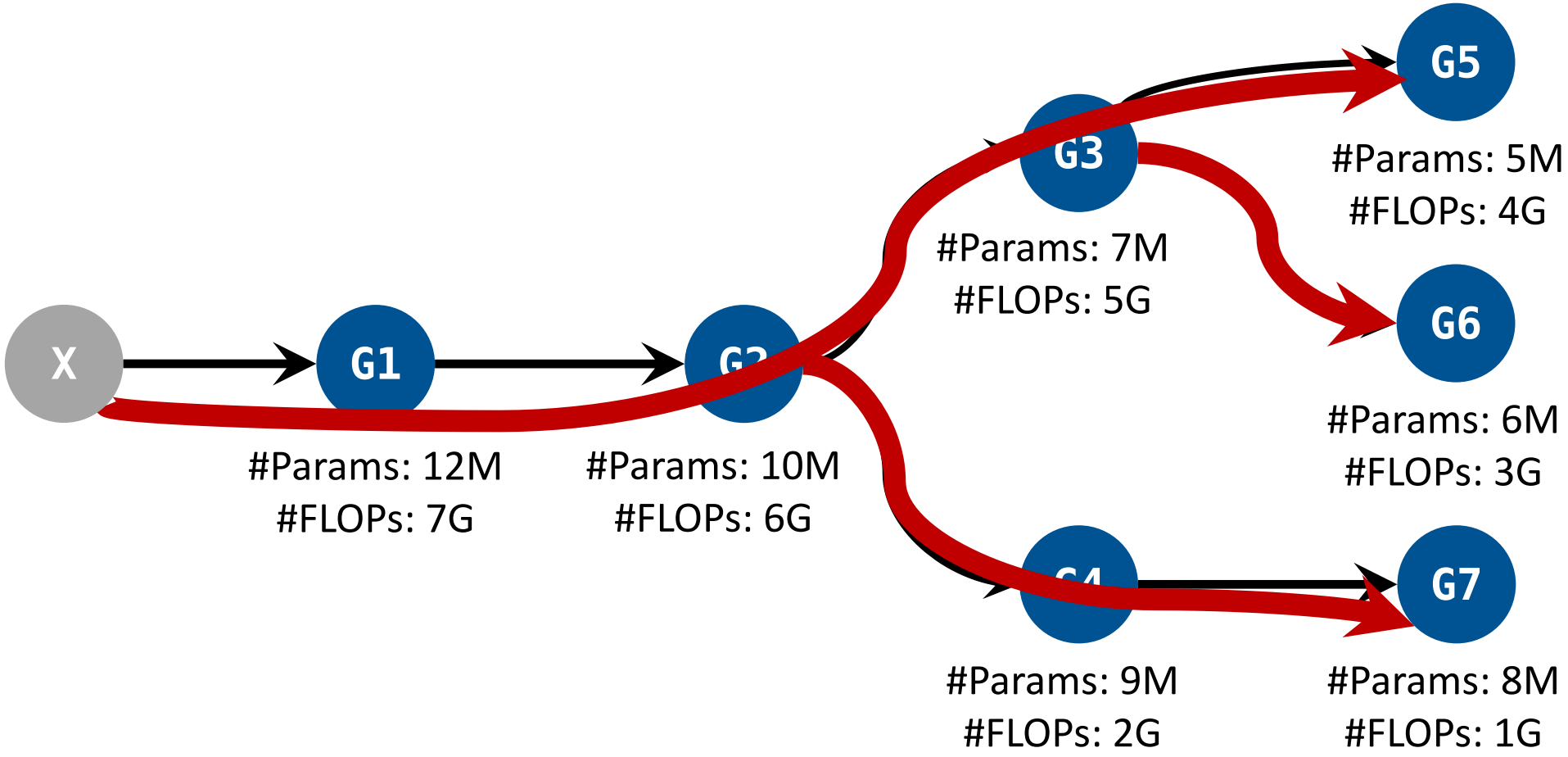
Adapt independently



Instead, collective adaptation



Instead, collective adaptation



Only **3** adaptation processes are needed

How *Mistify* addresses the challenges

- **Unscalable DNN tailoring needs**
  - Adaptation executor abstraction – *simplify manual efforts*
  - Collective adaptation – *scale with multiple adaptation processes*

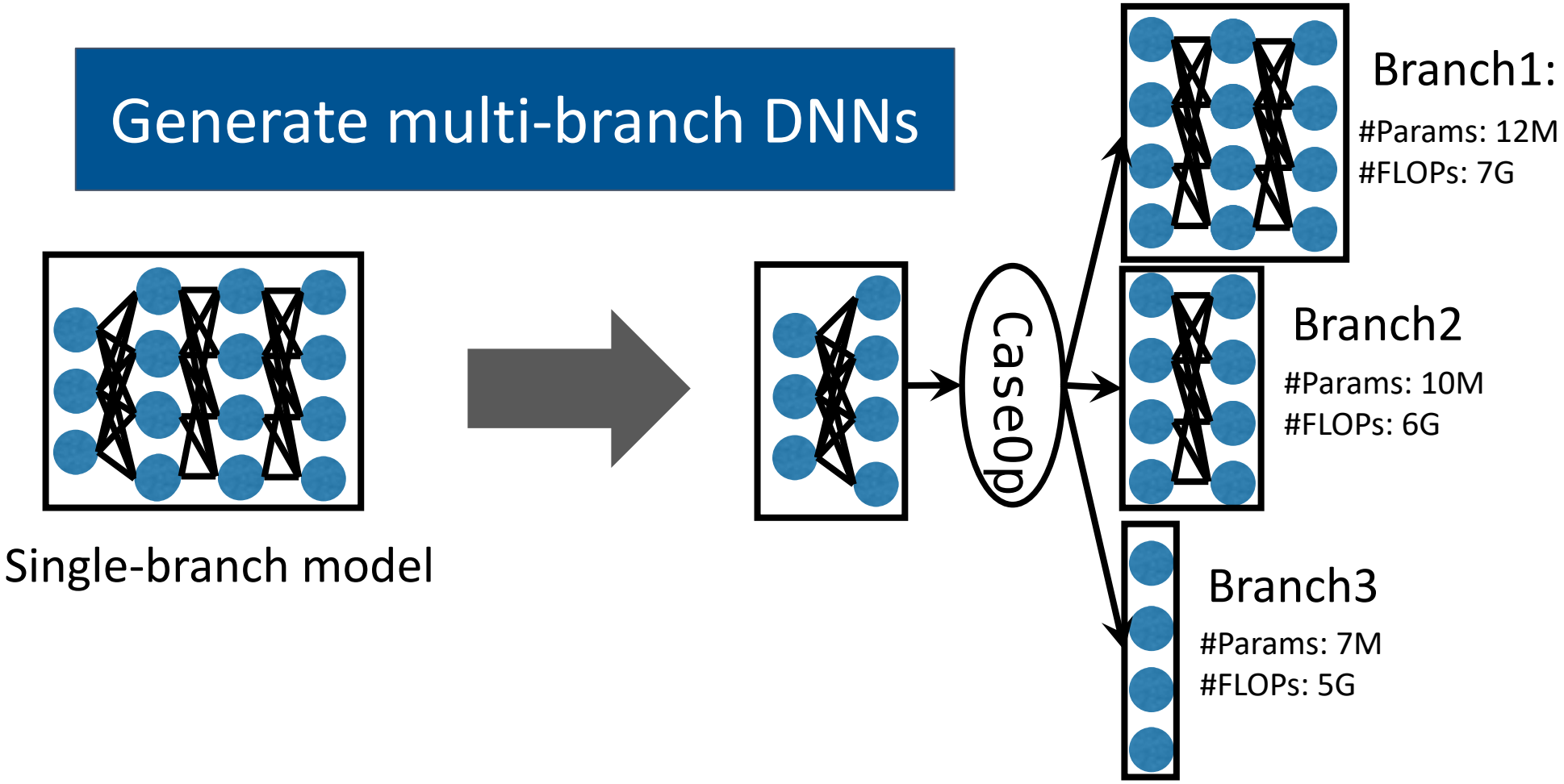
- **Runtime dynamics**
  - Switching on multi-branch models
  - Model re-adaptation

How to handle runtime dynamics?

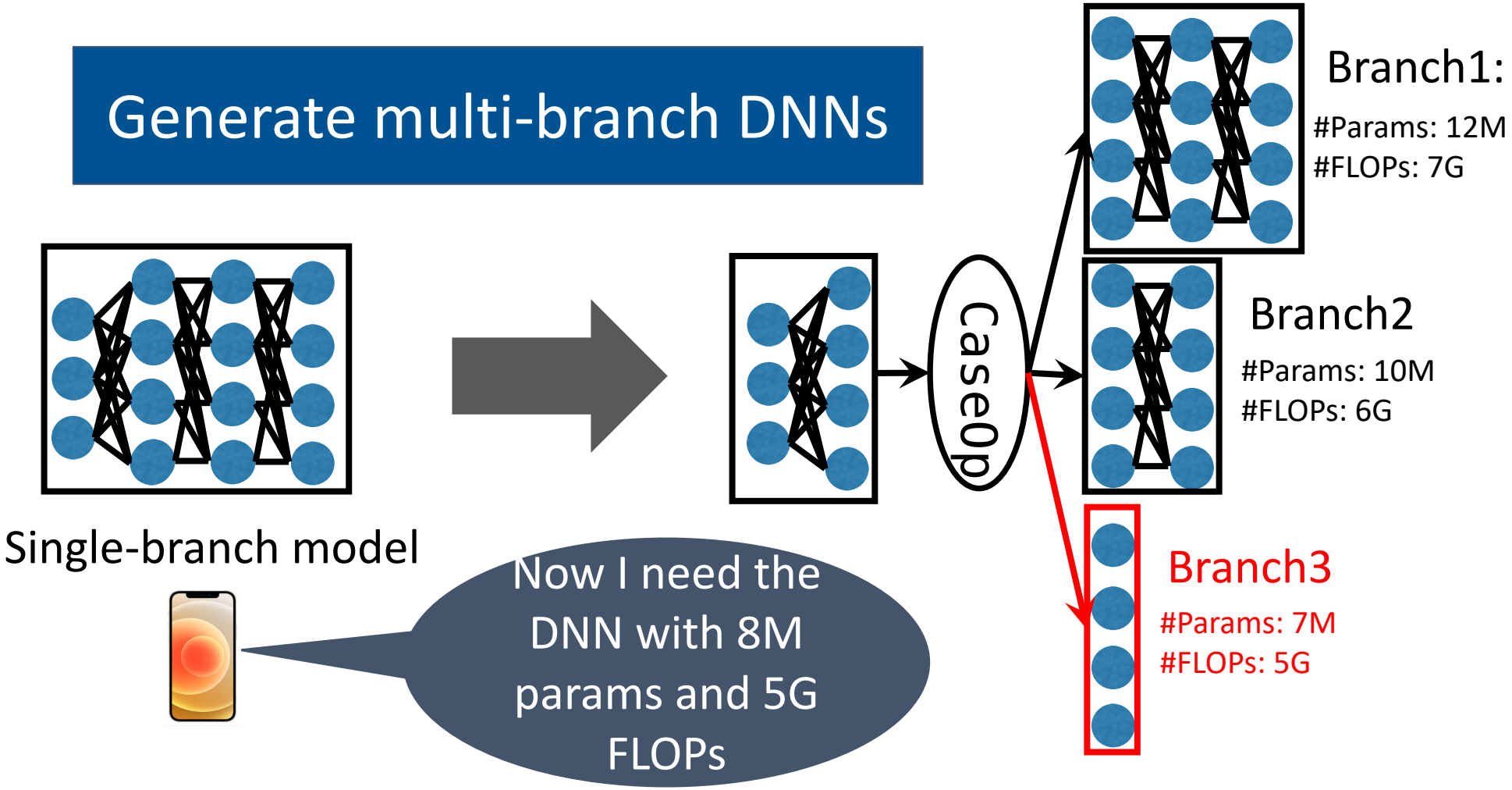
Foreground: switching on multi-branch DNNs

Background: on-demand model re-adaptation

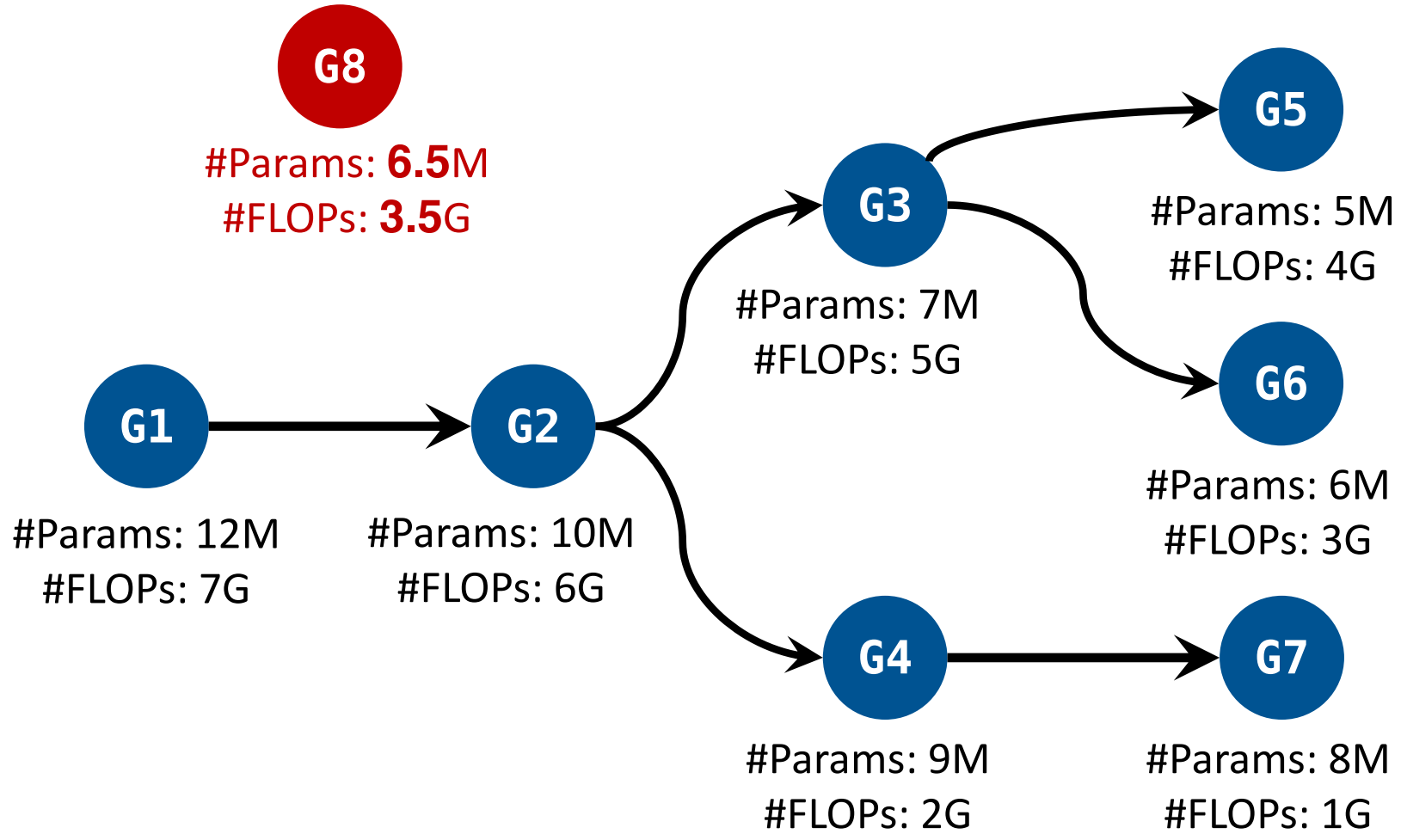
Foreground: branch switching



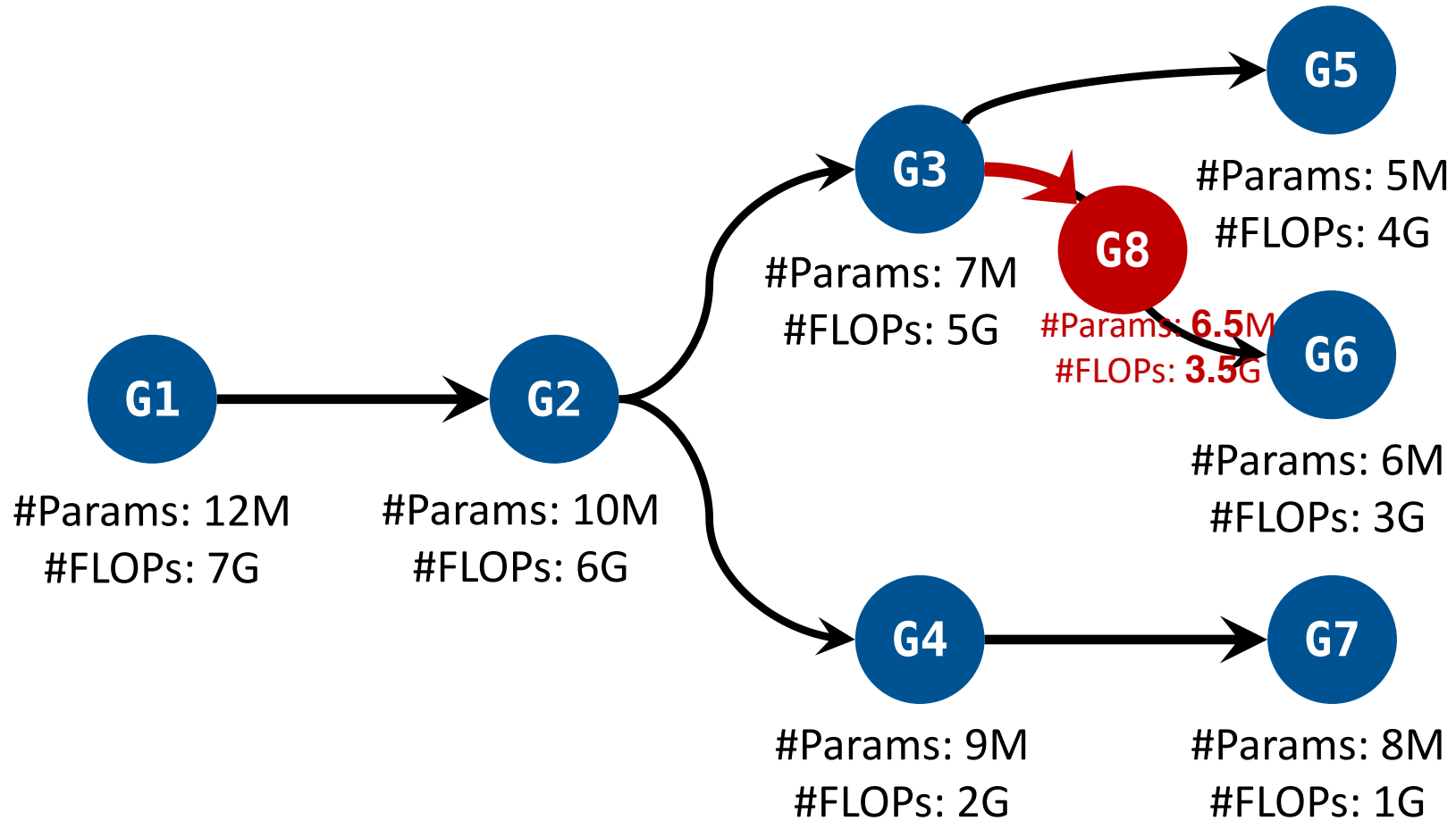
Foreground: branch switching



Background: re-adaptation



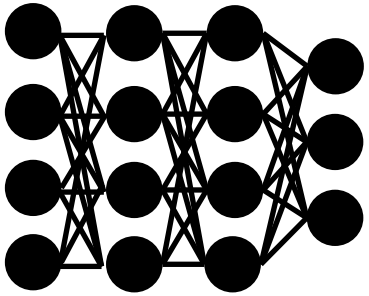
Background: re-adaptation



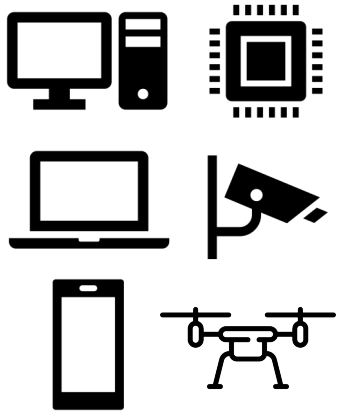
How *Mistify* addresses the challenges

- **Unscalable DNN tailoring needs** 
  - Adaptation executor abstraction – *simplify manual efforts*
  - Collective adaptation – *scale with multiple adaptation processes*
- **Runtime dynamics** 
  - Switching on multi-branch models
  - Model re-adaptation

Mistify system workflow

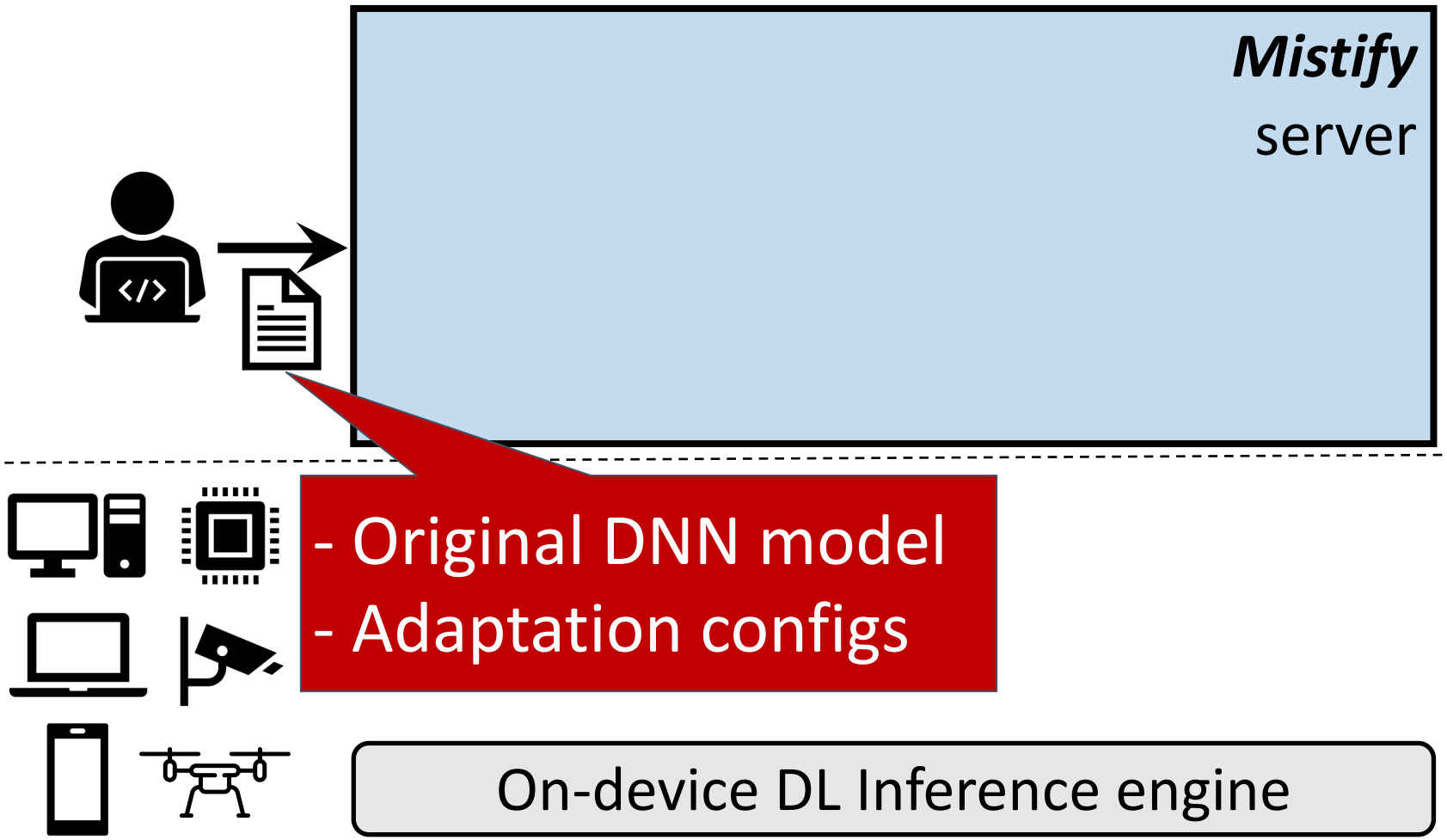


Original DNN model

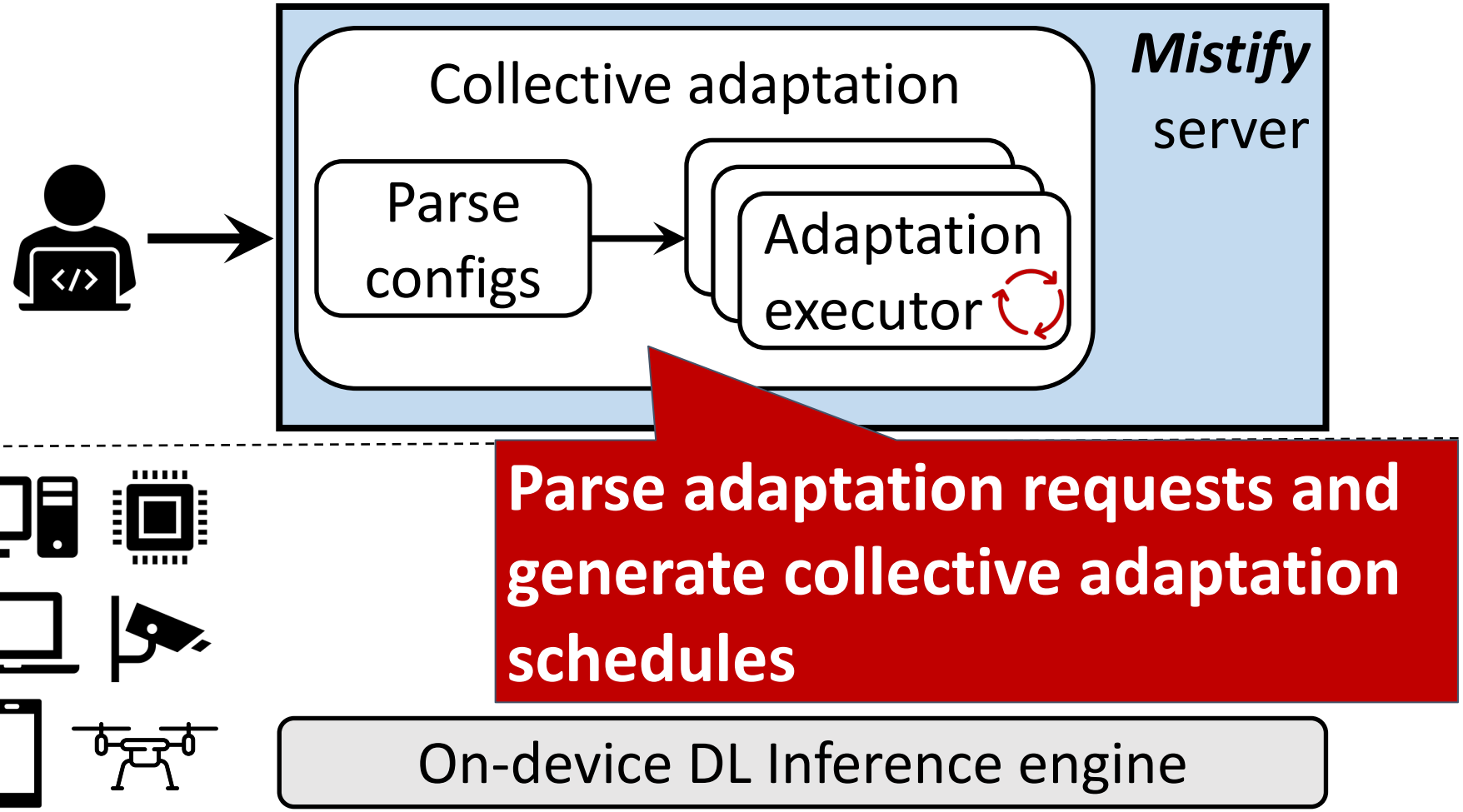


On-device DL Inference engine

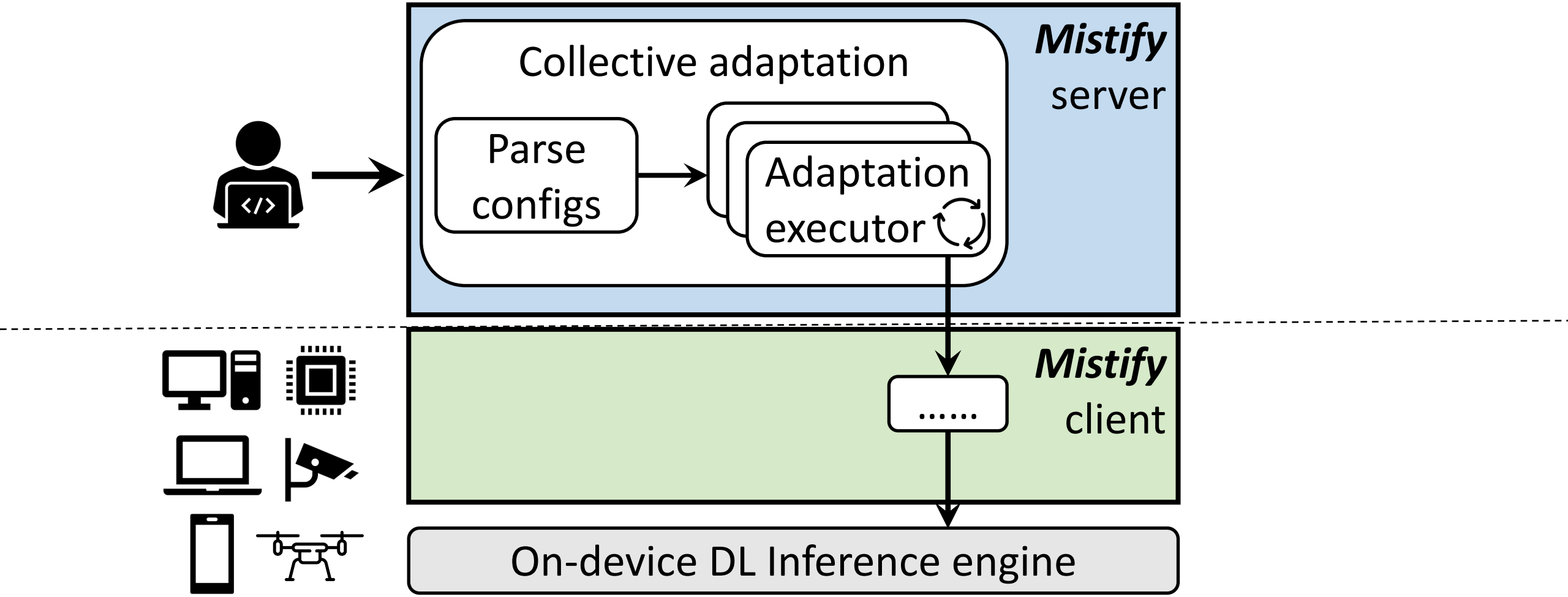
Mistify system workflow



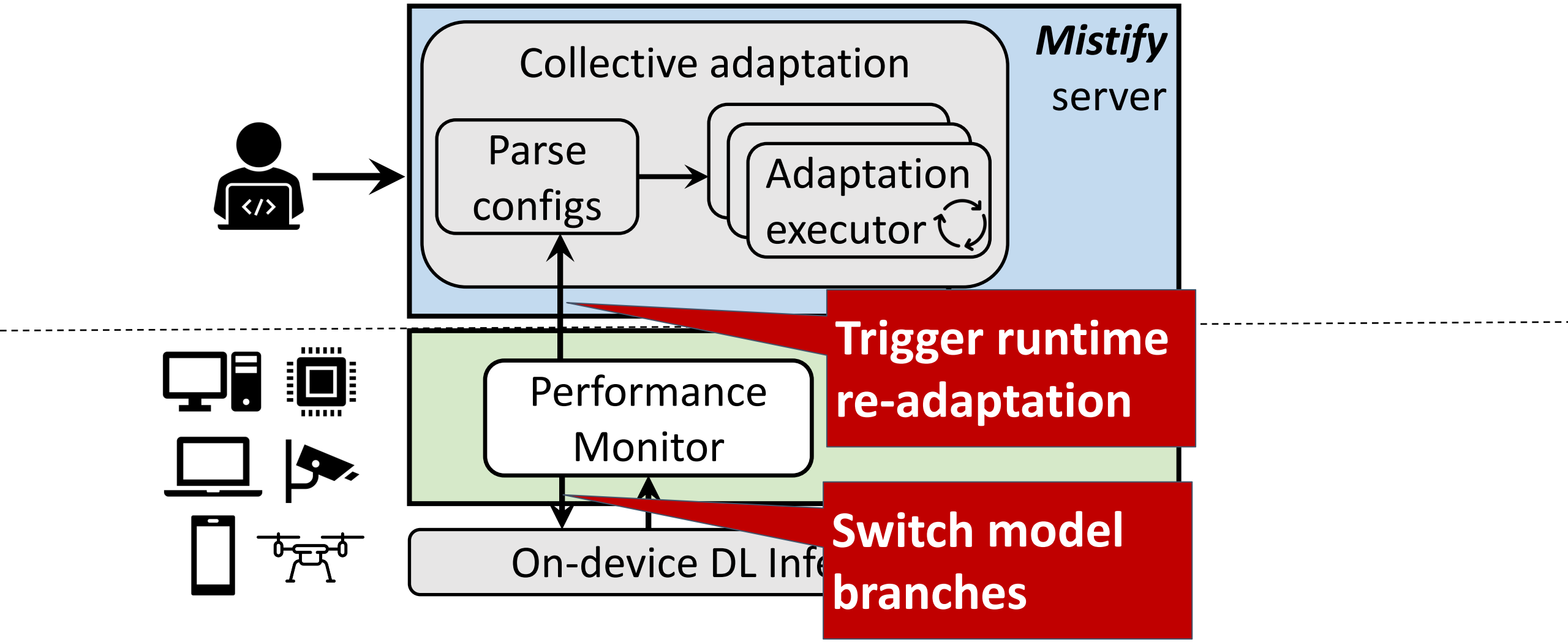
Mistify system workflow



Mistify system workflow



Mistify system workflow



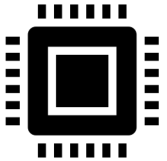
Mistify performance

General setup

Server

Linux server
with RTX 2070 GPU

Devices



Samsung S9
Google Edge TPU
Nvidia P600 GPU

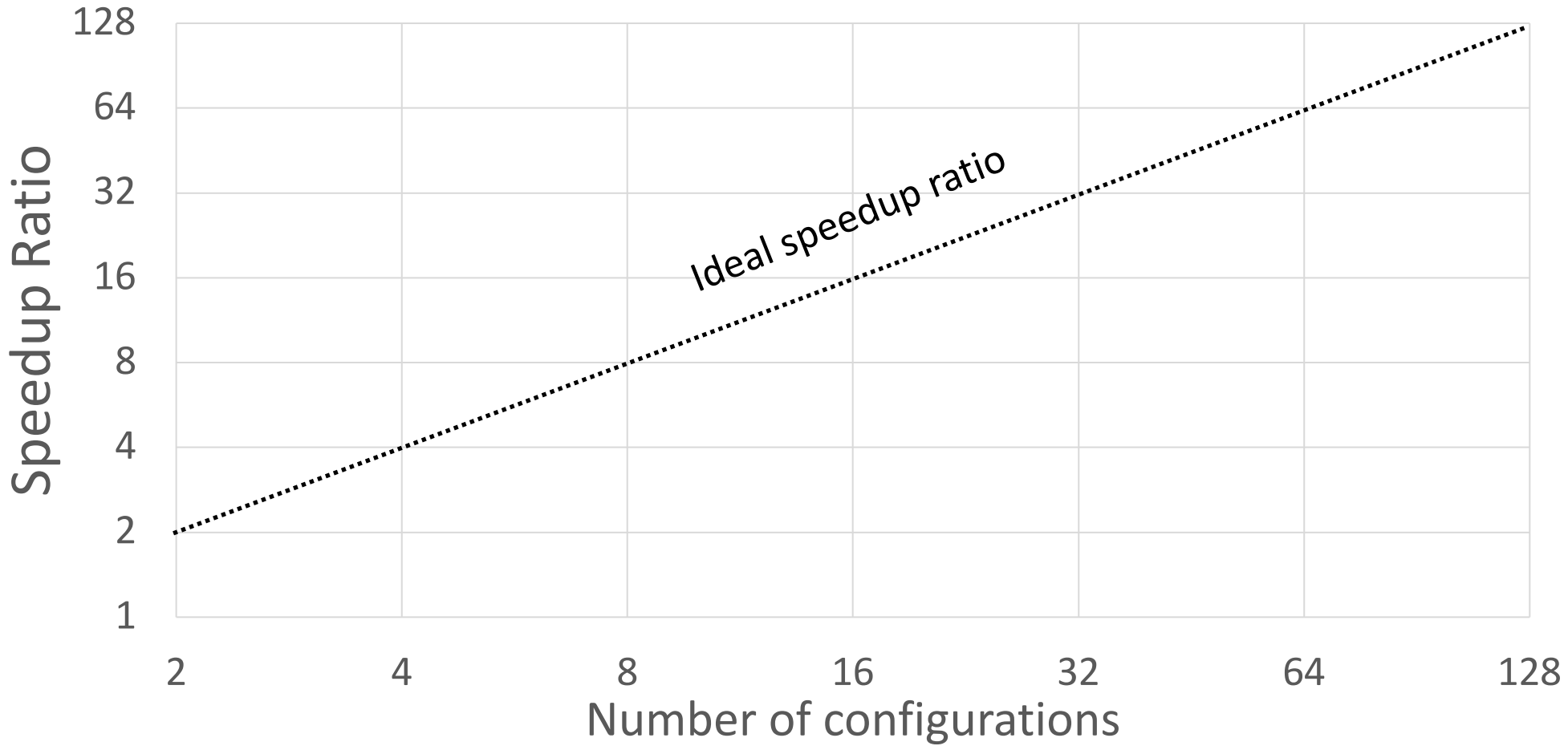
Models

- CV: MobileNet, ResNet50, ResNeXt101
- NLP: BiDAF, BERT

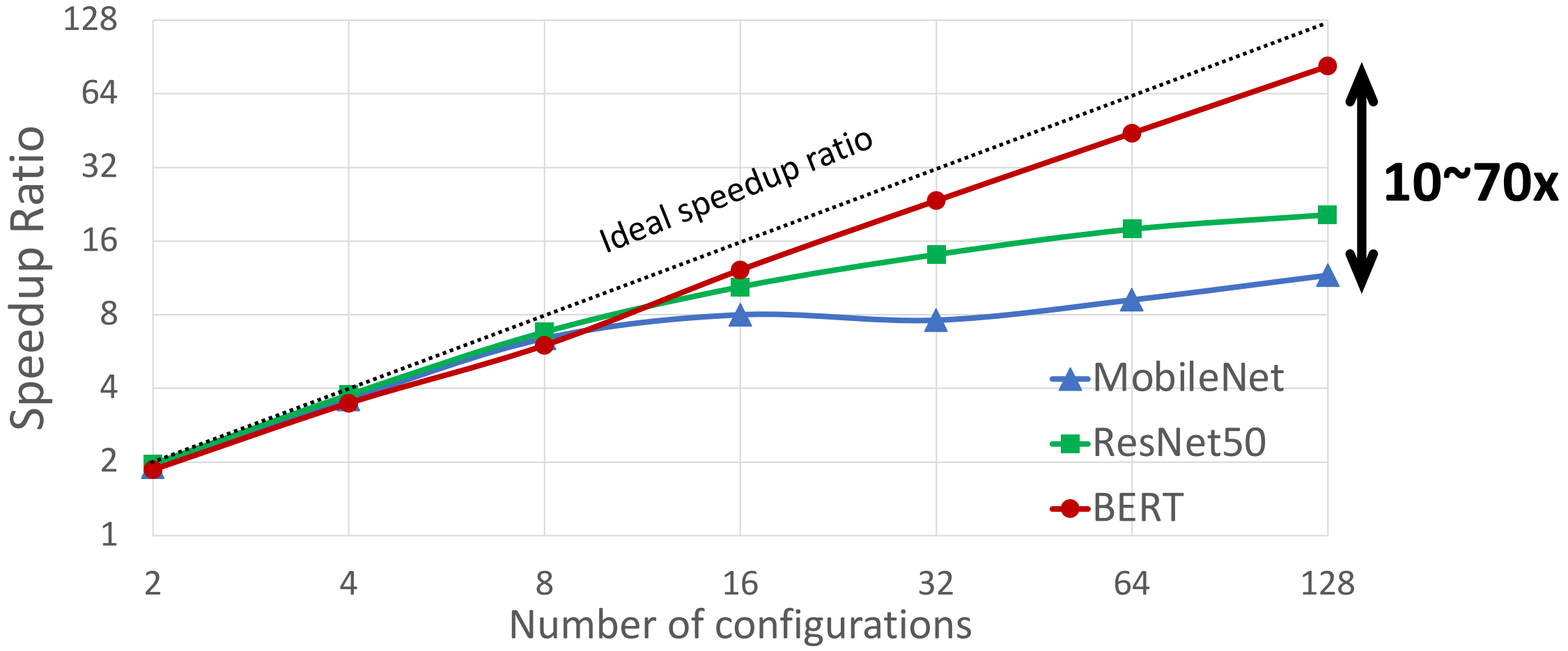
Workloads & datasets

- Image classification (ImageNet, Cifar100)
- Question & Answering (SQuADv1.1)

Scalability



Scalability



Minimizing manual efforts

Metrics	10 configurations			100 configurations		
	Manual	MorphNet	***Mistify***	Manual	MorphNet	***Mistify***

- **Implements** and **executes** the algorithm from scratch
- Adapts to each configuration **individually**

Minimizing manual efforts

Metrics	10 configurations			100 configurations		
	Manual	MorphNet	***Mistify***	Manual	MorphNet	***Mistify***

- **Annotates** adaptation logic and termination conditions
- Adapts to each configuration **individually**

Minimizing manual efforts

Metrics	10 configurations			100 configurations		
	Manual	MorphNet	***Mistify***	Manual	MorphNet	***Mistify***

- Fully **automated**
- Adapt to multiple configurations **collectively**

Minimizing manual efforts

Metrics	10 configurations			100 configurations		
	Manual	MorphNet	***Mistify***	Manual	MorphNet	***Mistify***
Lines of Code	>1k	138	**14**	>10k	782	**104**
Num of Files	30	12	**1**	300	102	**1**

- ***Orders of magnitude*** fewer lines of code changed
- ***Constant*** number of files changed

Minimizing manual efforts

Metrics	10 configurations			100 configurations		
	Manual	MorphNet	***Mistify***	Manual	MorphNet	***Mistify***
Lines of Code	>1k	138	14	>10k	782	104
Num of Files	30	12	1	300	102	1
Time (normalized)	10		**1.25**	100		**2.86**

Time: from linear to ***nearly constant***

Conclusion

Mistify – automated and scalable DNN porting service

- Decoupling DNN design and deployment and bridging them with an end-to-end framework
- Orders of magnitude reduction of computation overhead and manual efforts

Thank you