# AutoML for Object Detection

Xiangyu Zhang
MEGVII Research

# AutoML for Object Detection

1

- Advances in AutoML

2

- Search for Detection Systems

# AutoML for Object Detection

1

- **Advances in AutoML**

2

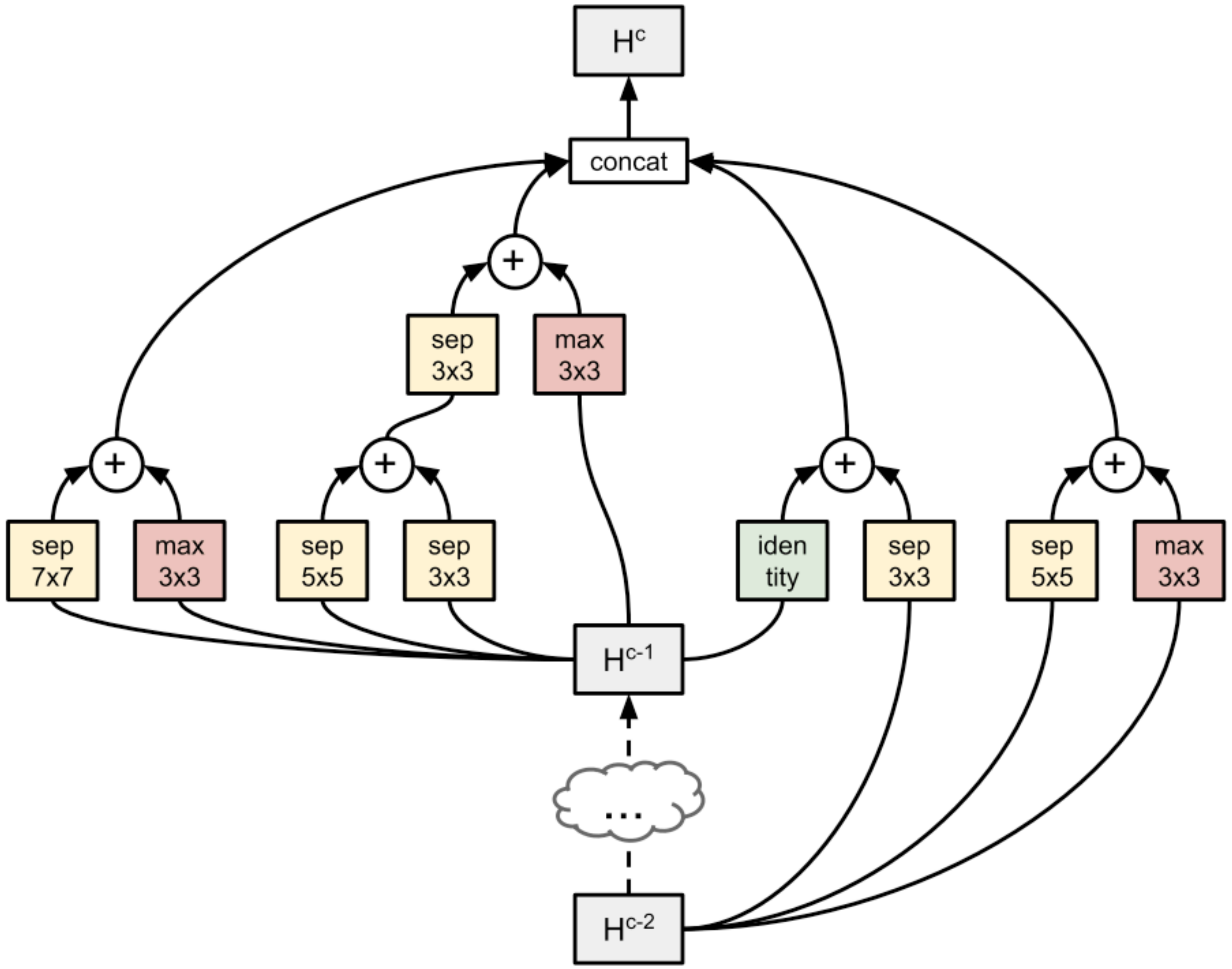
- Search for Detection Systems

# Introduction

MEGVII 旷视

- AutoML
  - A meta-approach to generate machine learning systems
  - Automatically search vs. manually design
- AutoML for Deep Learning
  - Neural architecture search (NAS)
  - Hyper-parameters turning
  - Loss function
  - Data augmentation
  - Activation function
  - Backpropagation

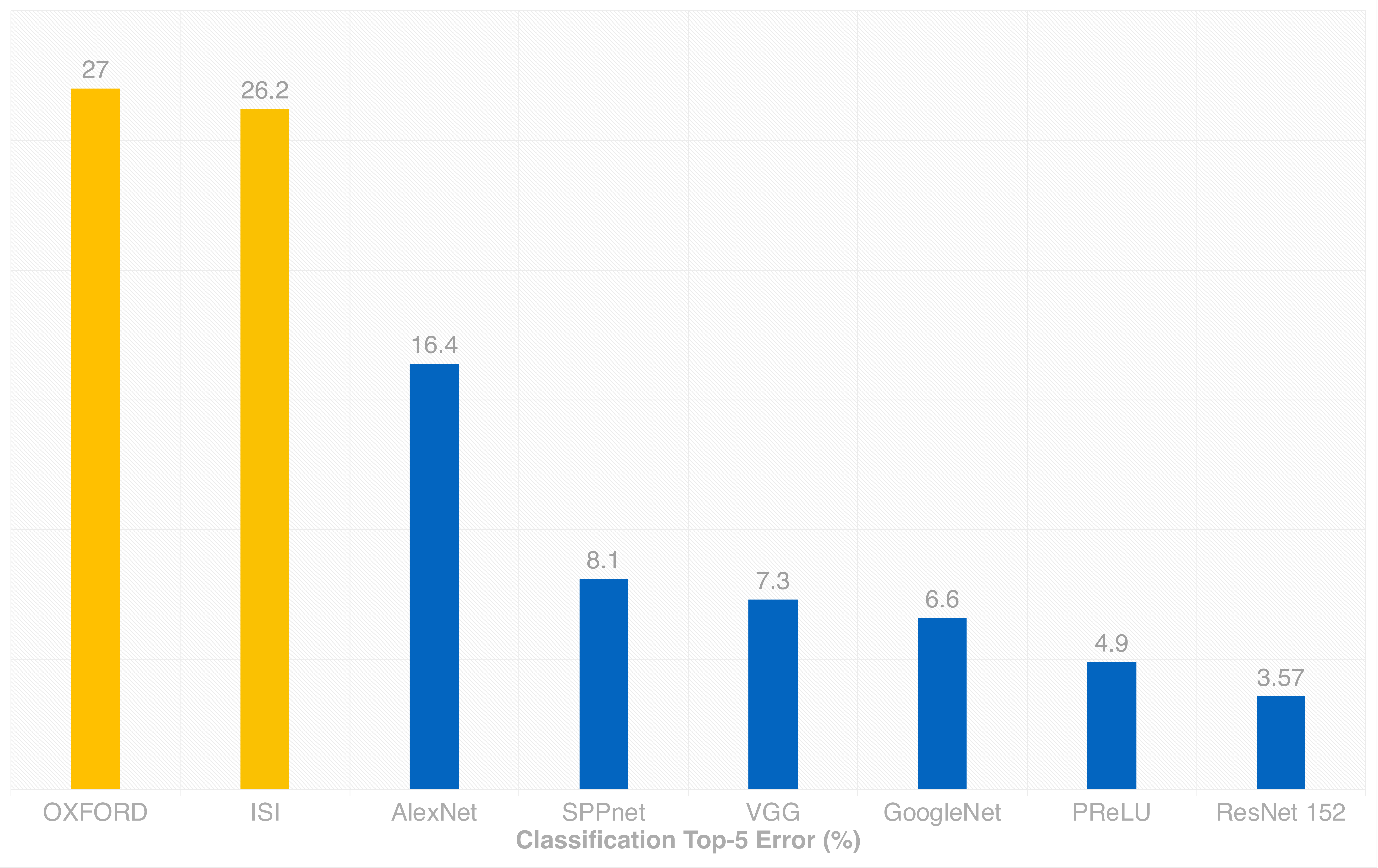
…



# Revolution of AutoML

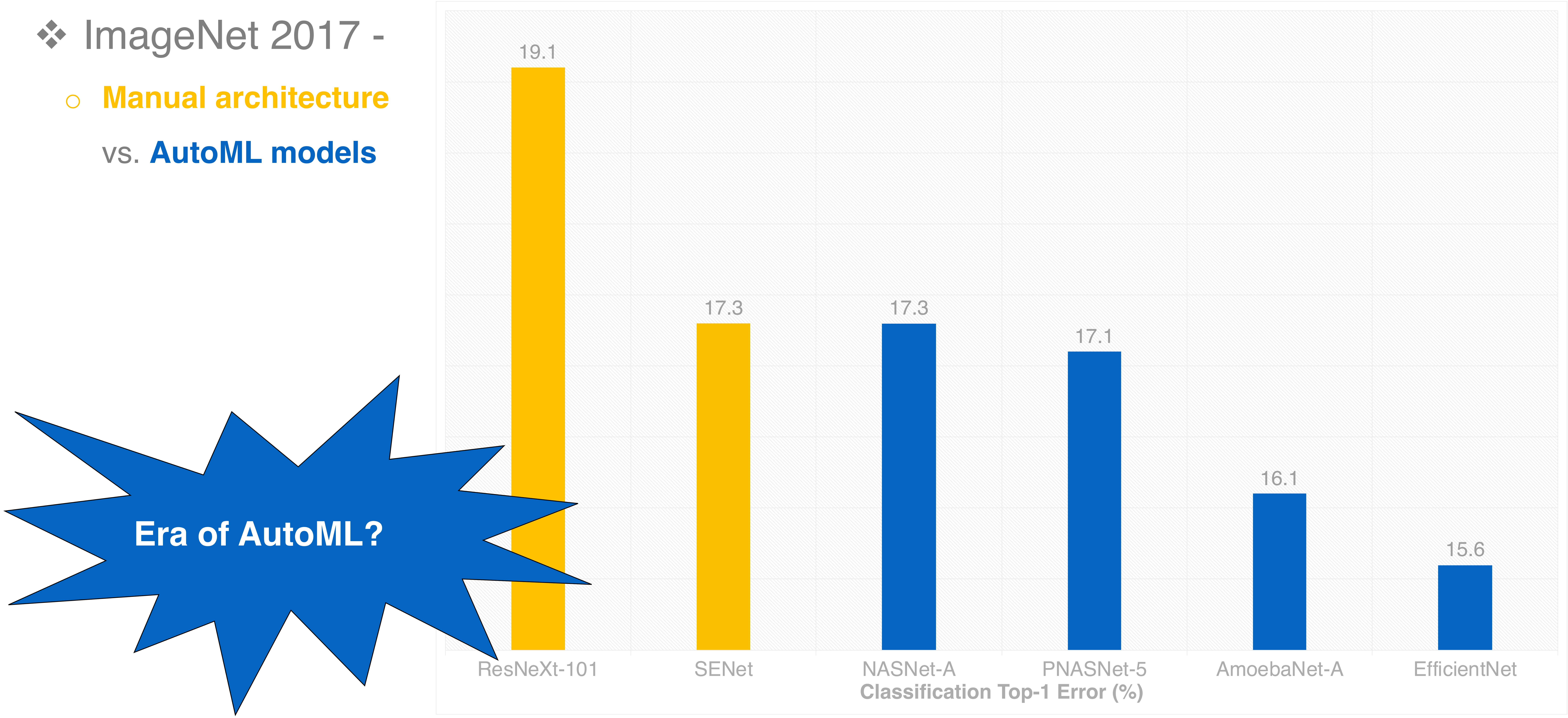
MEGVII 旷视

- ImageNet 2012 -
  - **Hand-craft feature** vs. **deep learning**
- Era of **Deep Learning** begins!



# Revolution of AutoML (cont’d)

- ImageNet 2017 -
  - Manual architecture vs. AutoML models



# Revolution of AutoML (cont’d)

MEGVII 旷视

- Literature
  - 200+ since 2017



LITERATURE ON NEURAL ARCHITECTURE SEARCH

The following list considers papers related to neural architecture search. It is by no means a complete list. If you miss a paper on the list, please let us know.

***Update (Dec 2018):*** Since the list is already quite long by now, we will highlight papers accepted at conferences and journals in the future. This should hopefully provide some guidance towards high-quality papers.

- Architecture Search (and Hyperparameter Optimization):
  - **Surrogate-Assisted Evolutionary Deep Learning Using an End-to-End Random Forest-based Performance Predictor** (Sun et al. 2019; accepted by IEEE Transactions on Evolutionary Computation)
    https://ieeexplore.ieee.org/document/8744404
  - **Adaptive Genomic Evolution of Neural Network Topologies (AGENT) for State-to-Action Mapping in Autonomous Agents** (Behjat et al. 2019; accepted and presented in ICRA 2019)
    https://arxiv.org/abs/1903.07107
  - Densely Connected Search Space for More Flexible Neural Architecture Search (Fang et al. 2019)
    https://arxiv.org/abs/1906.09607
  - SwiftNet: Using Graph Propagation as Meta-knowledge to Search Highly Representative Neural Architectures (Cheng et al. 2019)
    https://arxiv.org/abs/1906.08305
  - Transfer NAS: Knowledge Transfer between Search Spaces with Transformer Agents (Borsos et al. 2019)
    https://arxiv.org/abs/1906.08102
  - XNAS: Neural Architecture Search with Expert Advice (Nayman et al. 2019)
    https://arxiv.org/abs/1906.08031
  - A Study of the Learning Progress in Neural Architecture Search Techniques (Singh et al. 2019)

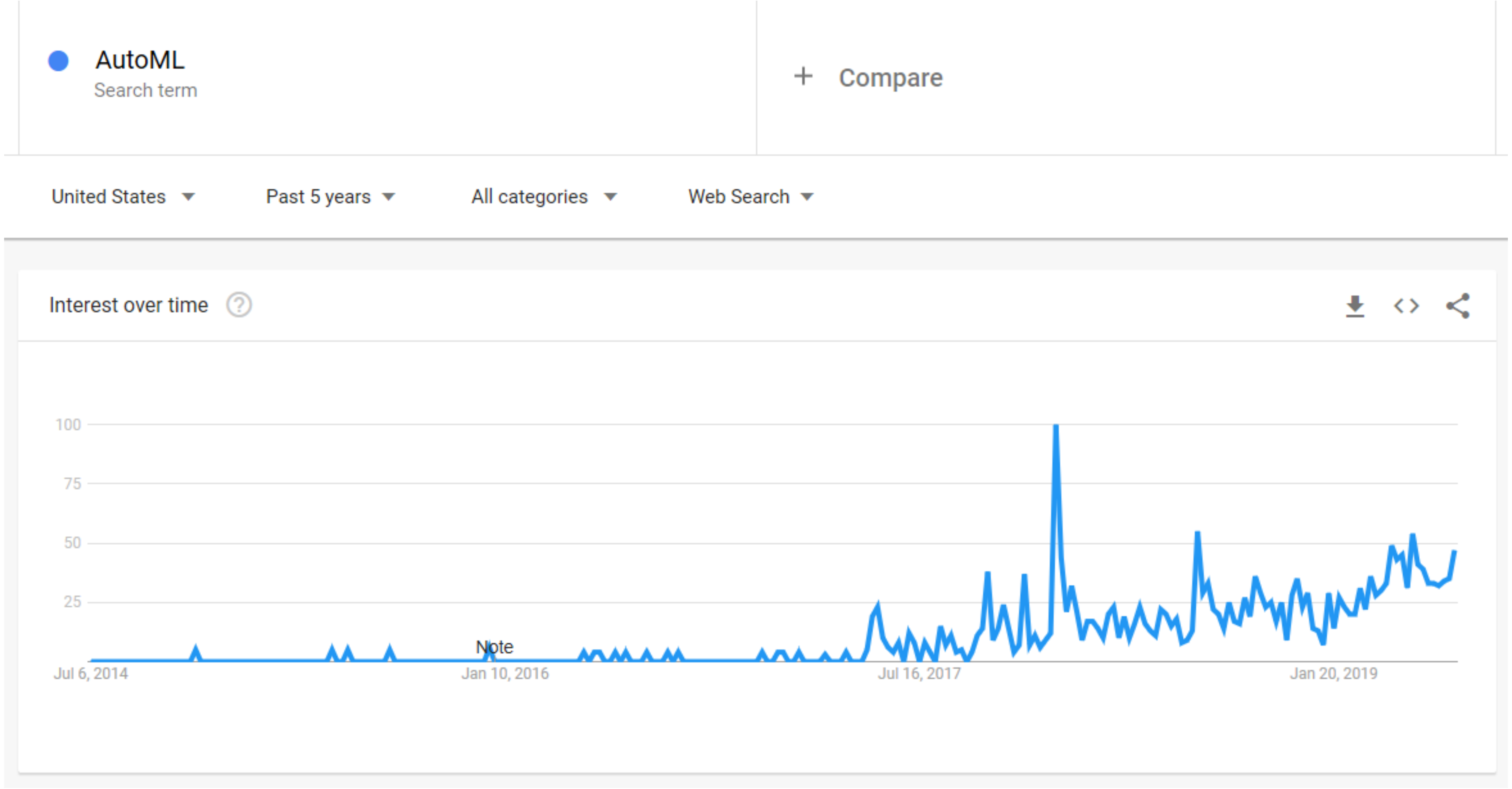
# Revolution of AutoML (cont’d)

MEGVII 旷视

❖ Literature

- 200+ since 2017

❖ Google Trends

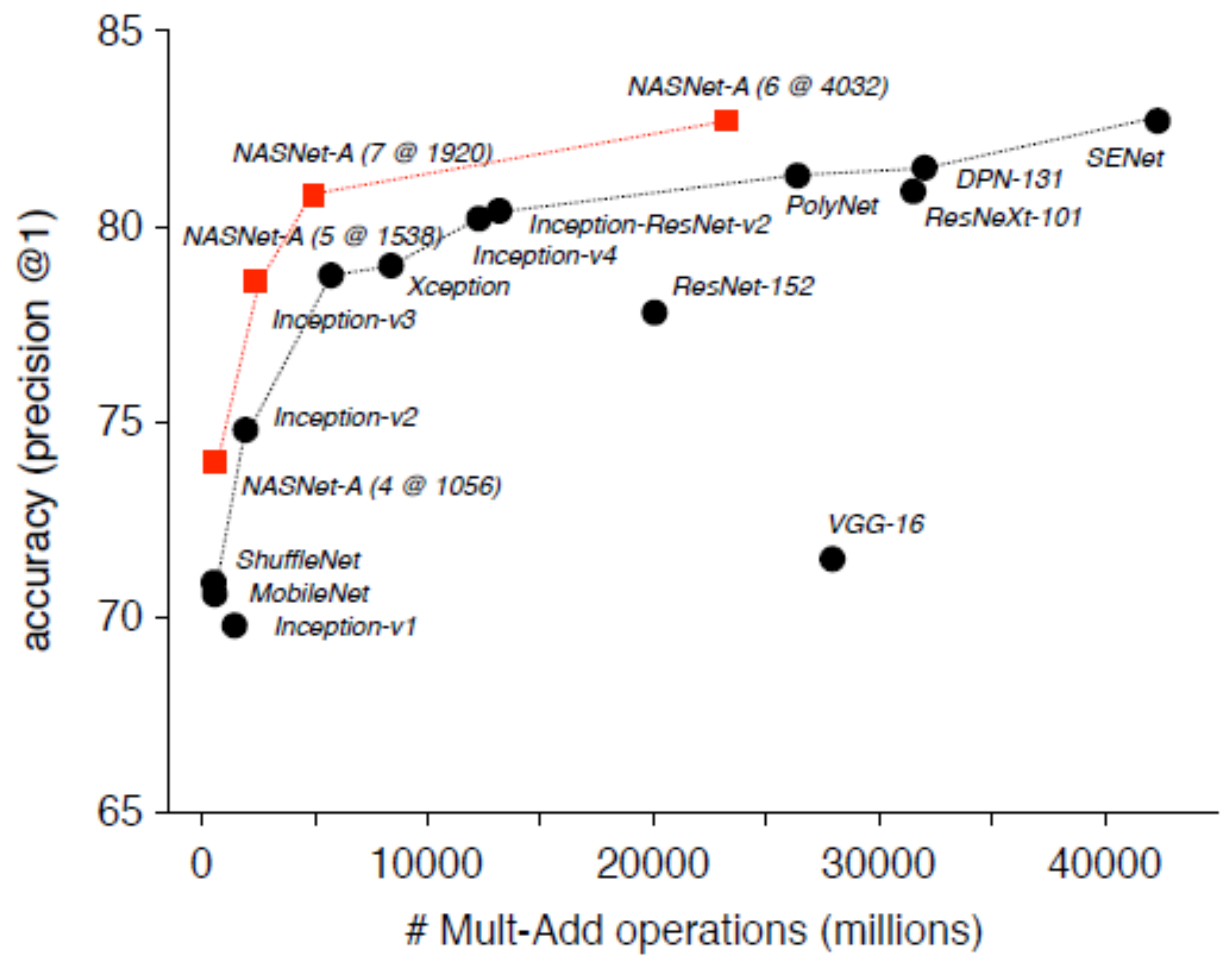


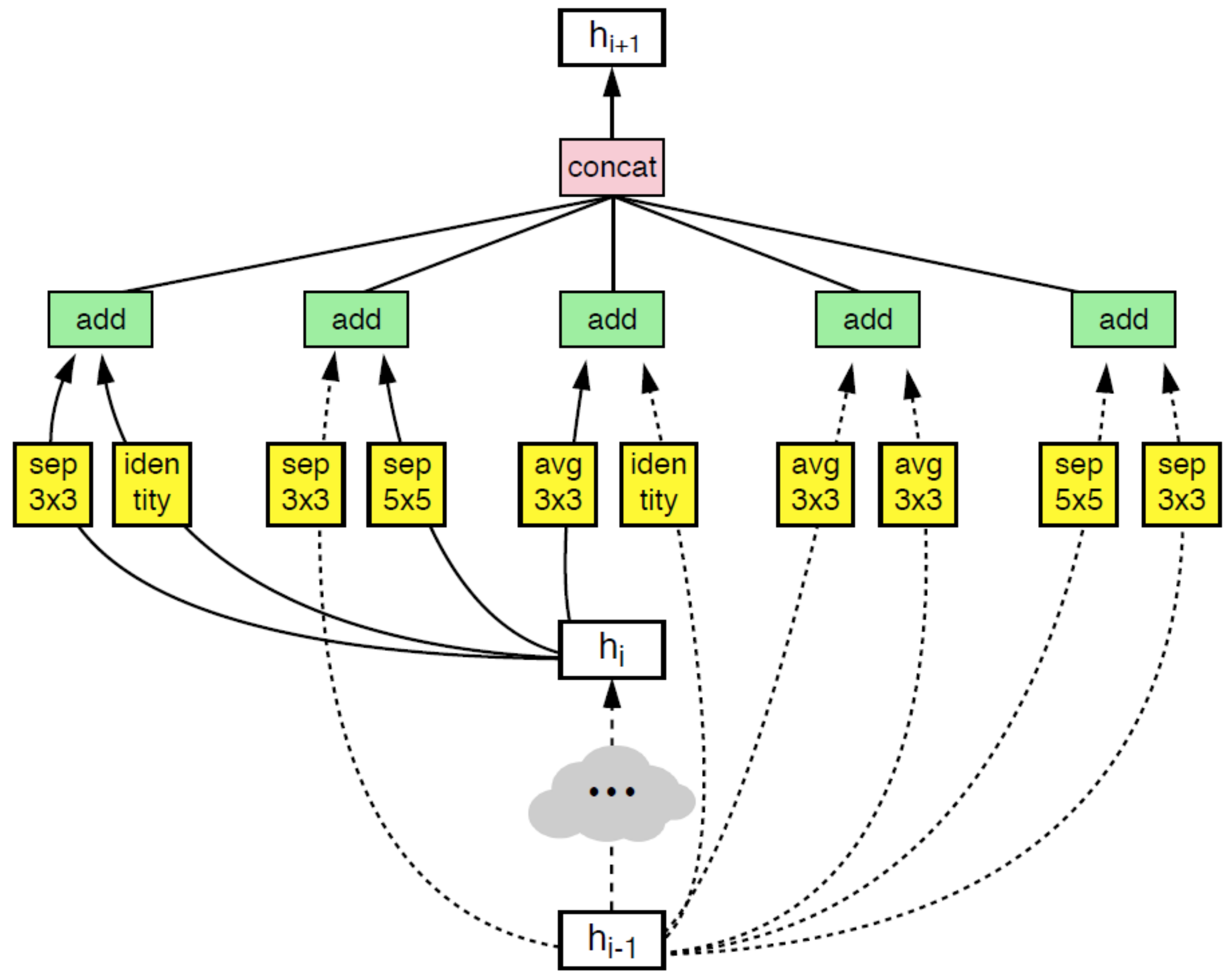
# Recent Advances in AutoML (1)

MEGVII 旷视

❖ Surpassing handcraft models
- NASNet

❖ Keynotes
- RNN controller + policy gradient
- Flexible search space
- Proxy task needed




Zoph et al. Learning Transferable Architectures for Scalable Image Recognition
Zoph et al. Neural Architecture Search with Reinforcement Learning

# Recent Advances in AutoML (2)

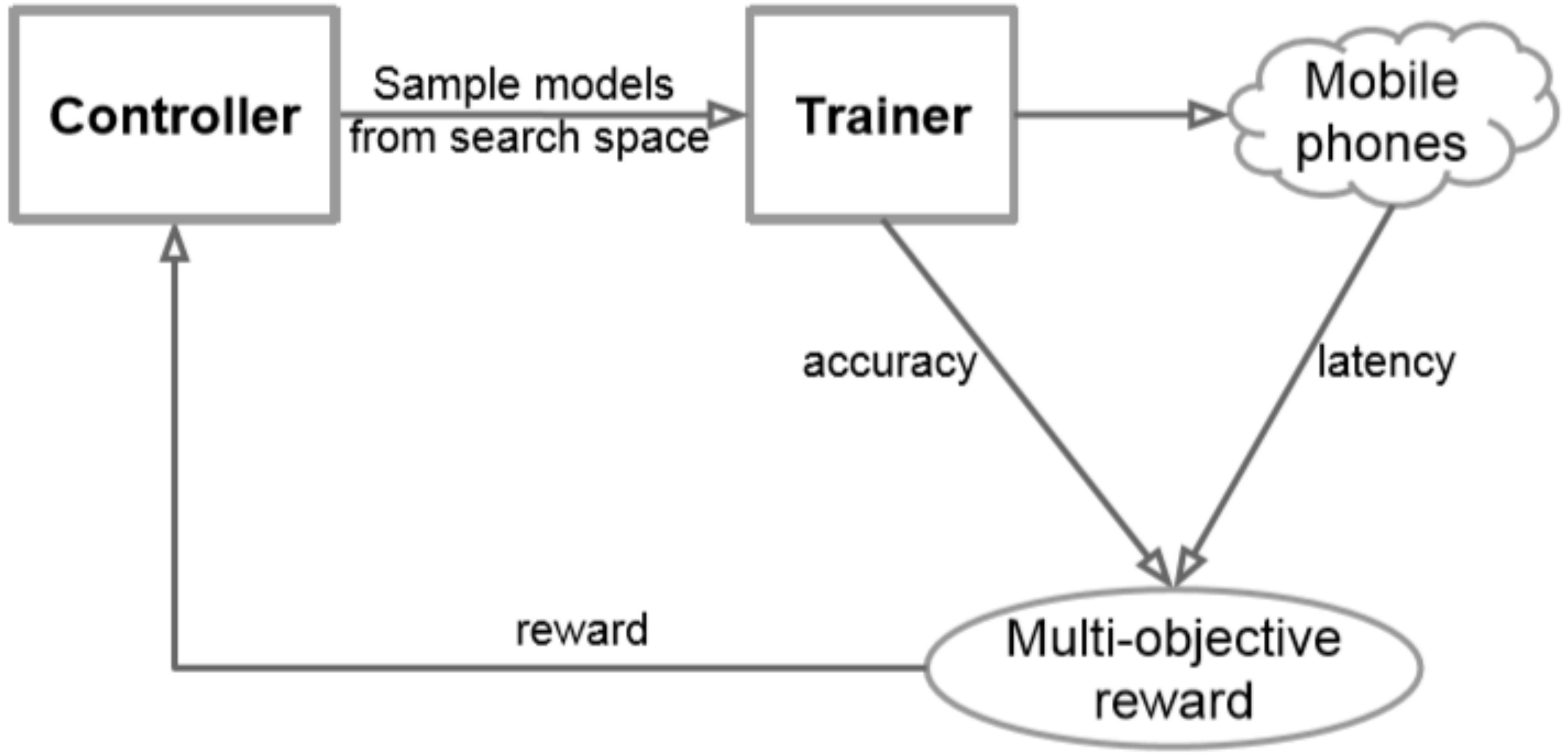
MEGVII 旷视

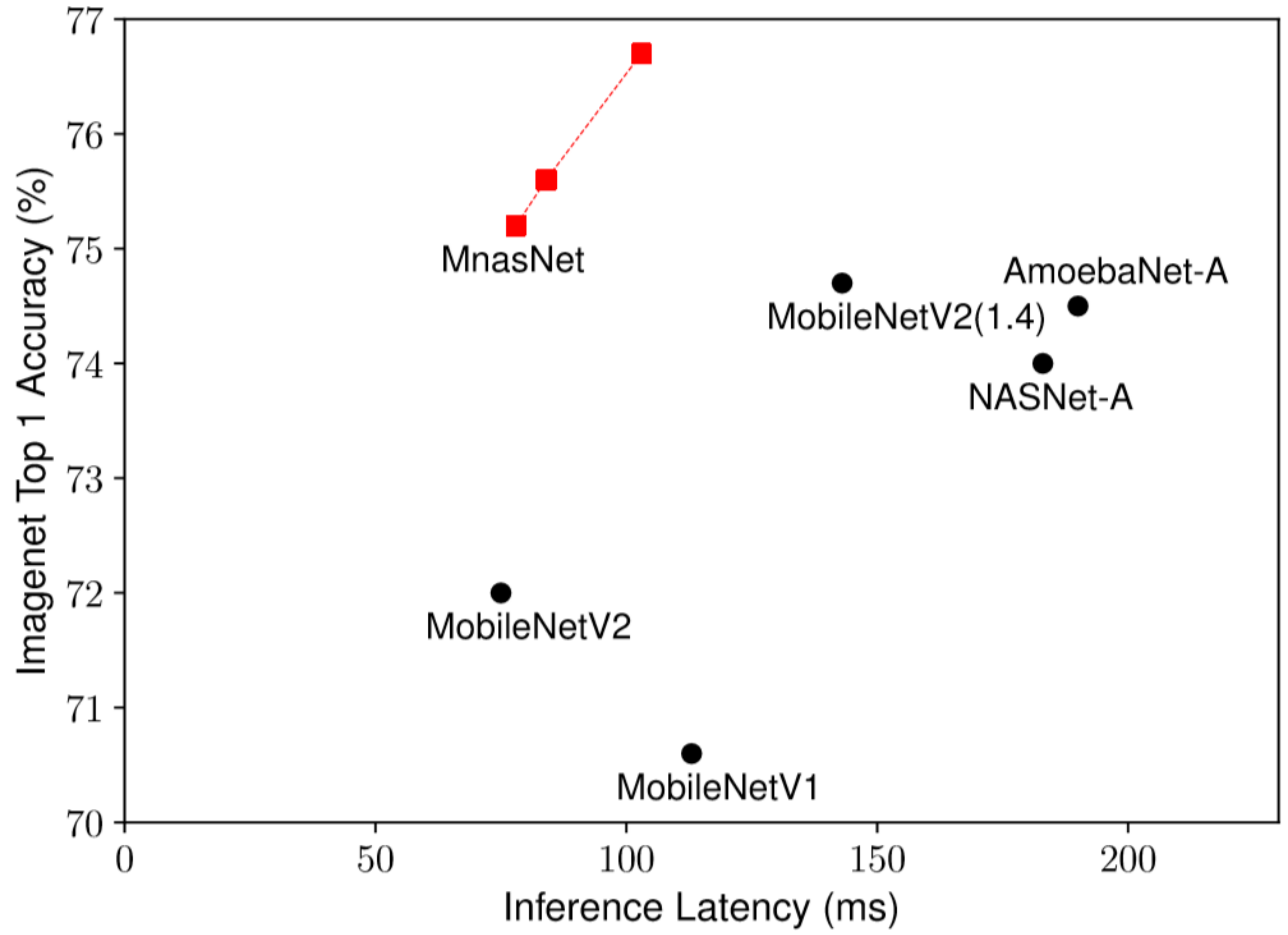
❖ Search on the target task

- MnasNet

❖ Keynotes

- Search directly on ImageNet
- Platform aware search
- Very costly (thousands of TPU-days)



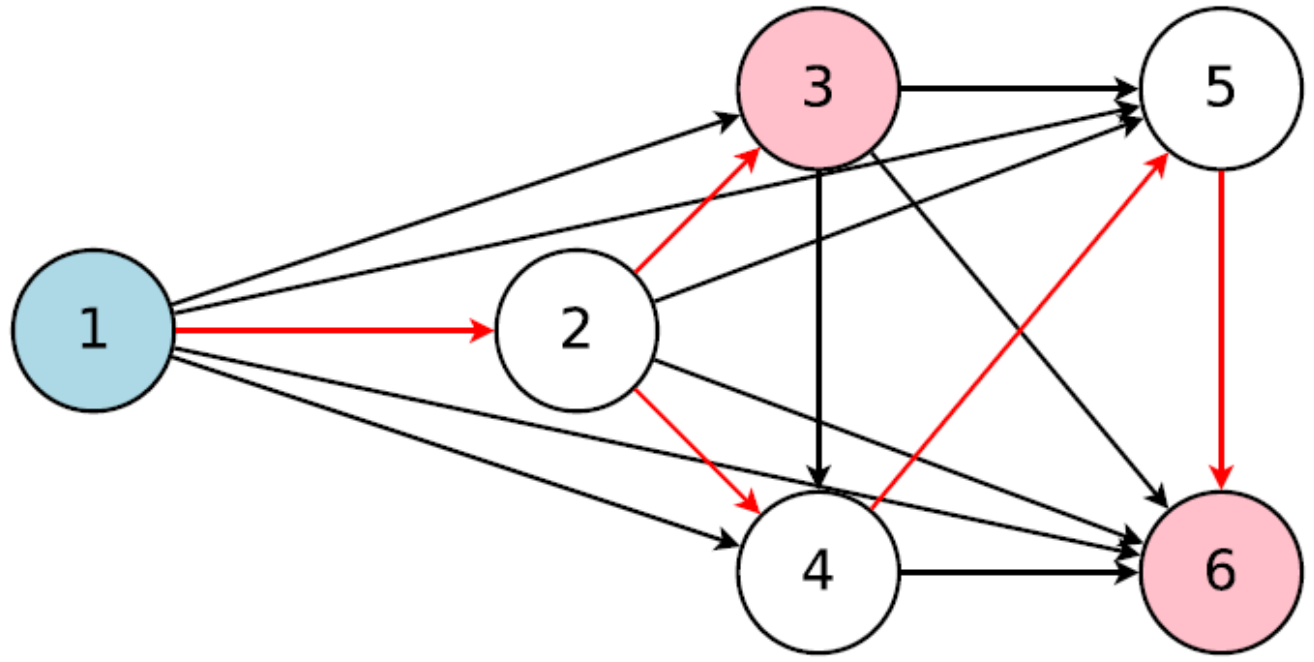



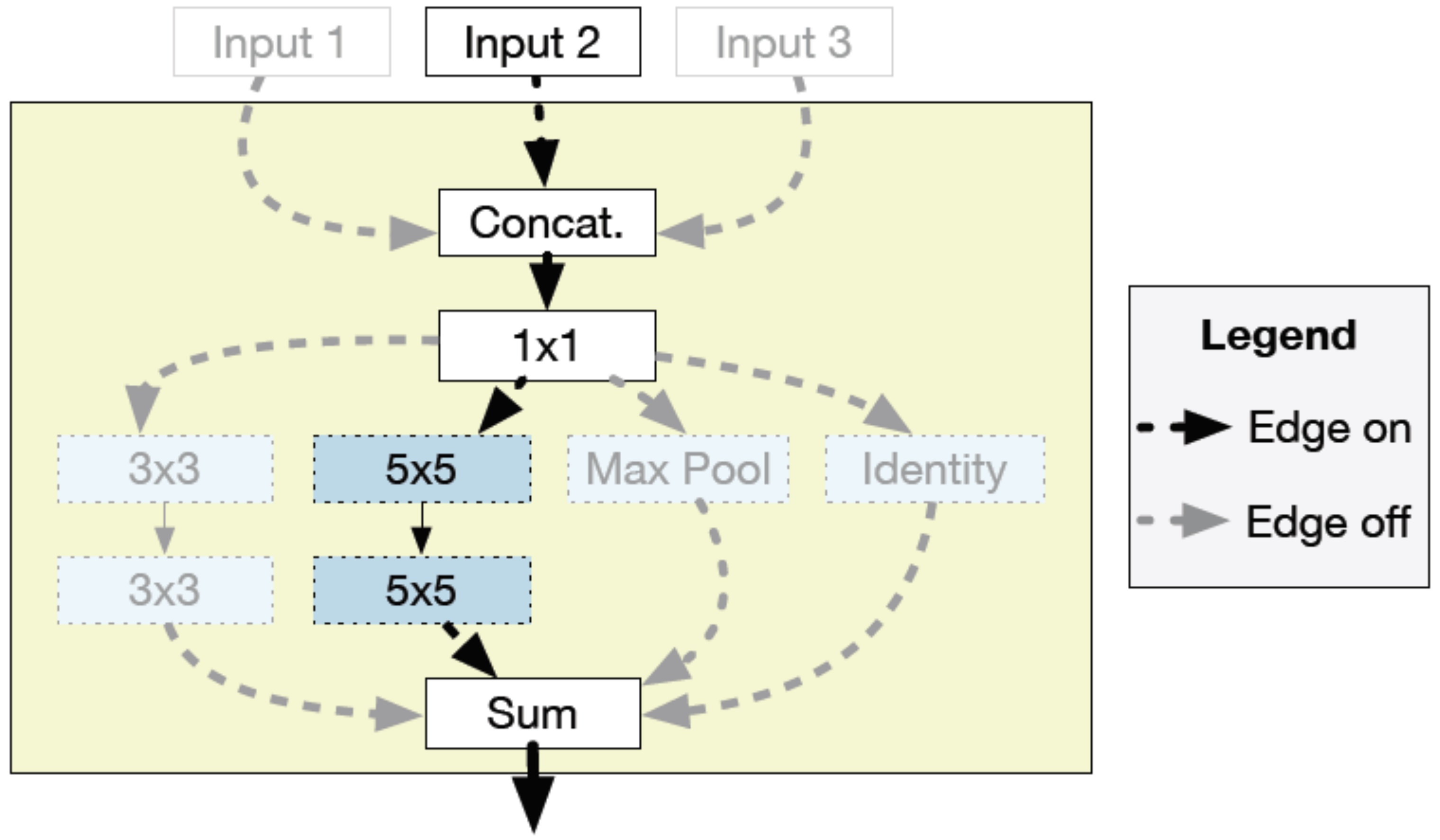

Tan et al. MnasNet: Platform-Aware Neural Architecture Search for Mobile

# Recent Advances in AutoML (3)

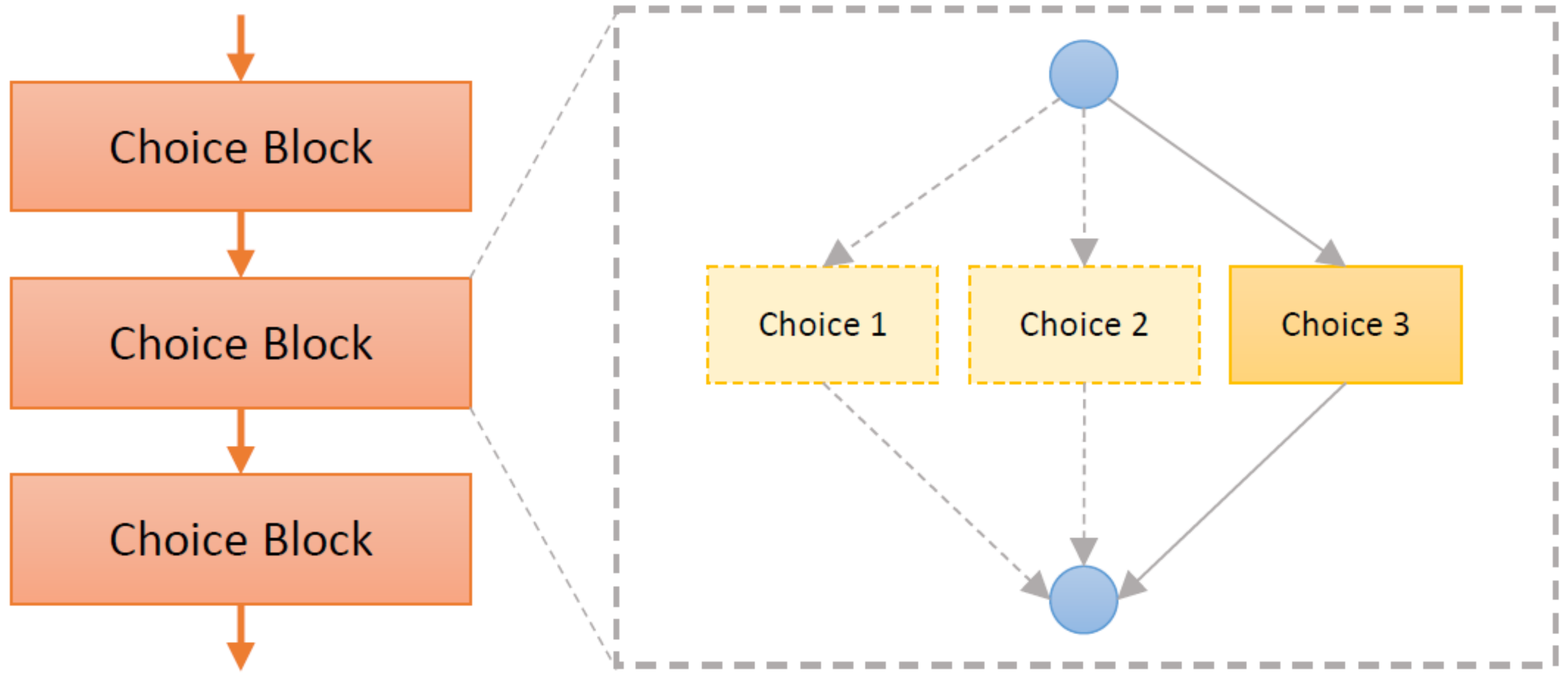
MEGVII 旷视

- Weight Sharing for Efficient Search & Evaluation
  - ENAS
  - One-shot methods




- Keynotes
  - Super network
  - Finetuning & inference only instead of retraining
  - Inconsistency in super net evaluation



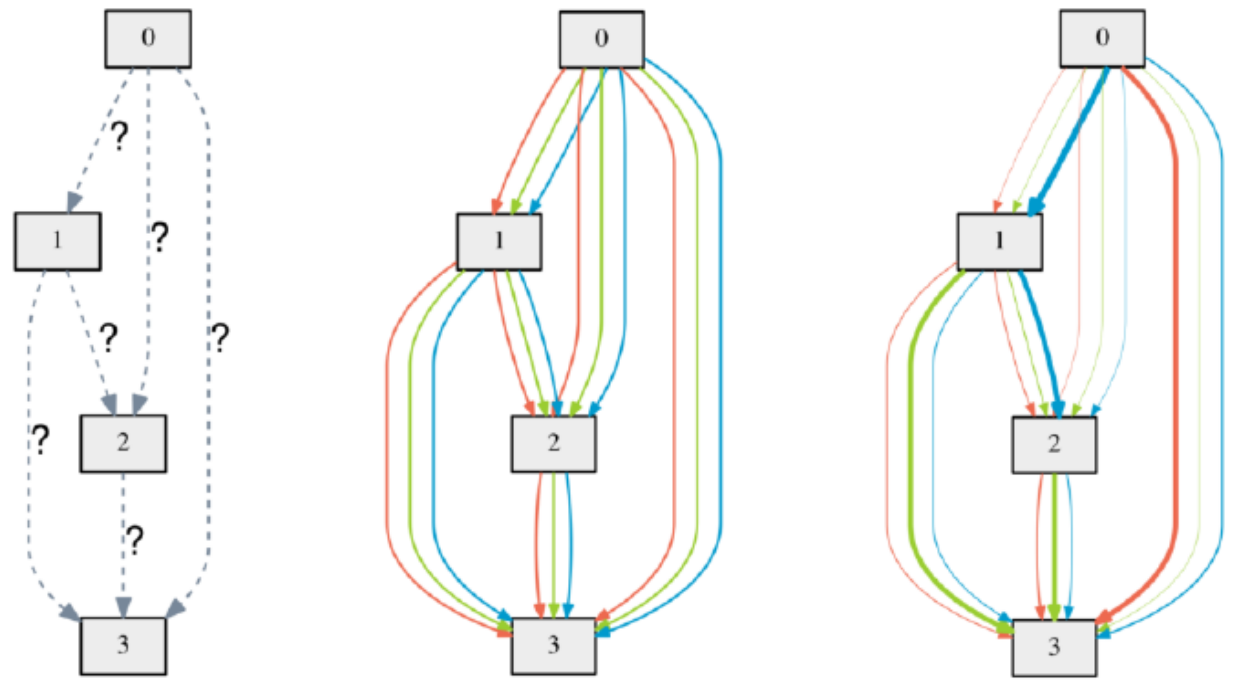


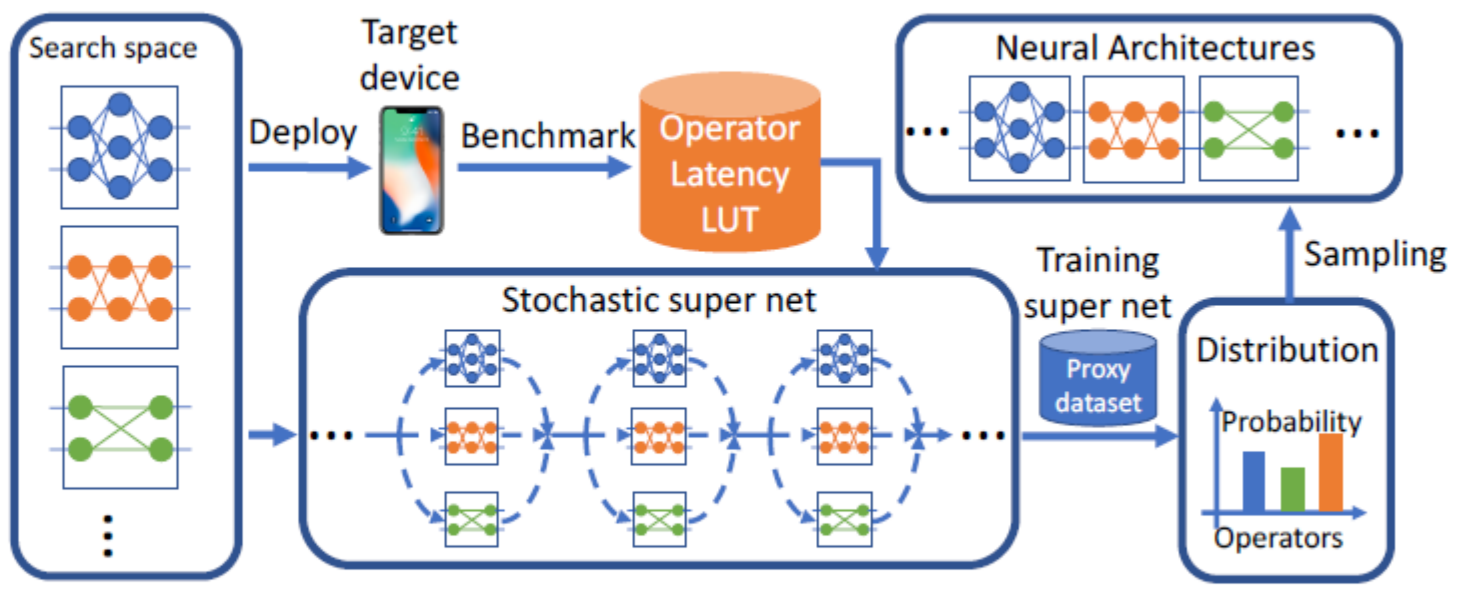
Pham et al. Efficient Neural Architecture Search via Parameter Sharing
Bender et al. Understanding and Simplifying One-Shot Architecture Search
Guo et al. Single Path One-Shot Neural Architecture Search with Uniform Sampling

# Recent Advances in AutoML (4)

❖ Gradient-based methods

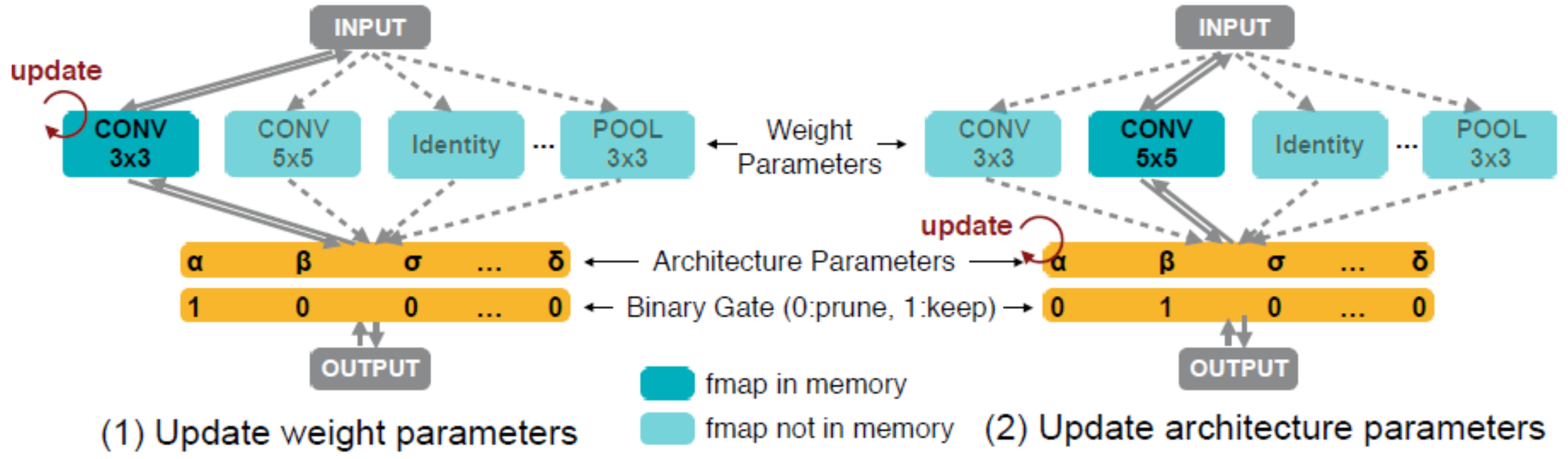
- DARTS
- SNAS, FBNet, ProxylessNAS, …






❖ Keynotes

- Joint optimization of architectures and weights
- Weight sharing implied
- Sometimes less flexible

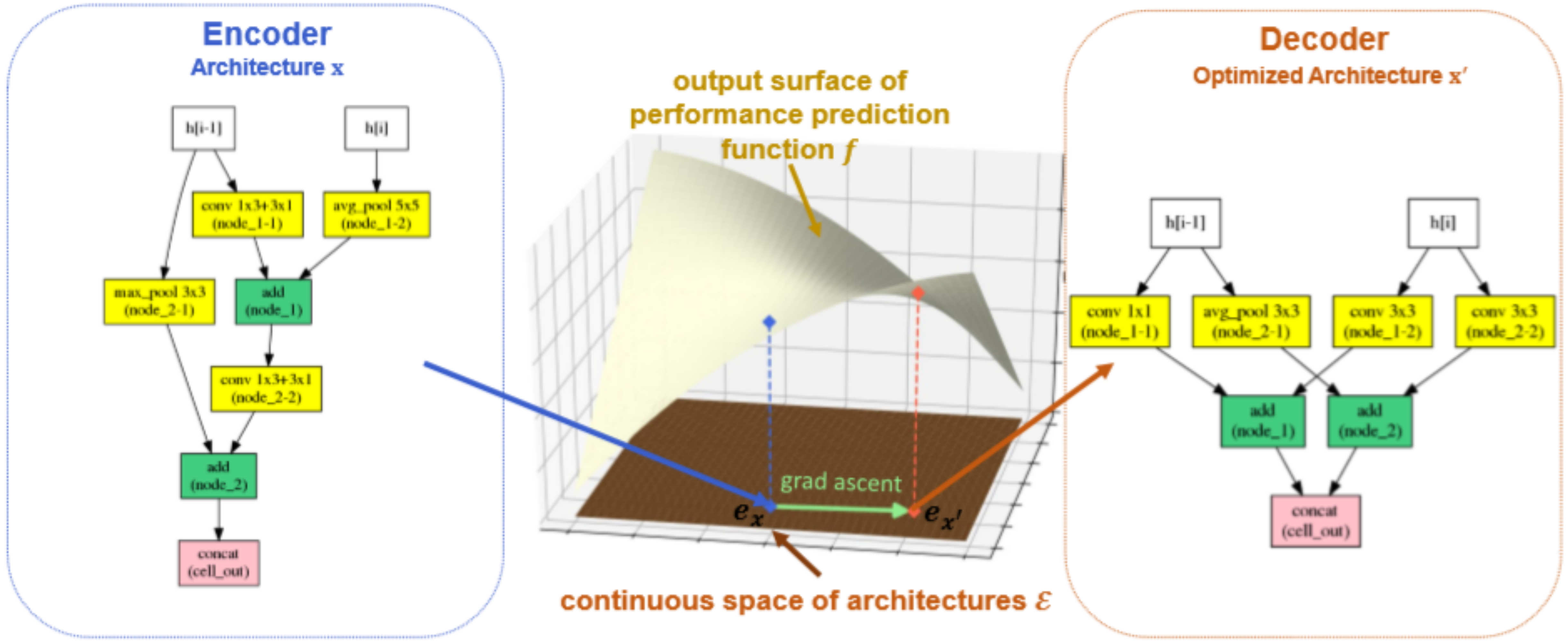



Liu et al. DARTS: Differentiable Architecture Search
Xie et al. SNAS: Stochastic Neural Architecture Search
Cai et al. ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware
Wu et al. FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search

# Recent Advances in AutoML (5)

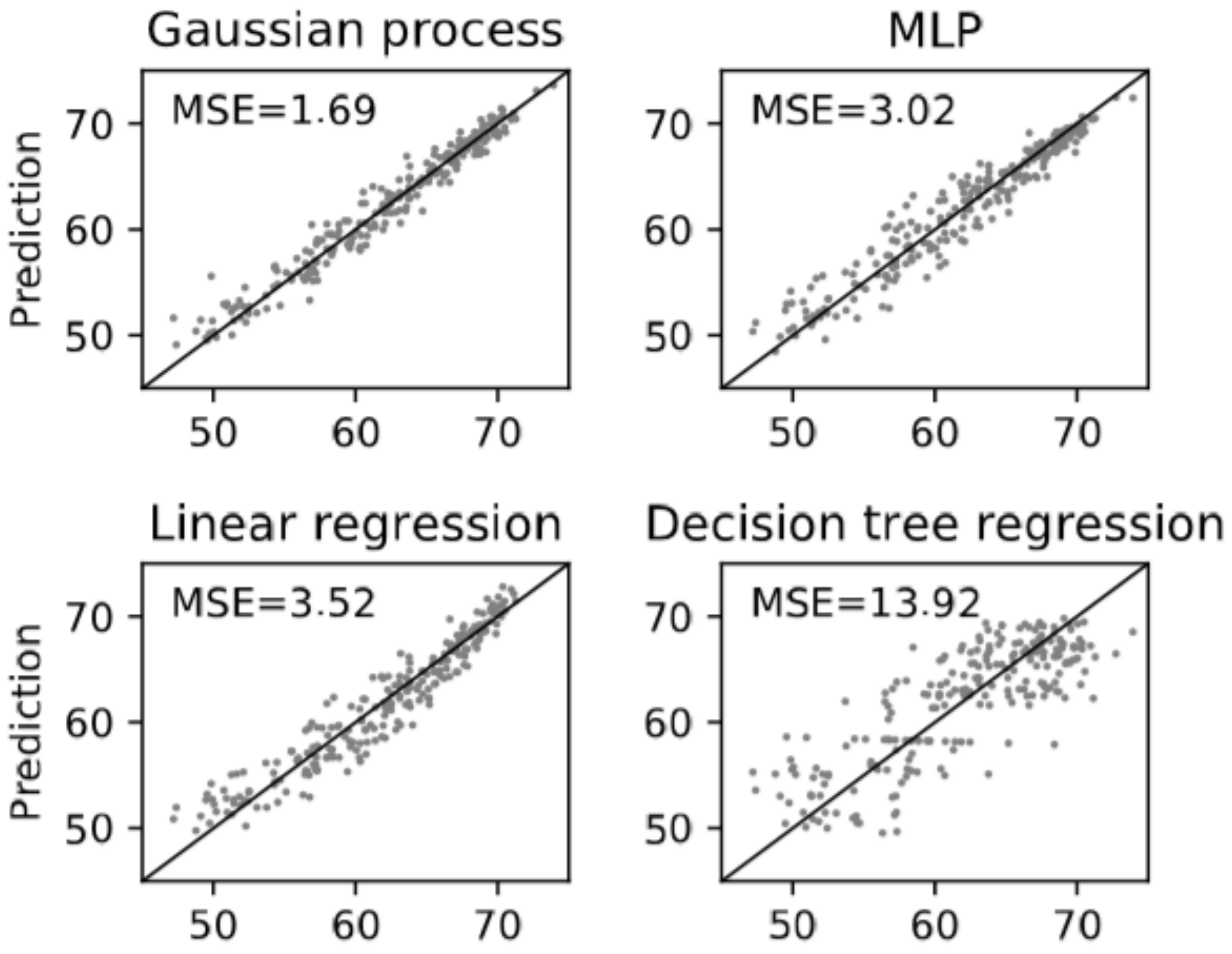
## ❖ Performance Predictor

- Neural Architecture Optimization
- ChamNet




## ❖ Keynotes

- Architecture encoding
- Performance prediction models
- Cold start problem




Luo et al. Neural Architecture Optimization

Dai et al. ChamNet: Towards Efficient Network Design through Platform-Aware Model Adaptation

# Recent Advances in AutoML (6)

MEGVII 旷视

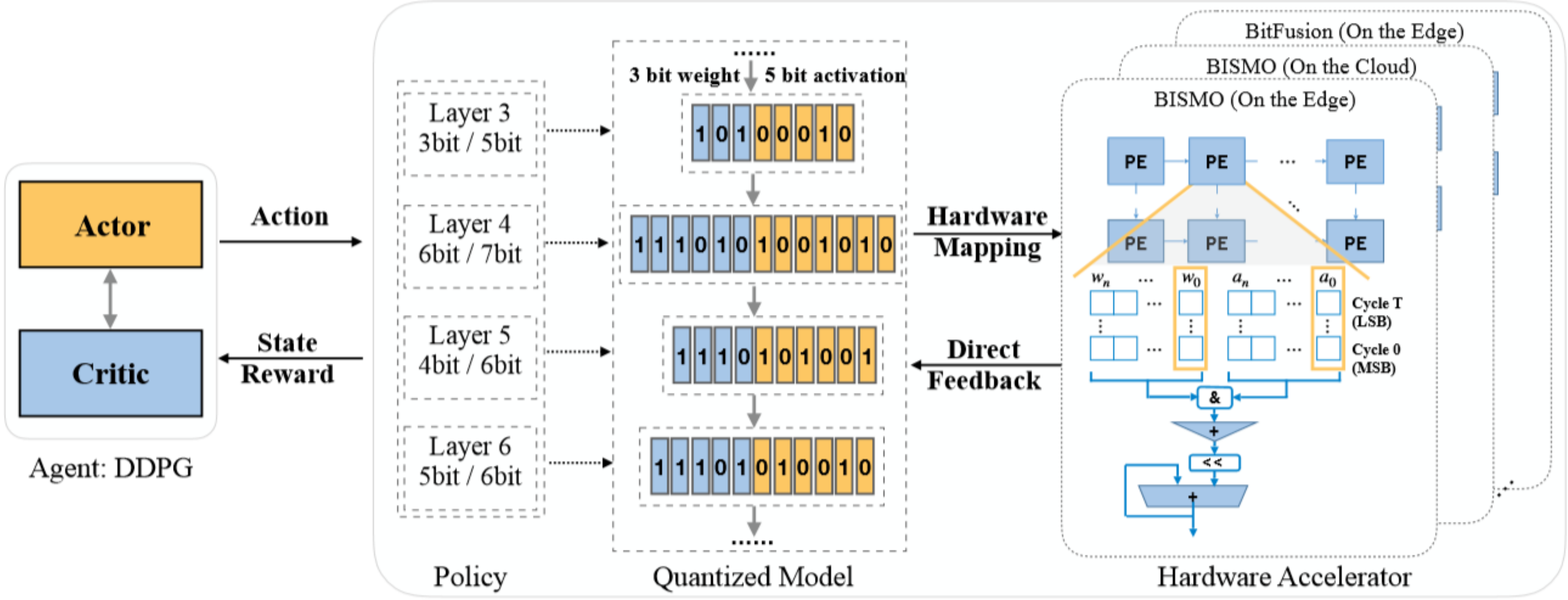
❖ Hardware-aware Search

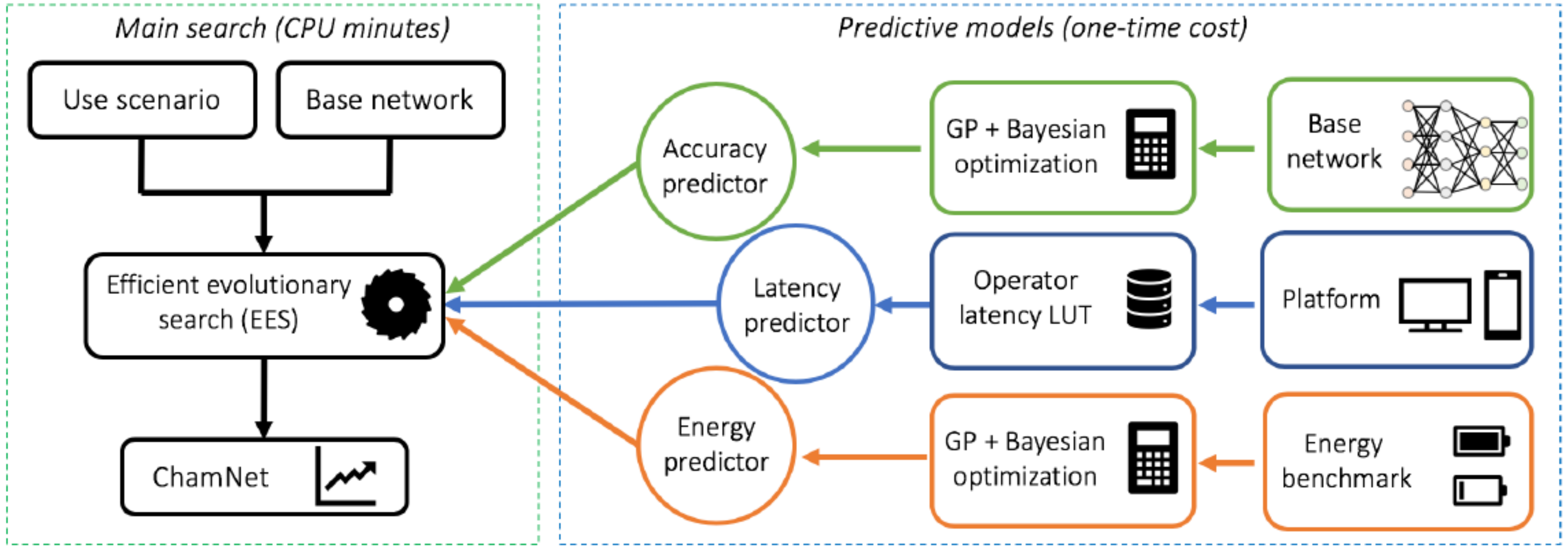
- Search with complexity budget
- Quantization friendly
- Energy-aware search

…

❖ Keynotes

- Complexity-aware loss & reward
- Multi-target search
- Device in the loop





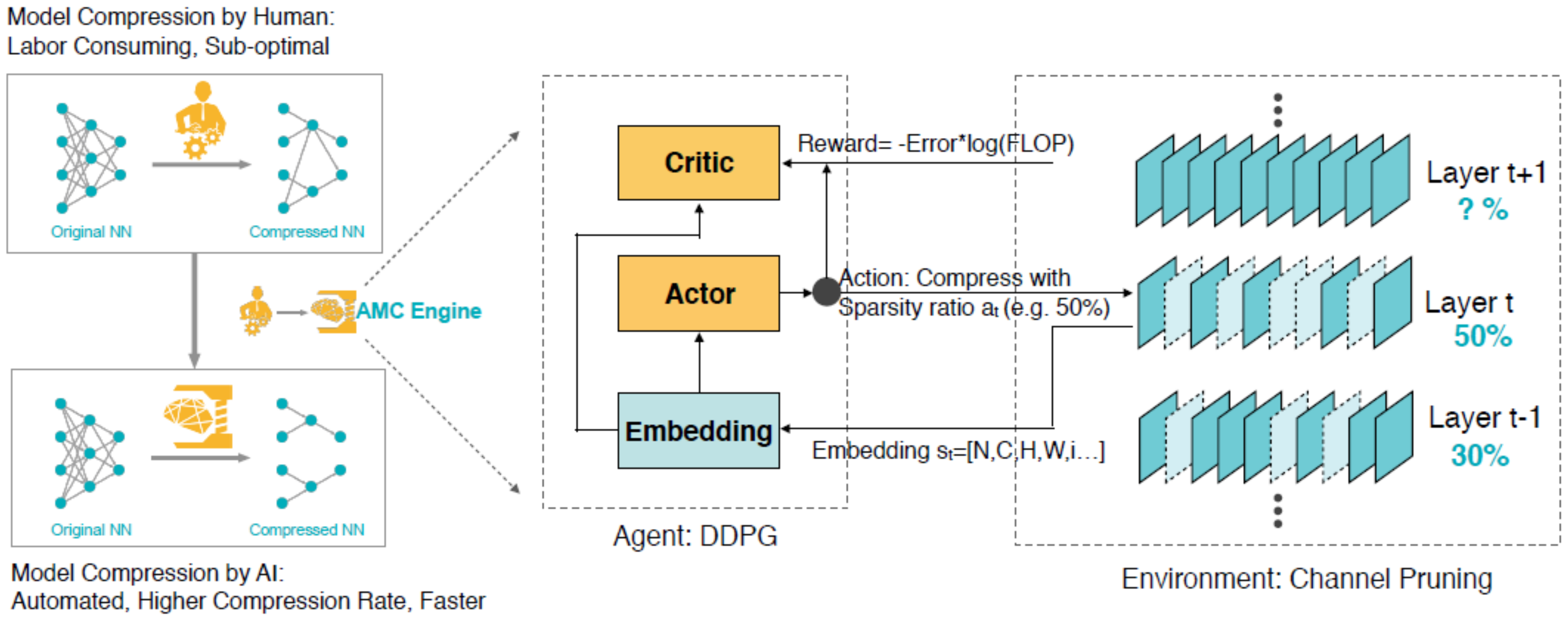
Wu et al. Mixed Precision Quantization of ConvNets via Differentiable Neural Architecture Search
V´eniat et al. Learning Time/Memory-Efficient Deep Architectures with Budgeted Super Networks
Wang et al. HAQ: Hardware-Aware Automated Quantization with Mixed Precision

# Recent Advances in AutoML (7)

MEGVII 旷视

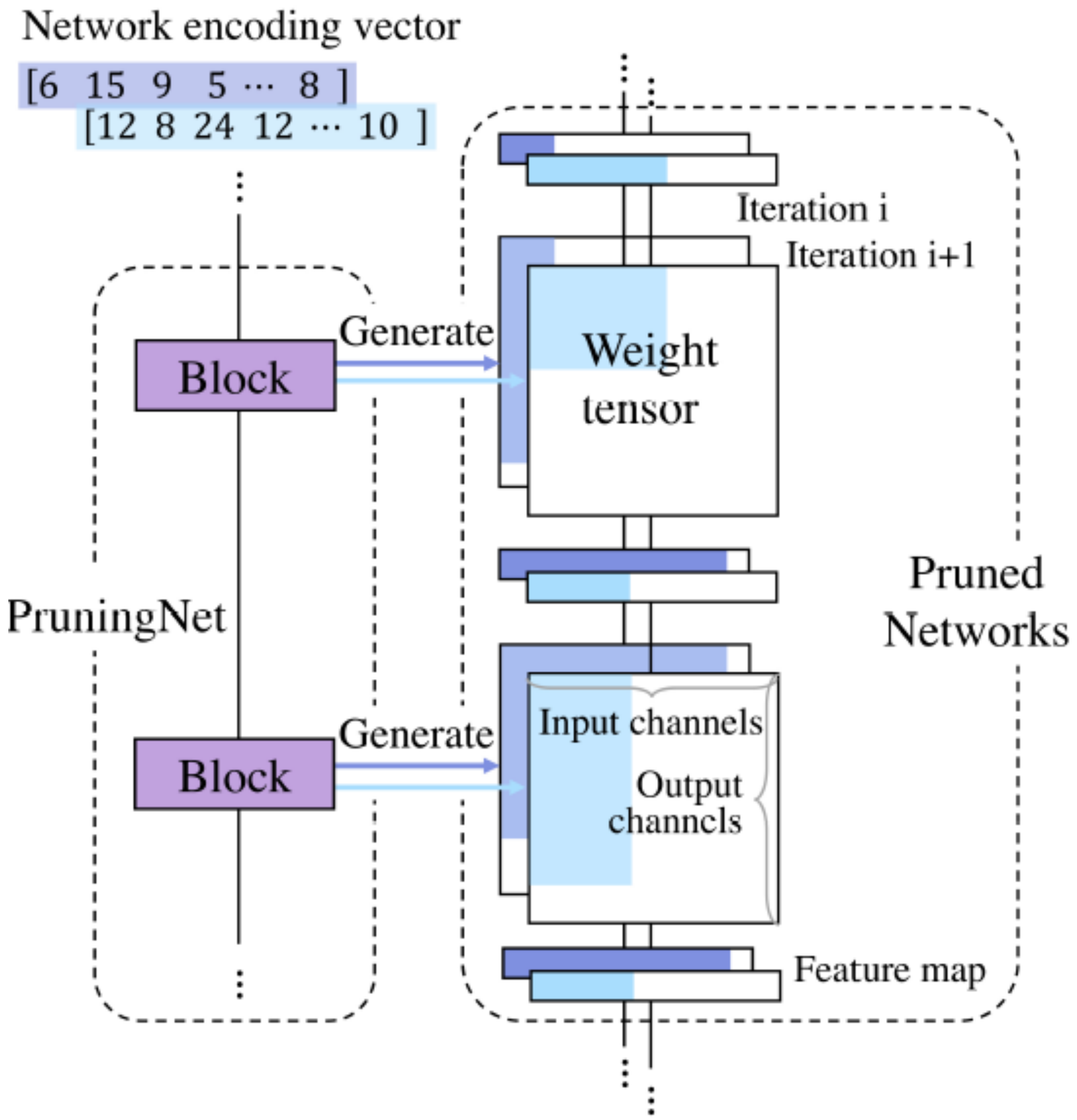
❖ AutoML in Model Pruning

- NetAdapt
- AMC
- MetaPruning




❖ Keynotes

- Search for the pruned architecture
- Hyper-parameters like channels, spatial size, …

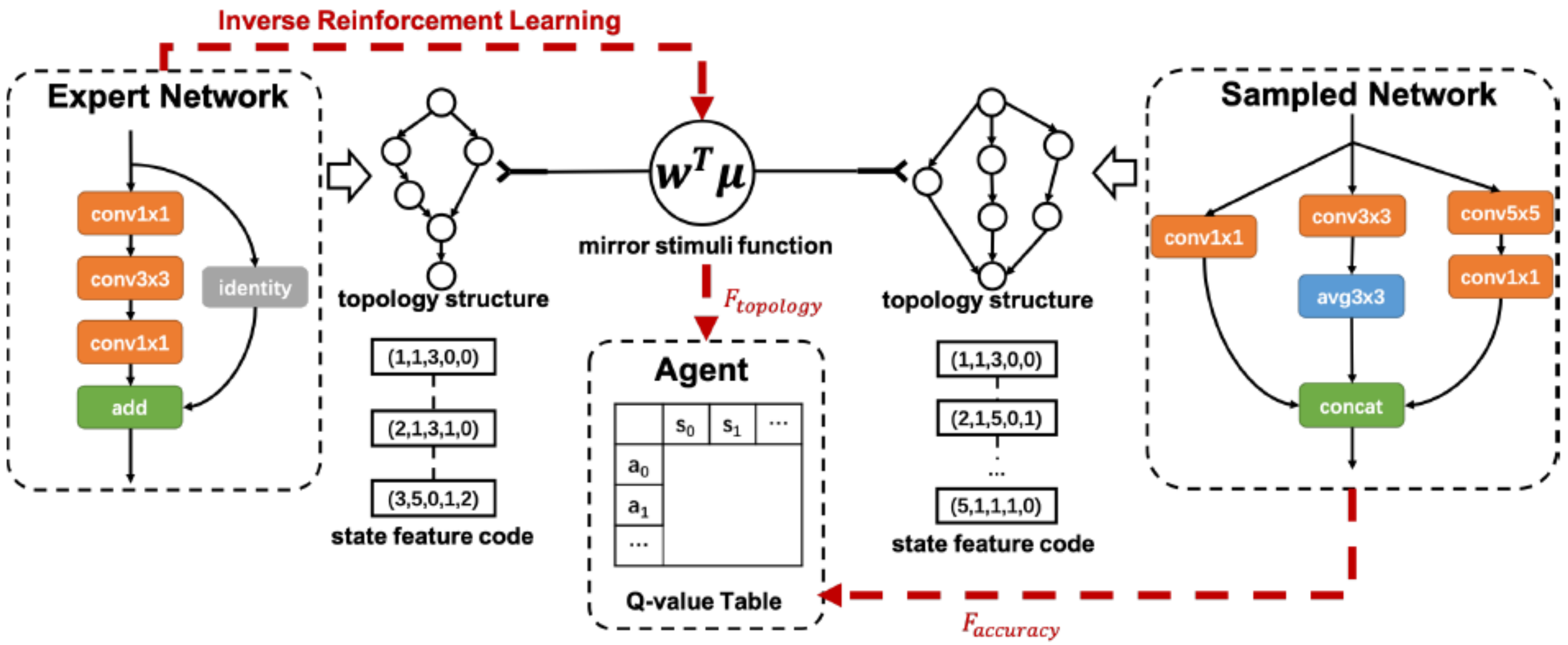




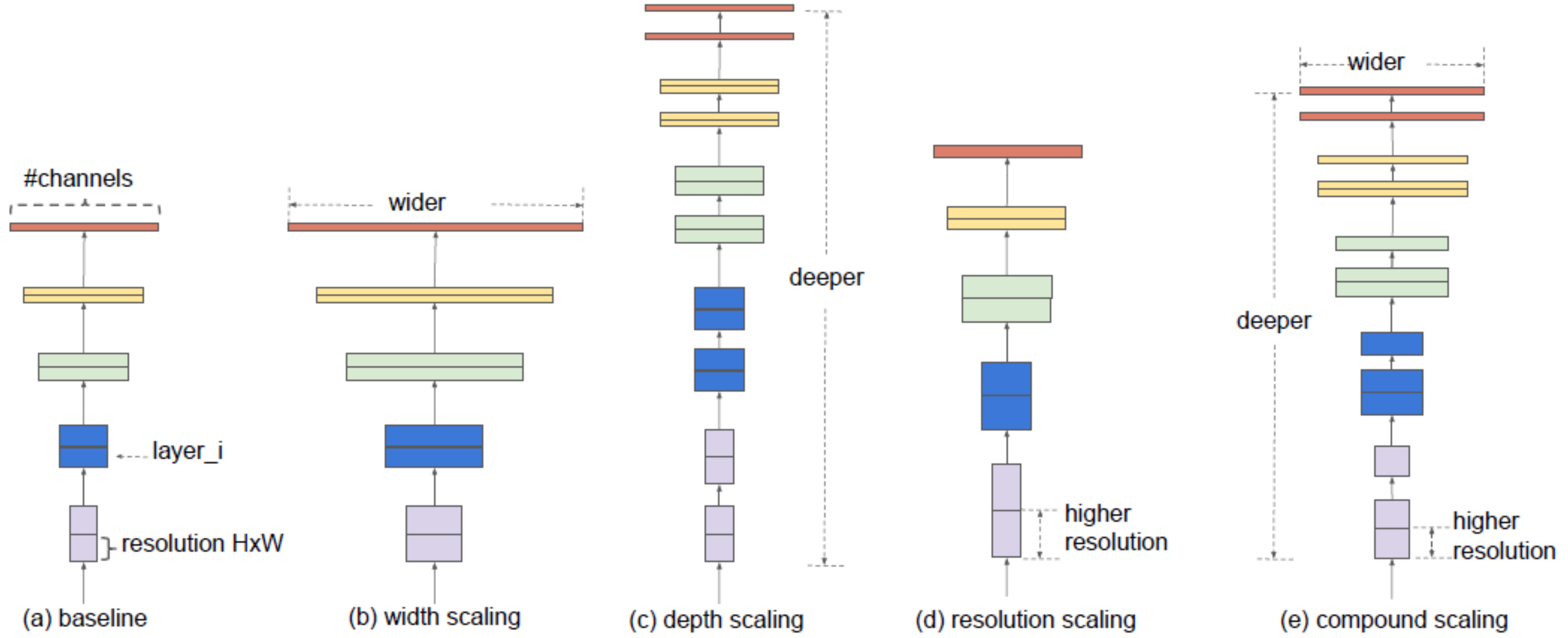
Yang et al. NetAdapt: Platform-Aware Neural Network Adaptation for Mobile Applications
He et al. AMC: AutoML for Model Compression and Acceleration on Mobile Devices
Liu et al. MetaPruning: Meta Learning for Automatic Neural Network Channel Pruning

# Recent Advances in AutoML (8)

- Handcraft + NAS
  - Human-expert guided search (IRLAS)
  - Boosting existing handcraft models (EfficientNet, MobileNet v3)




- Keynotes
  - Very competitive performance
  - Efficient
  - Search space may be restricted





Howard et al. Searching for MobileNetV3
Tan et al. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks
Guo et al. IRLAS: Inverse Reinforcement Learning for Architecture Search

# Recent Advances in AutoML (9)

MEGVII 旷视

- Various Tasks
  - Object Detection
  - Semantic Segmentation
  - Super-resolution
  - Face Recognition

  …

- Not only NAS, search for everything!
  - Activation function
  - Loss function
  - Data augmentation
  - Backpropagation

  …

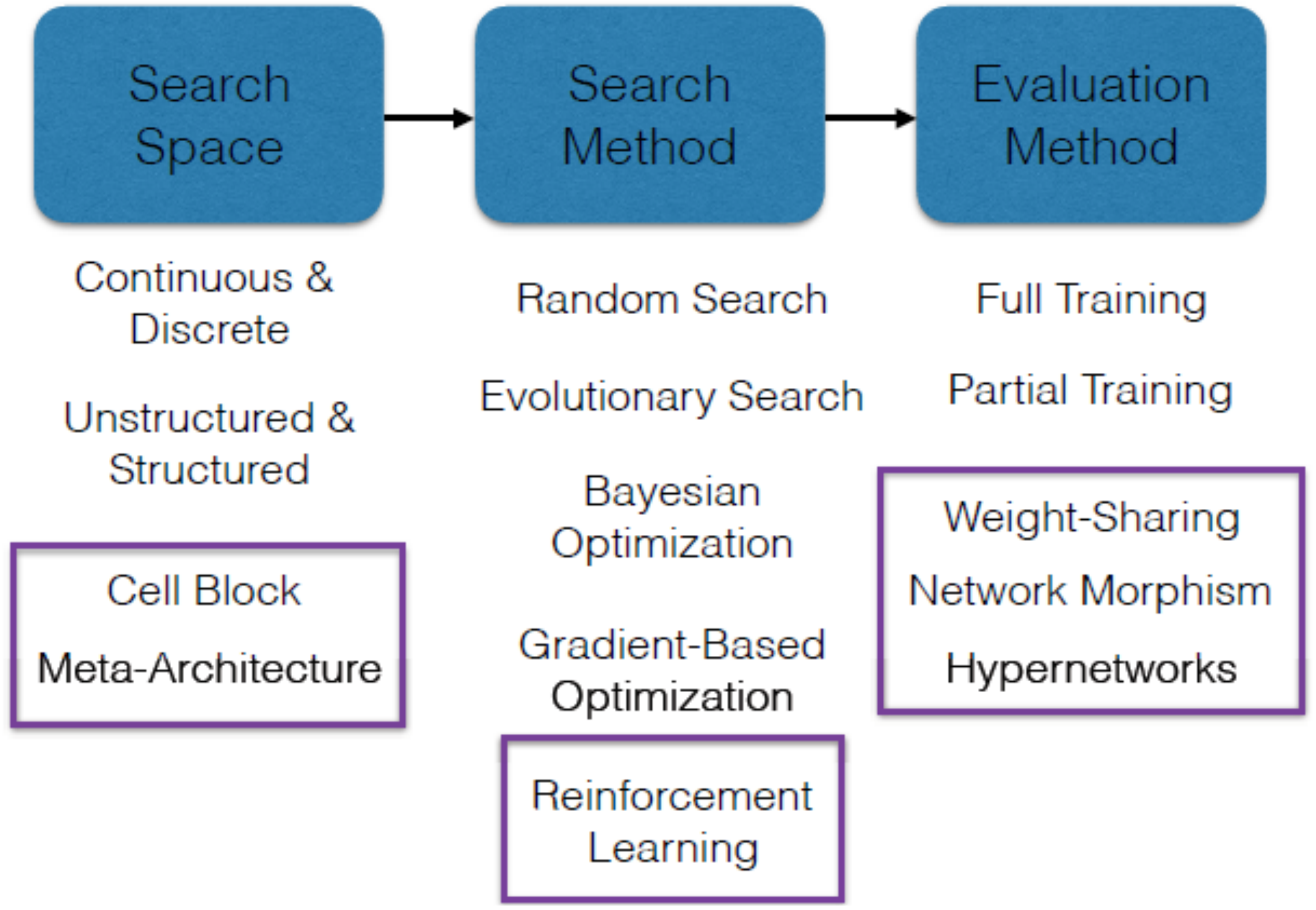

Liu et al. Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation
Chu et al. Fast, Accurate and Lightweight Super-Resolution with Neural Architecture Search
Ramachandra et al. Searching for Activation Functions
Alber et al. Backprop Evolution

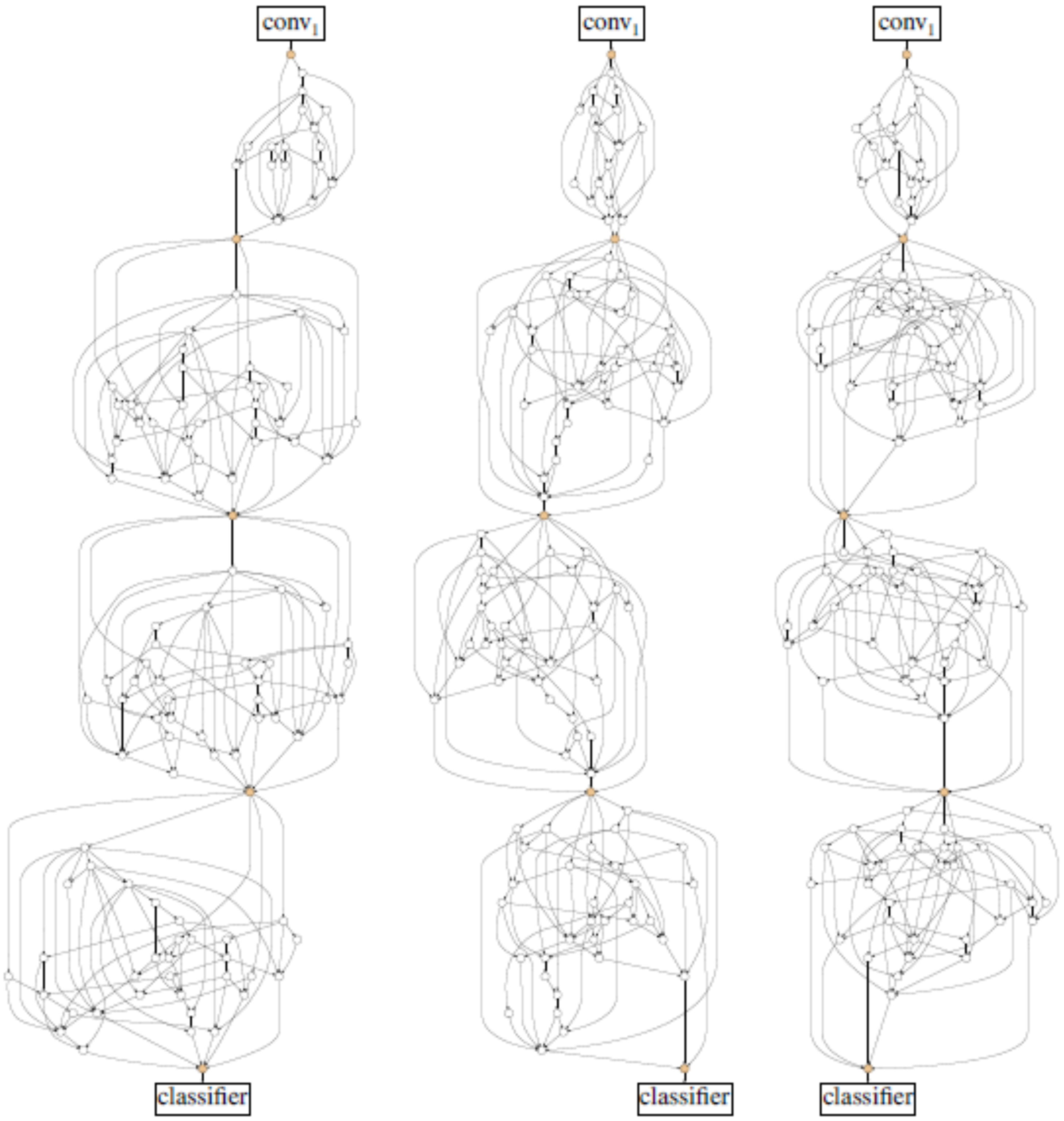
# Recent Advances in AutoML (10)

MEGVII 旷视

- Rethinking the Effectiveness of NAS
  - Random search
  - Random wire network

- Keynotes
  - Reproducibility
  - Search algorithm or search space?
  - Baselines

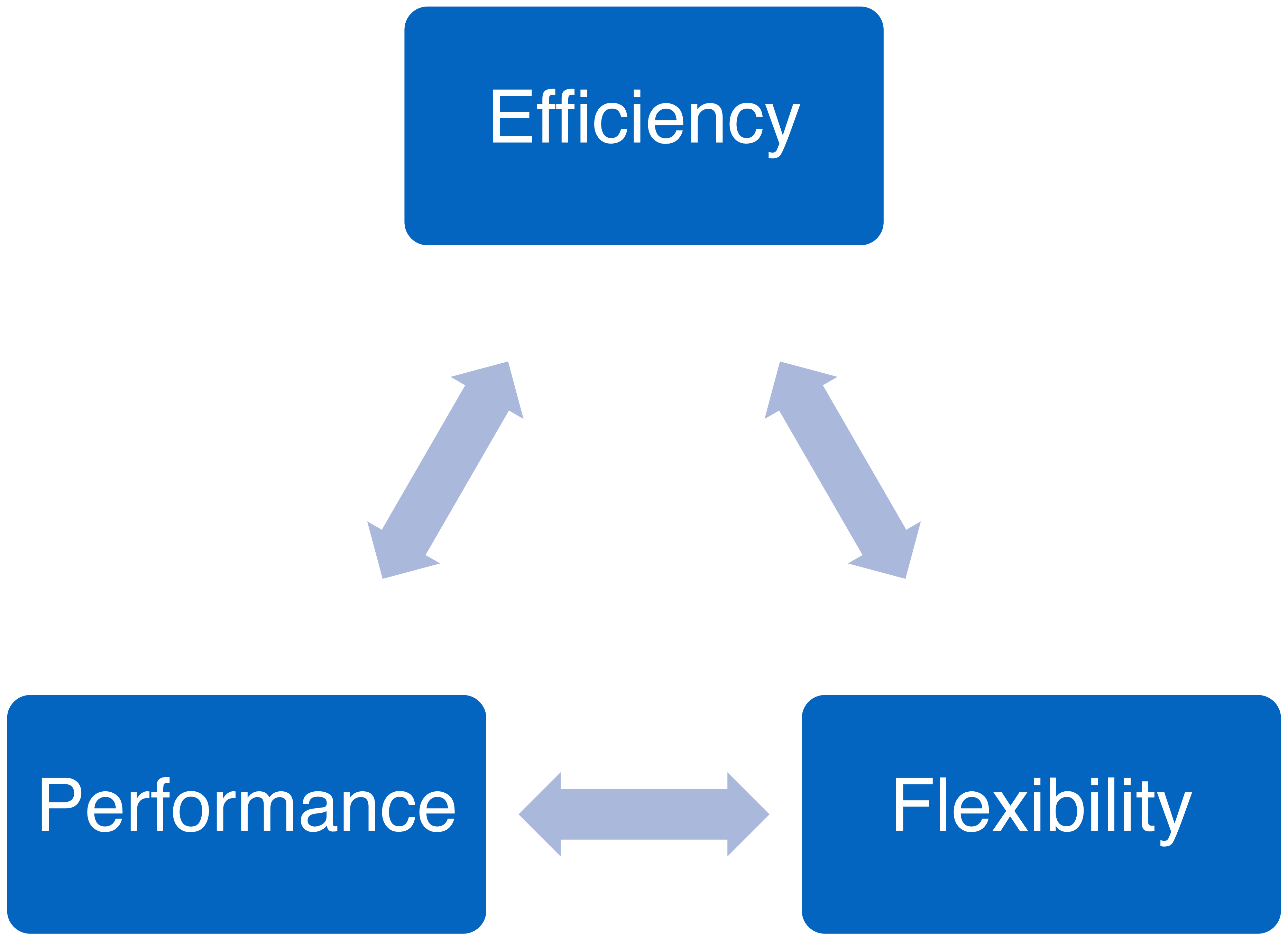



Li et al. Random Search and Reproducibility for Neural Architecture Search
Xie et al. Exploring Randomly Wired Neural Networks for Image Recognition

# Summary: Trends and Challenges

- Trends
  - Efficient & high-performance algorithm
  - Flexible search space
  - Device-aware optimization
  - Multi-task / Multi-target search
- Challenges
  - Trade-offs between efficiency, performance and flexibility
  - Search space matters!
  - Fair benchmarks
  - Pipeline search



# AutoML for Object Detection

1

- Advances in AutoML

2

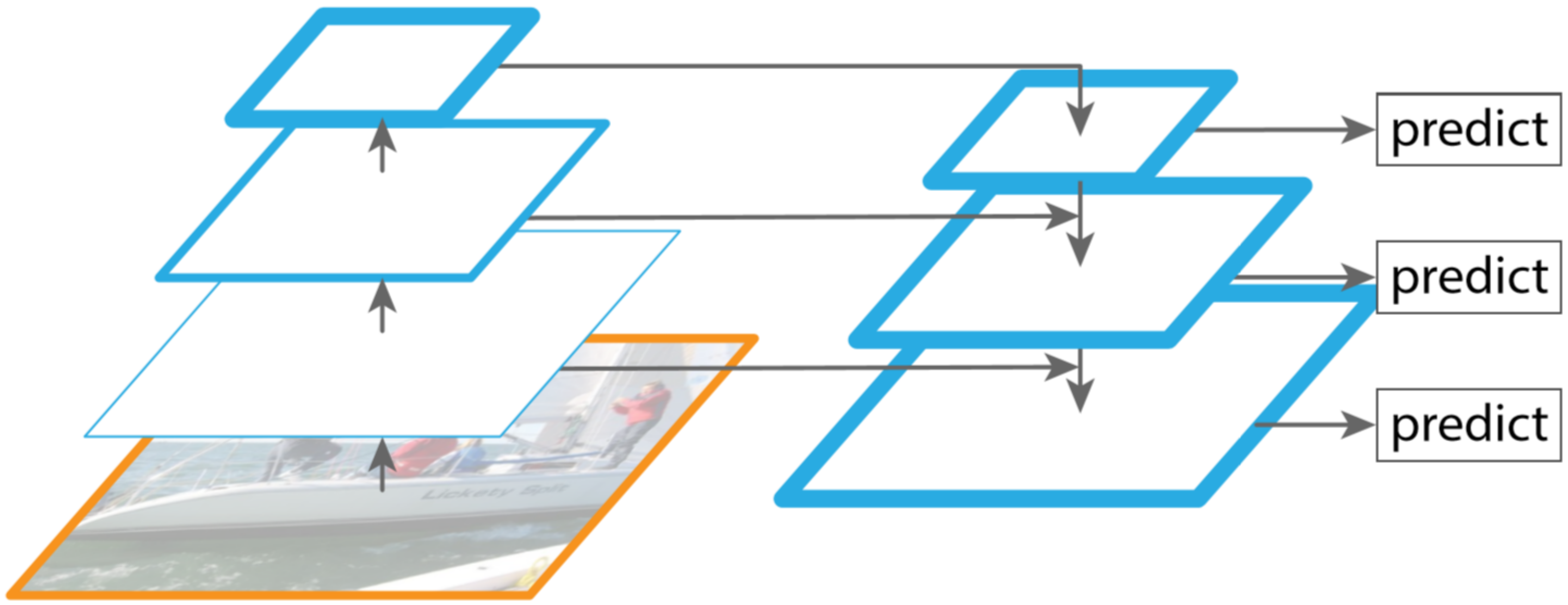
- **Search for Detection Systems**

# AutoML for Object Detection

MEGVII 旷视

- Components to search
  - Image preprocessing
  - Backbone
  - Feature fusion
  - Detection head & loss function

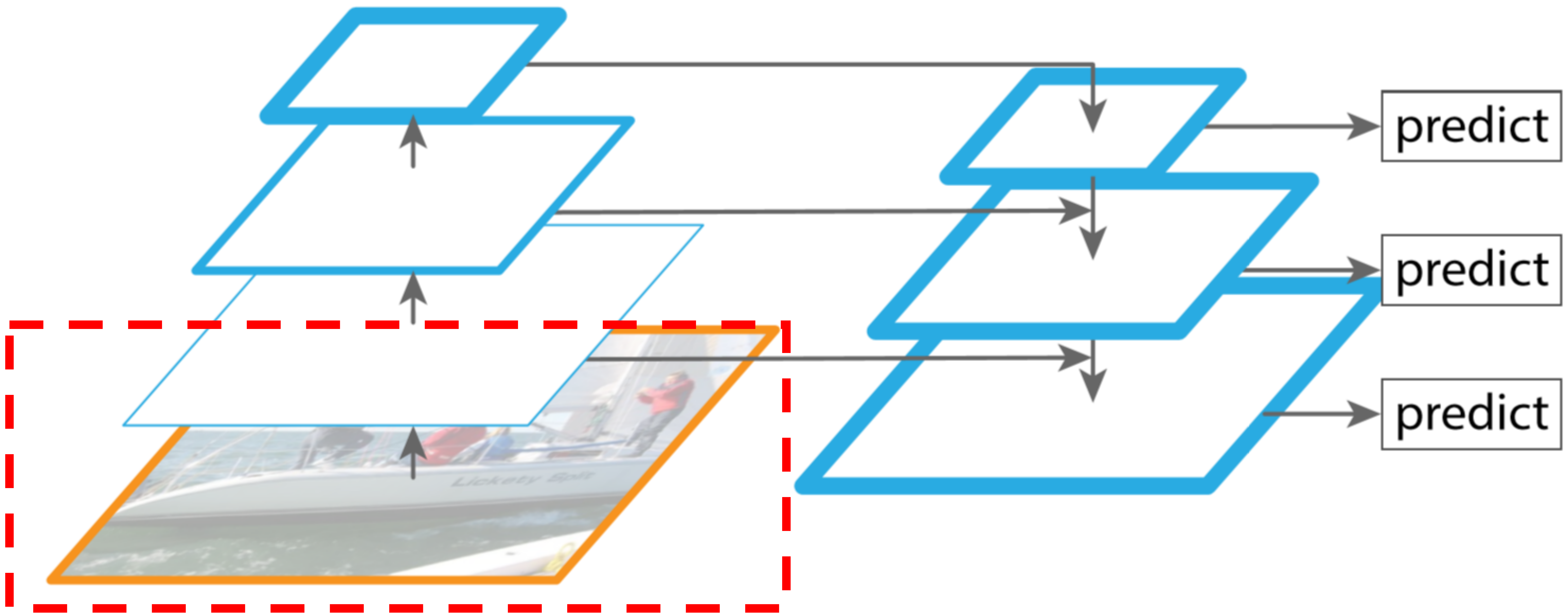
  …



# AutoML for Object Detection

- Components to search
  - **Image preprocessing**
  - Backbone
  - Feature fusion
  - Detection head & loss function

…



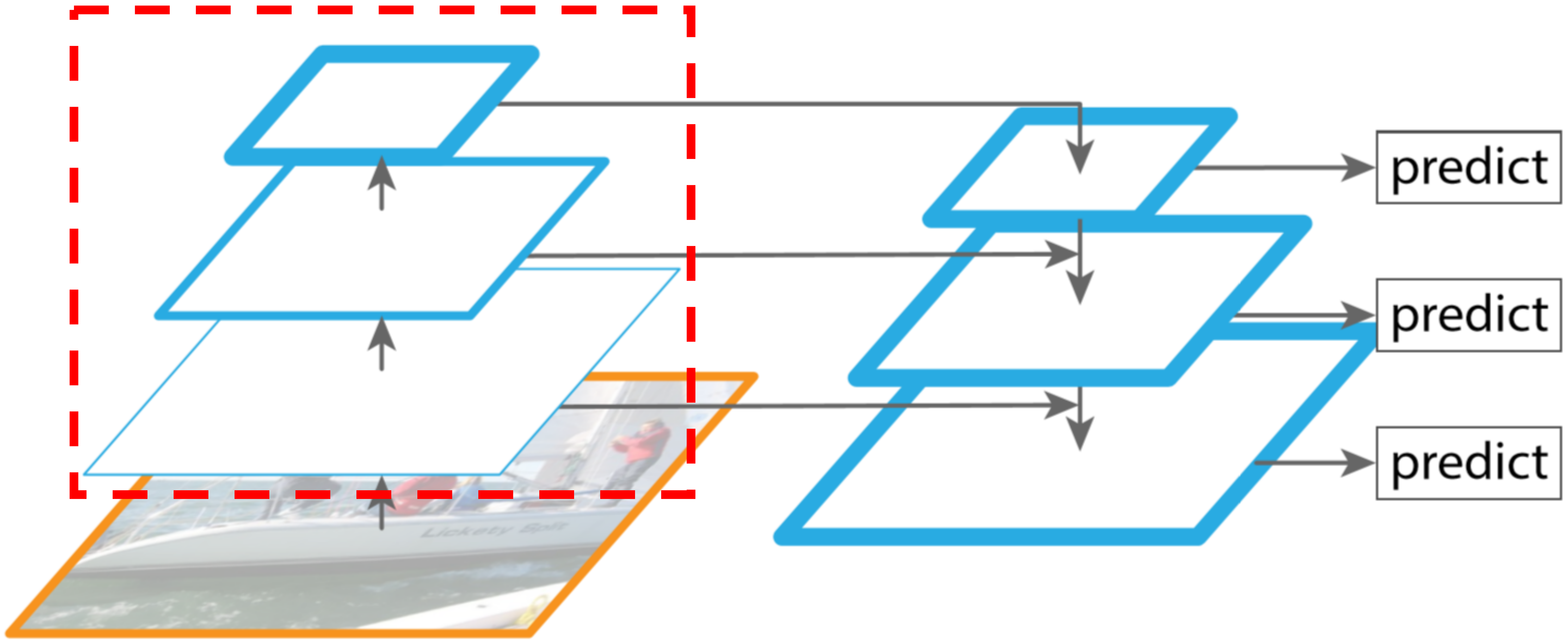
# AutoML for Object Detection

MEGVII 旷视

❖ Components to search

- Image preprocessing
- **Backbone**
- Feature fusion
- Detection head & loss function

…



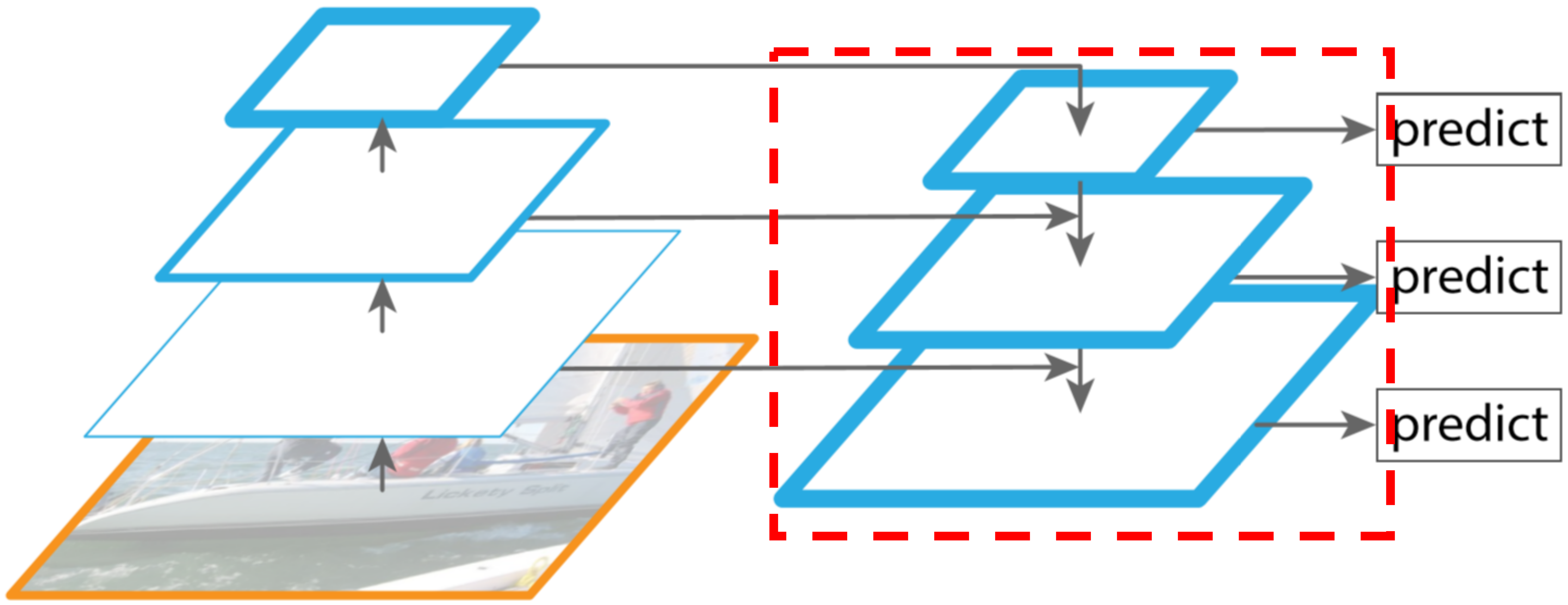
# AutoML for Object Detection

MEGVII 旷视

❖ Components to search

- Image preprocessing
- Backbone
- **Feature fusion**
- Detection head & loss function

…

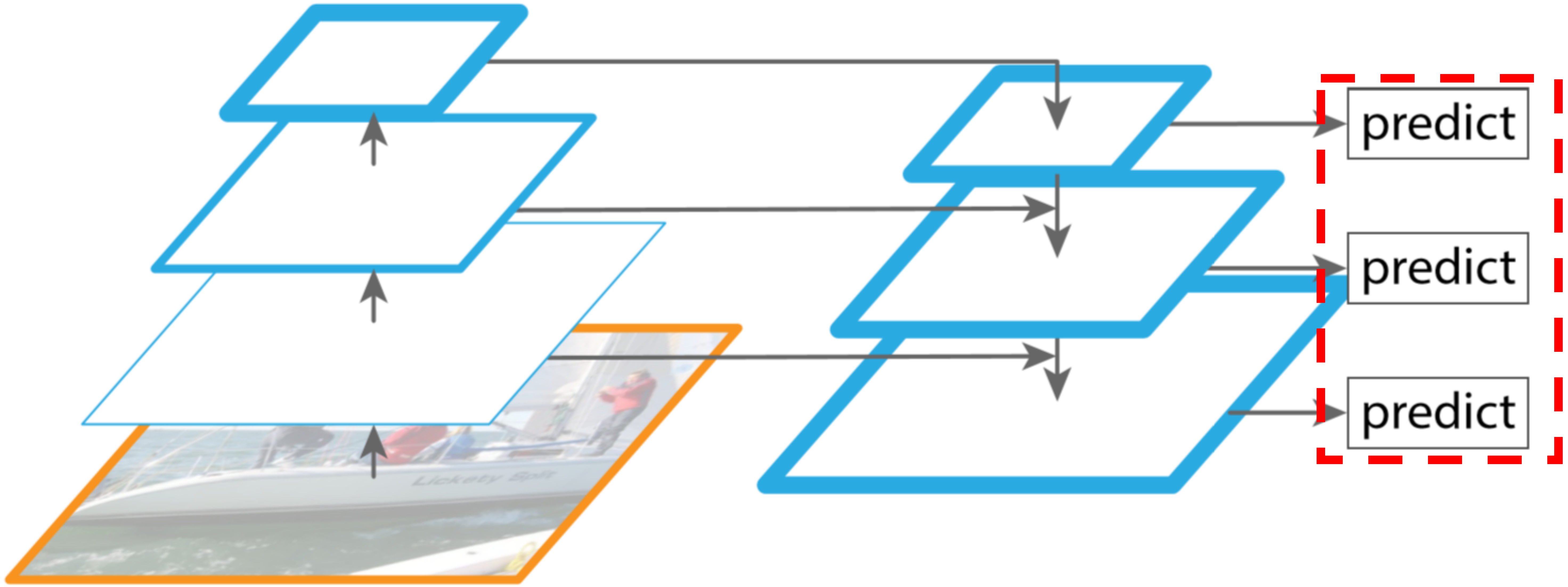


# AutoML for Object Detection

❖ Components to search

- Image preprocessing
- Backbone
- Feature fusion
- **Detection head & loss function**

…



# Search for Detection Systems

MEGVII 旷视

**Backbone**

DetNAS

**Feature Fusion**

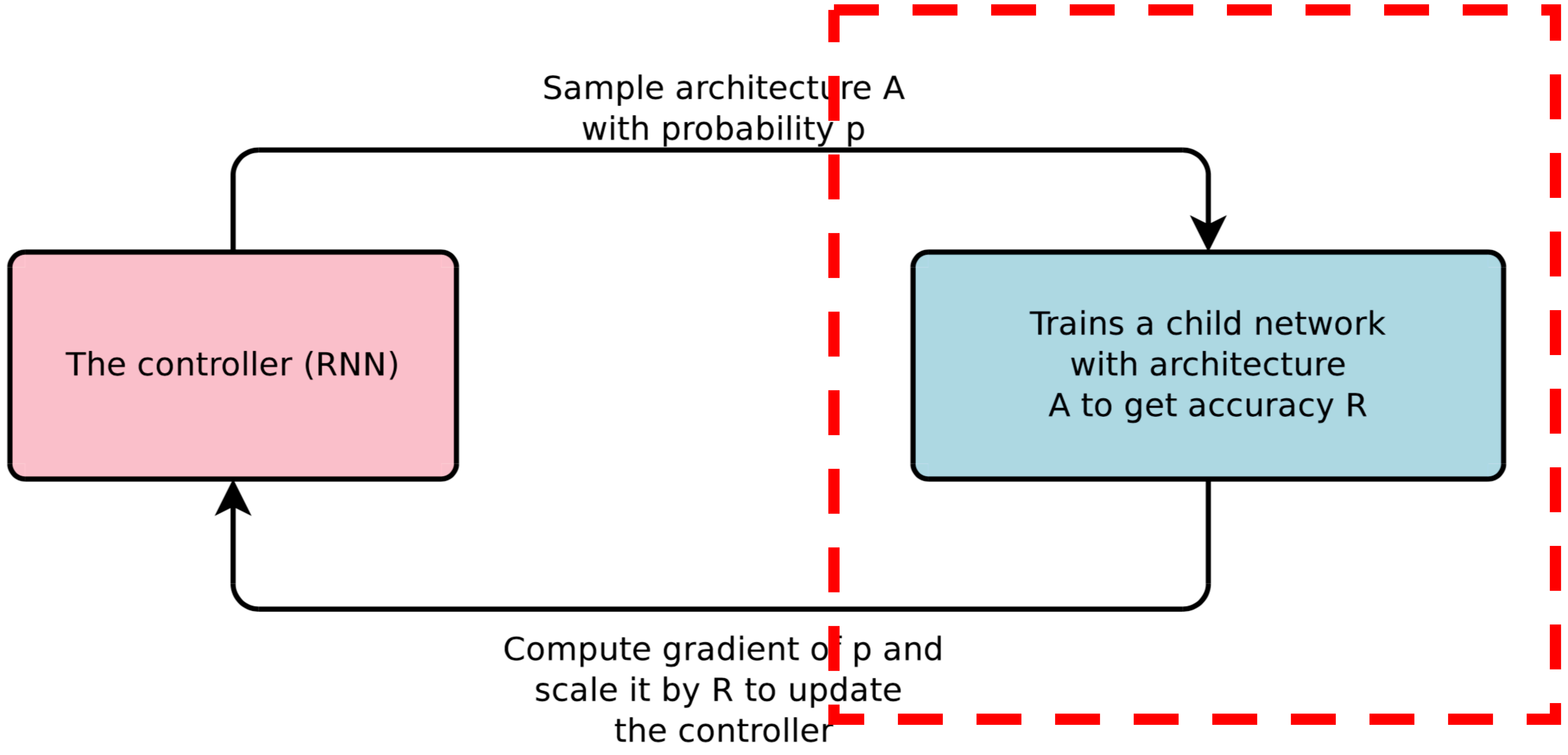
**Augmentation**

Chen et al. DetNAS: Backbone Search for Object Detection

# Challenges of Backbone Search

- Similar to general NAS, but …
  - Controller & evaluator loop
  - Performance evaluation is very slow

- Detection backbone evaluation involves a costly pipeline
  - ImageNet pretraining
  - Finetuning on the detection dataset (e.g. COCO)
  - Evaluation on the validation set



# Related Work: Single Path One-shot NAS

MEGVII 旷视

- Decoupled weight training and architecture optimization

$$w_a = \operatorname{argmin} \mathcal{L}_{\text{train}}\left(\mathcal{N}(a, w)\right),$$
$$a^* = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \text{ACC}_{\text{val}}\left(\mathcal{N}(a, w_a)\right),$$

$$W_{\mathcal{A}} = \underset{W}{\operatorname{argmin}} \mathcal{L}_{\text{train}}\left(\mathcal{N}(\mathcal{A}, W)\right).$$
$$a^* = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \text{ACC}_{\text{val}}\left(\mathcal{N}(a, W_{\mathcal{A}}(a))\right).$$

- Super net training

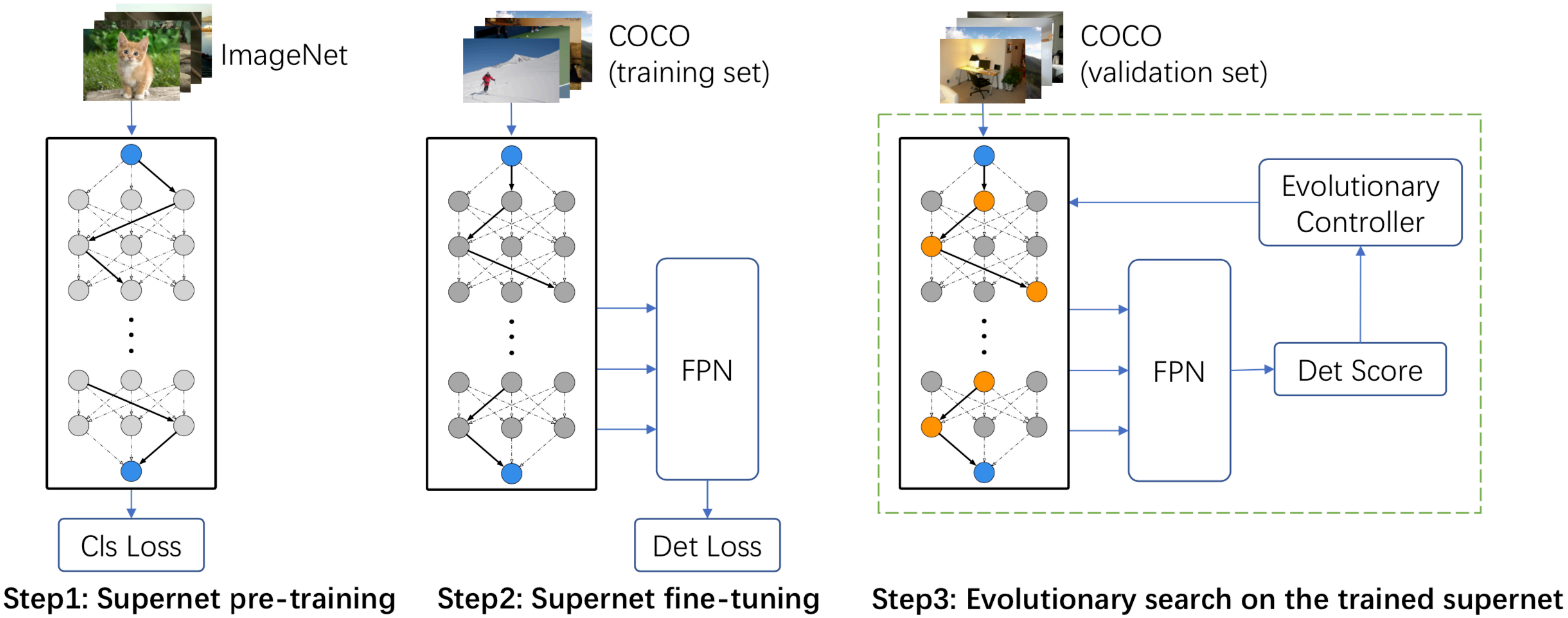
$$W_{\mathcal{A}} = \underset{W}{\operatorname{argmin}} \mathbb{E}_{a \sim \Gamma(\mathcal{A})}\left[\mathcal{L}_{\text{train}}(\mathcal{N}(a, W(a)))\right],$$

Guo et al. Single Path One-Shot Neural Architecture Search with Uniform Sampling

# Pipeline

MEGVII 旷视

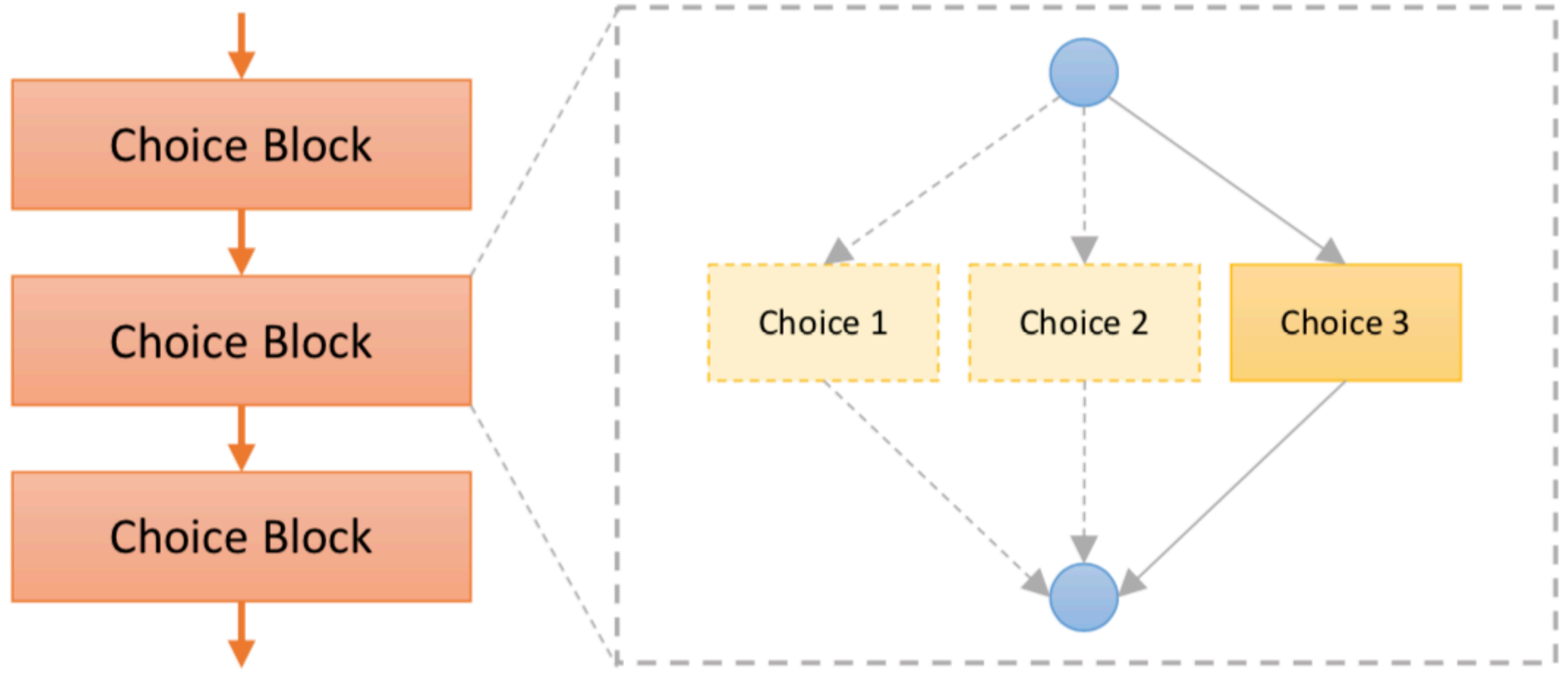
- Single-pass approach
  - Pretrain and finetune super net only once

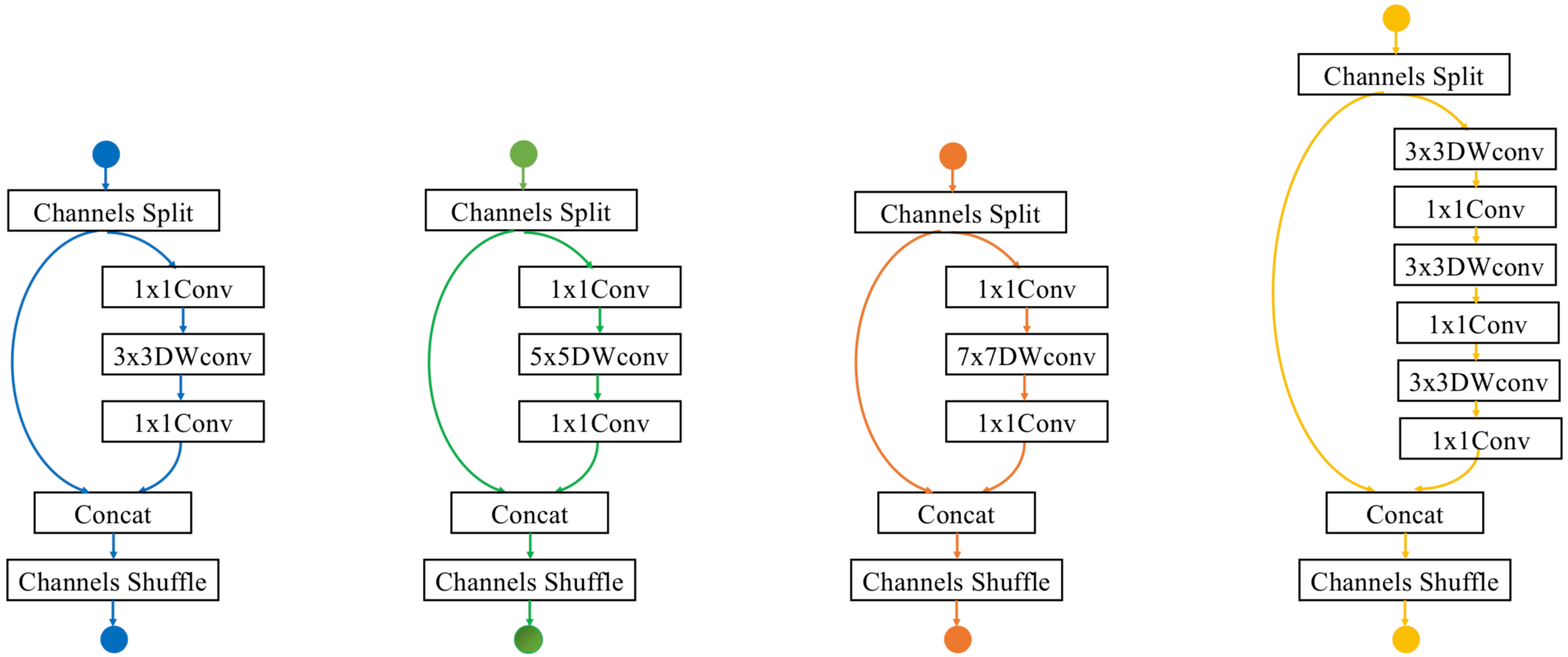


# Search Space



- Single path super net
  - 20 (small) or 40 (large) choice blocks
  - 4 candidates for each choice block
  - Search space size: $4^{20}$ or $4^{40}$



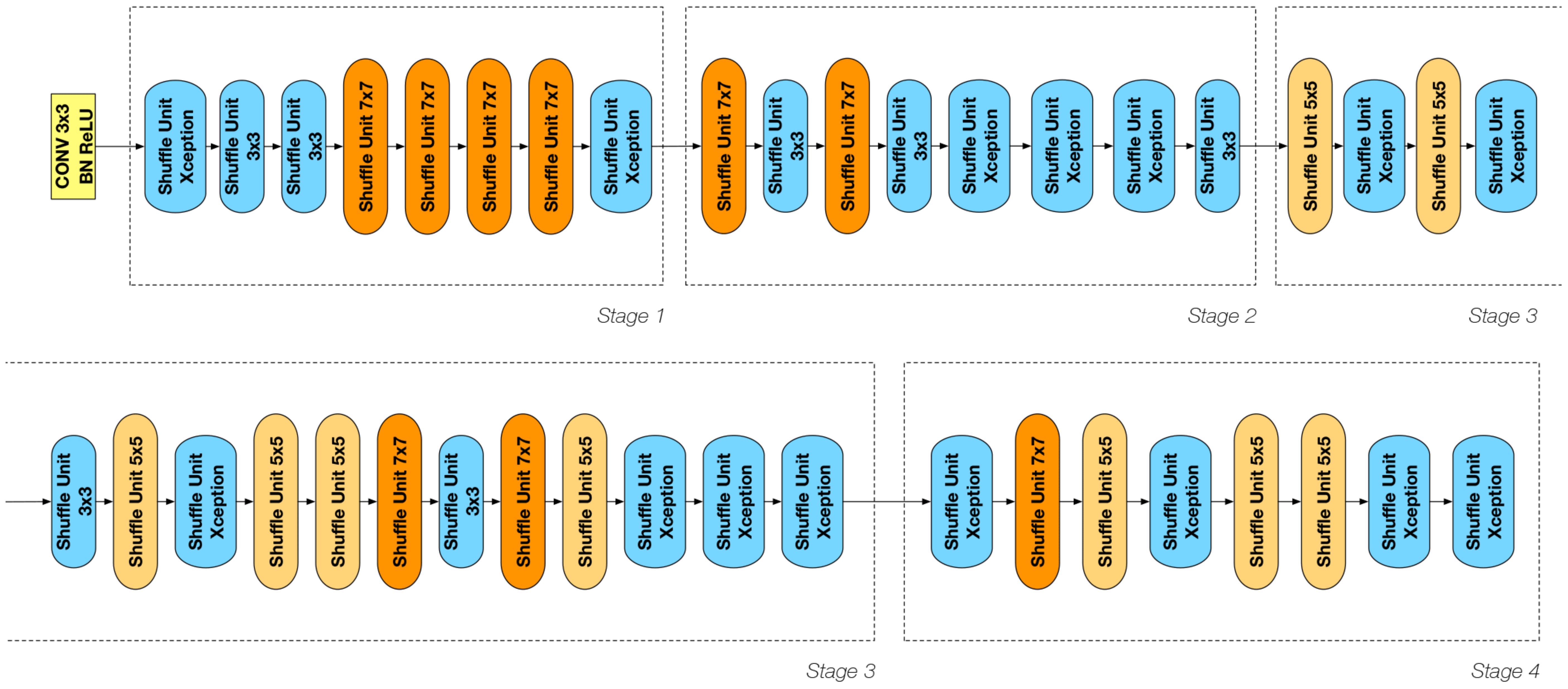
# Search Algorithm

MEGVII 旷视

## ❖ Evolutionary search

- Sample & reuse the weights from super net
- Very efficient

Epoch 0




**Algorithm 1** Evolutionary Architecture Search

**Input**: *supernet weights* $W_{\mathcal{A}}$*, population size* $P$*, architecture constraints* $\mathcal{C}$*, max iteration* $\mathcal{T}$*, validation dataset* $D_{val}$

**Output**: *the architecture with highest validation accuracy under architecture constraints*

$P_0 := Initialize\_population(P, \mathcal{C})$;

$n := P/2$; # Crossover number

$m := P/2$; # Mutation number

$prob := 0.1$; # Probability to mutate

$\mathrm{Topk} := \emptyset$;

**for** $i = 1 : \mathcal{T}$ **do**

  $ACC_{i-1} := Inference(W_{\mathcal{A}}, D_{val}, P_{i-1})$;

  $\mathrm{Topk} := Update\_Topk(\mathrm{Topk}, P_{i-1}, ACC_{i-1})$;

  $P_{crossover} := Crossover(\mathrm{Topk}, n, \mathcal{C})$;

  $P_{mutation} := Mutation(\mathrm{Topk}, m, prob, \mathcal{C})$;

  $P_i := P_{crossover} \cup P_{mutation}$;

**end for**

**return** the entry with highest accuracy in Topk;

# Results

- High performance
  - Significant improvements over commonly used backbones (e.g. ResNet 50) with fewer FLOPs
  - Best classification backbones may be suboptimal for object detection

Table 2: Main result comparisons.

| Backbone | ImageNet Classification FLOPs | ImageNet Classification Accuracy | Object Detection with FPN on COCO mAP | $AP_{50}$ | $AP_{75}$ | $AP_s$ | $AP_m$ | $AP_l$ |
|---|---|---|---|---|---|---|---|---|
| ResNet-50 | 3.8G | 76.15 | 37.3 | 58.2 | 40.8 | 21.0 | 40.2 | 49.4 |
| ShuffleNetv2-40 | 1.3G | 77.18 | 39.2 | 60.8 | 42.4 | 23.6 | 42.3 | 52.2 |
| ResNet-101 | 7.6G | 77.37 | 40.0 | 61.4 | 43.7 | 23.8 | 43.1 | 52.2 |
| DetNASNet | 1.3G | 77.20 | 40.0 | 61.5 | 43.6 | 23.3 | 42.5 | 53.8 |
| DetNASNet (3.8) | 3.8G | 78.44 | **42.0** | 63.9 | 45.8 | 24.9 | 45.1 | 56.8 |

Table 3: Ablation studies.

| | ImageNet (Top1 Acc, %) | COCO (mmAP, %) FPN | COCO (mmAP, %) RetinaNet | VOC (mAP, %) FPN | VOC (mAP, %) RetinaNet |
|---|---|---|---|---|---|
| ShuffleNetv2-20 | 73.1 | 34.8 | 32.1 | 80.6 | 79.4 |
| ClsNASNet | **74.3** | 35.1 | 31.2 | 78.5 | 76.5 |
| DetNAS-scratch | 73.8 - 74.3 | 35.9 | 32.8 | 81.1 | 79.9 |
| DetNAS | 73.9 - 74.1 | **36.4** | **33.3** | **81.5** | **80.1** |

# Results

MEGVII 旷视

- Search cost
  - Super nets greatly speed up search progress!

Table 5: Computation cost for each step on COCO.

| | Supernet pre-training | Supernet fine-tuning | Search on the supernet |
|---|---|---|---|
| DetNAS | $3 \times 10^5$ iterations | $9 \times 10^4$ iterations | $20 \times 50$ models |
| | 8 GPUs on 1.5 days | 8 GPUs on 1.5 days | 20 GPUs on 1 day |

* For the small search space, GPUs are GTX 1080Ti . For the large search space, GPUs are Tesla V100.

# Search for Detection Systems

MEGVII 旷视

**Backbone**

**Feature Fusion**

NAS-FPN

**Augmentation**

Ghaisi et al. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection

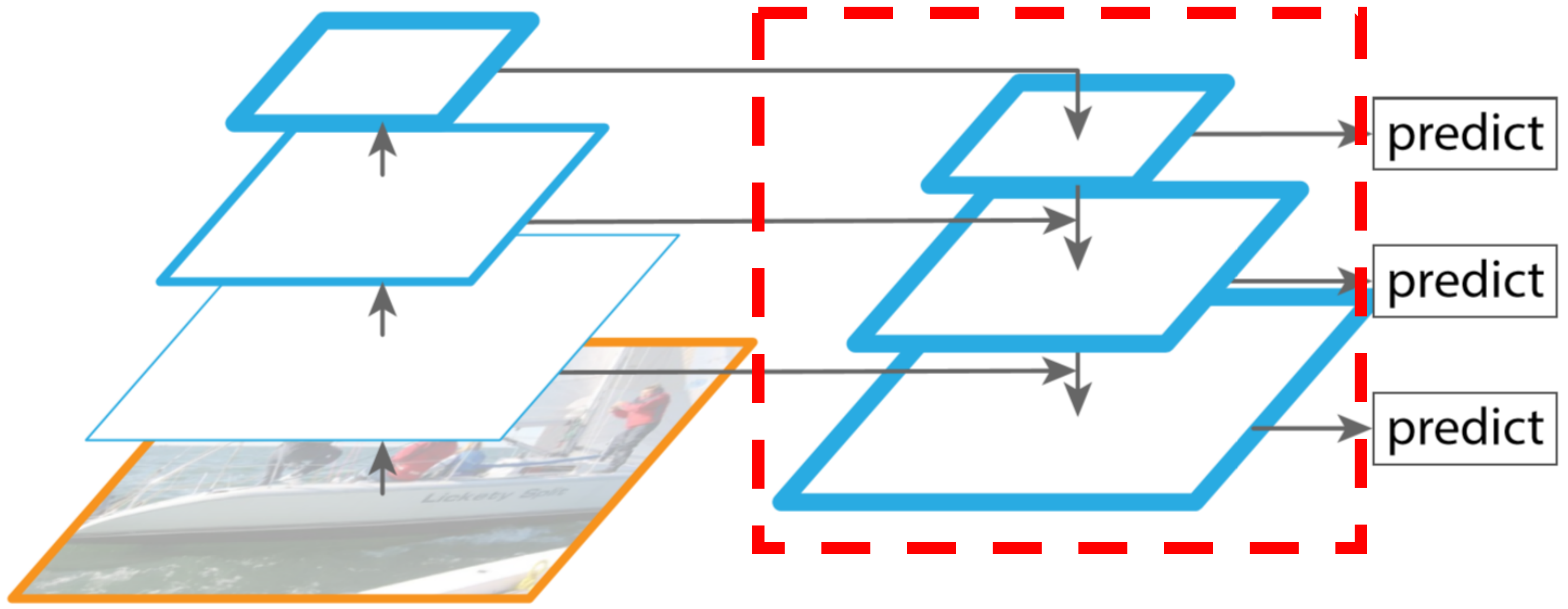
# Feature Fusion Modules

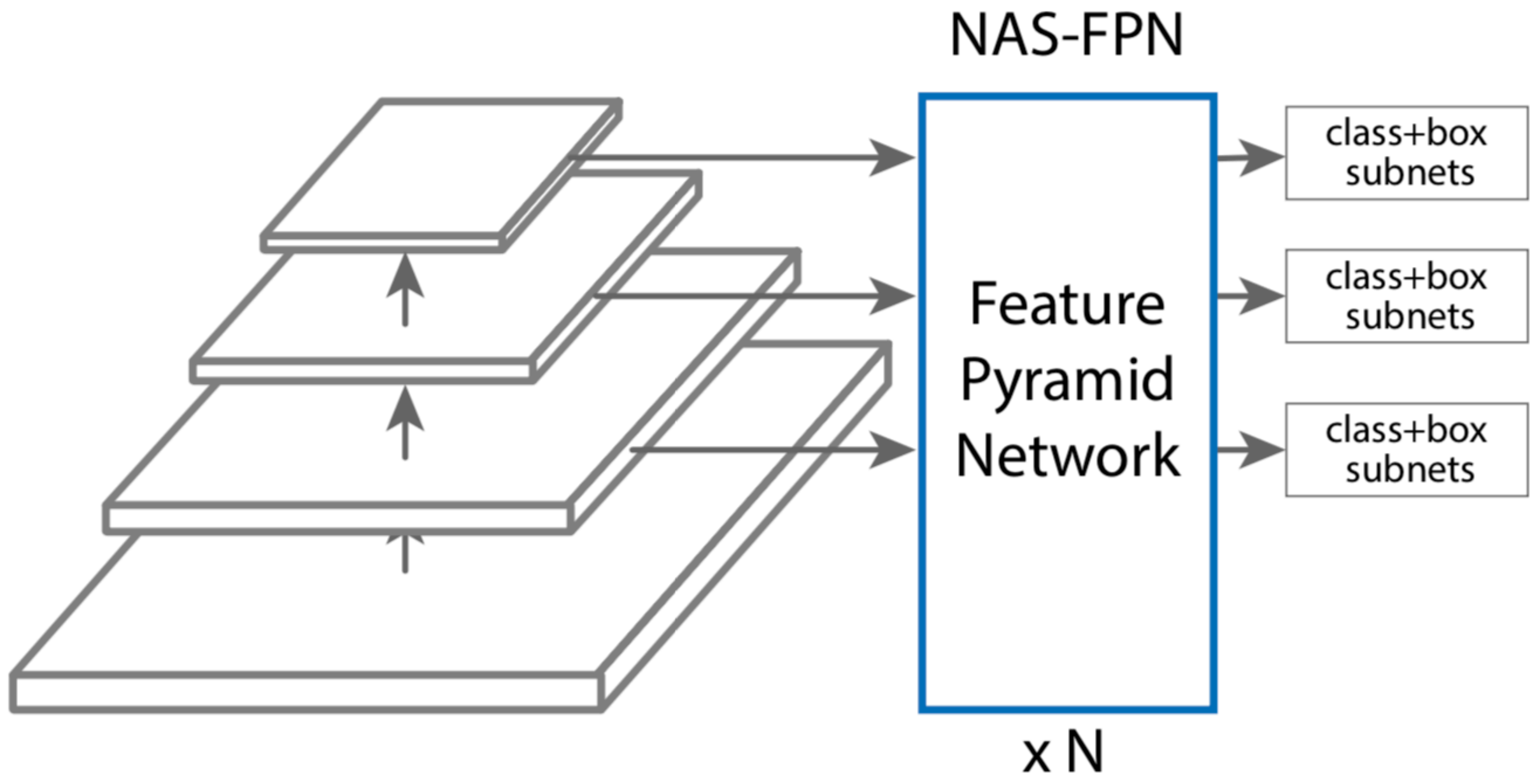
MEGVII 旷视

❖ Multi-scale feature fusion

- Used in state-of-the-art detectors (e.g. SSD, FPN, SNIP, FCOS, …)

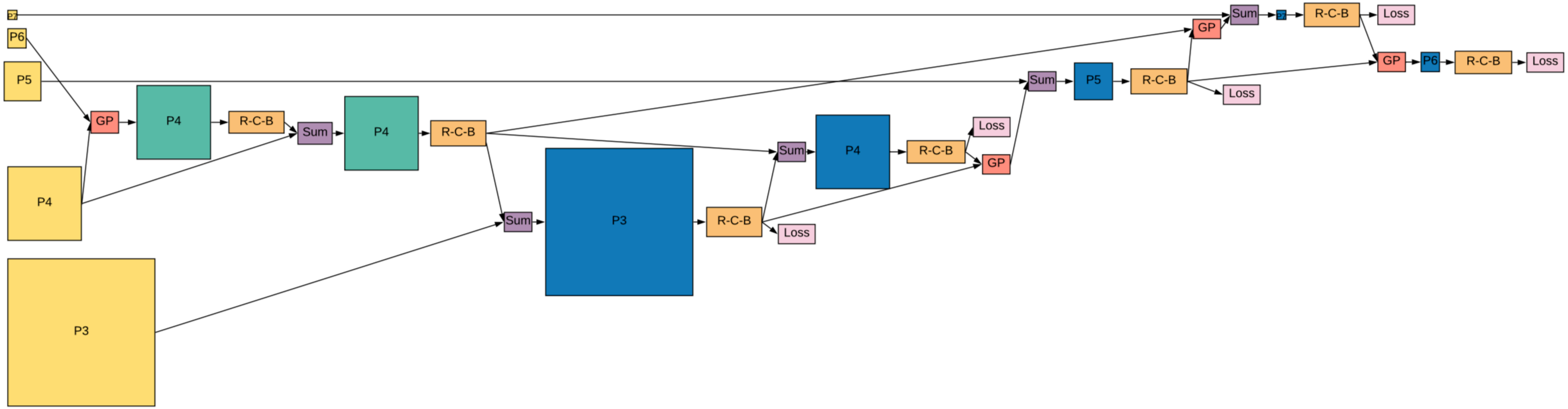
❖ Automatic search vs. manual design



# First Glance

- Searched architecture
  - Very different from handcraft structures

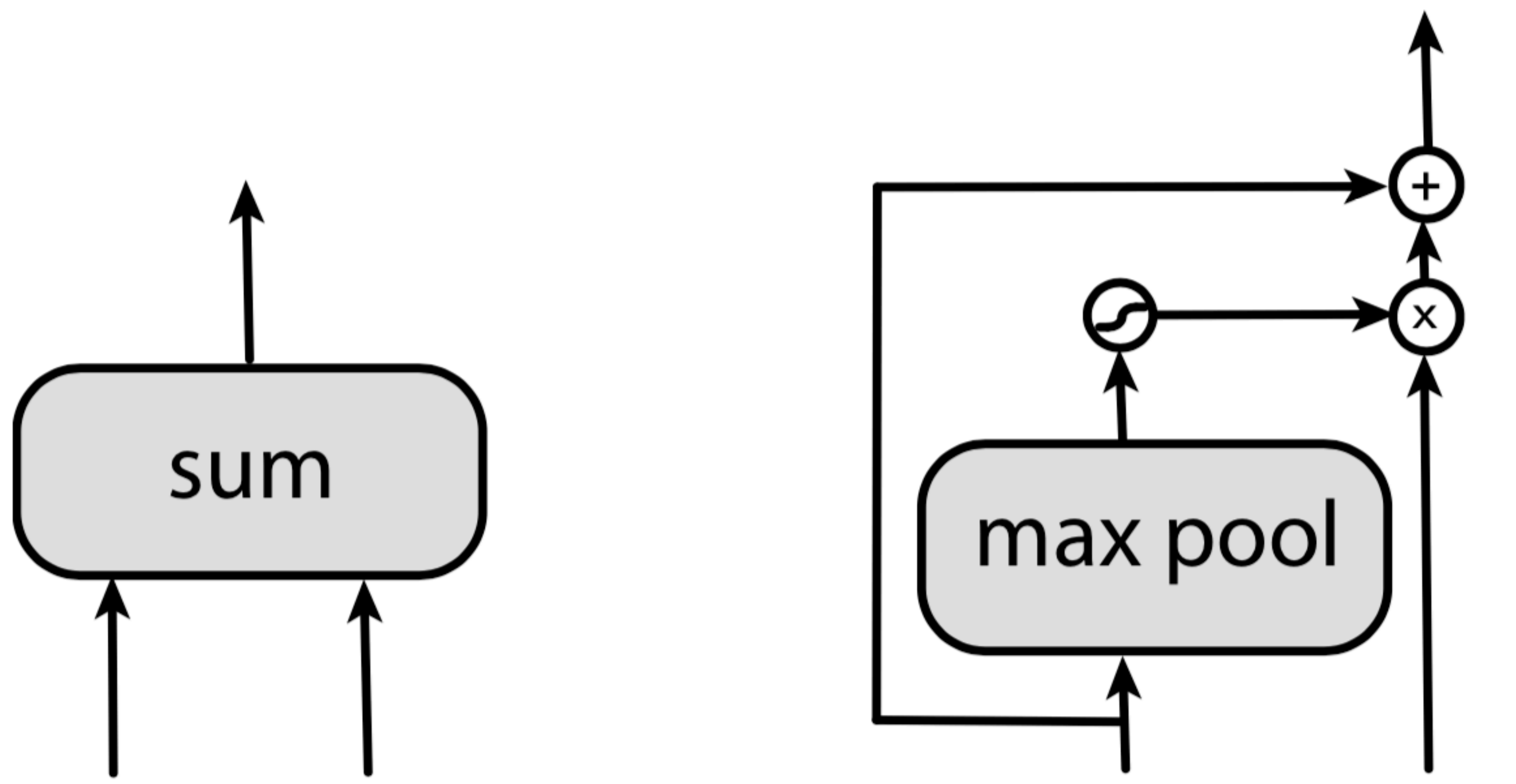


# Search Space

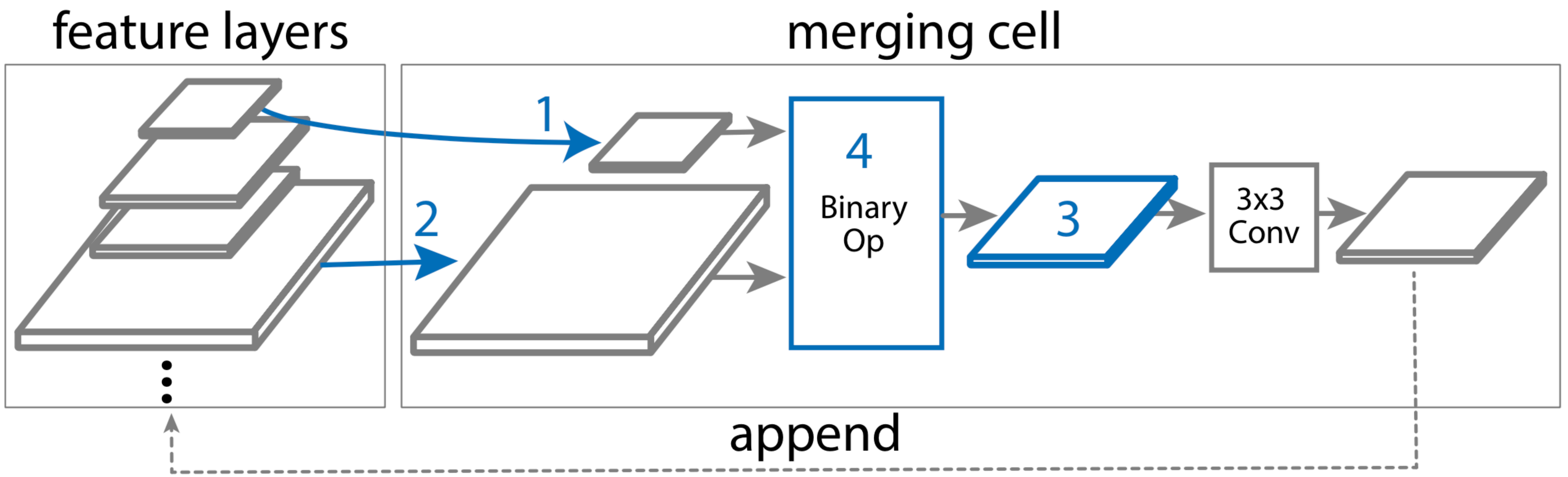
❖ Stacking repeated FPN blocks

❖ For each FPN block, N different merging cells

❖ For each merging cell, 4-step generations




(a) Sum (b) Global pooling

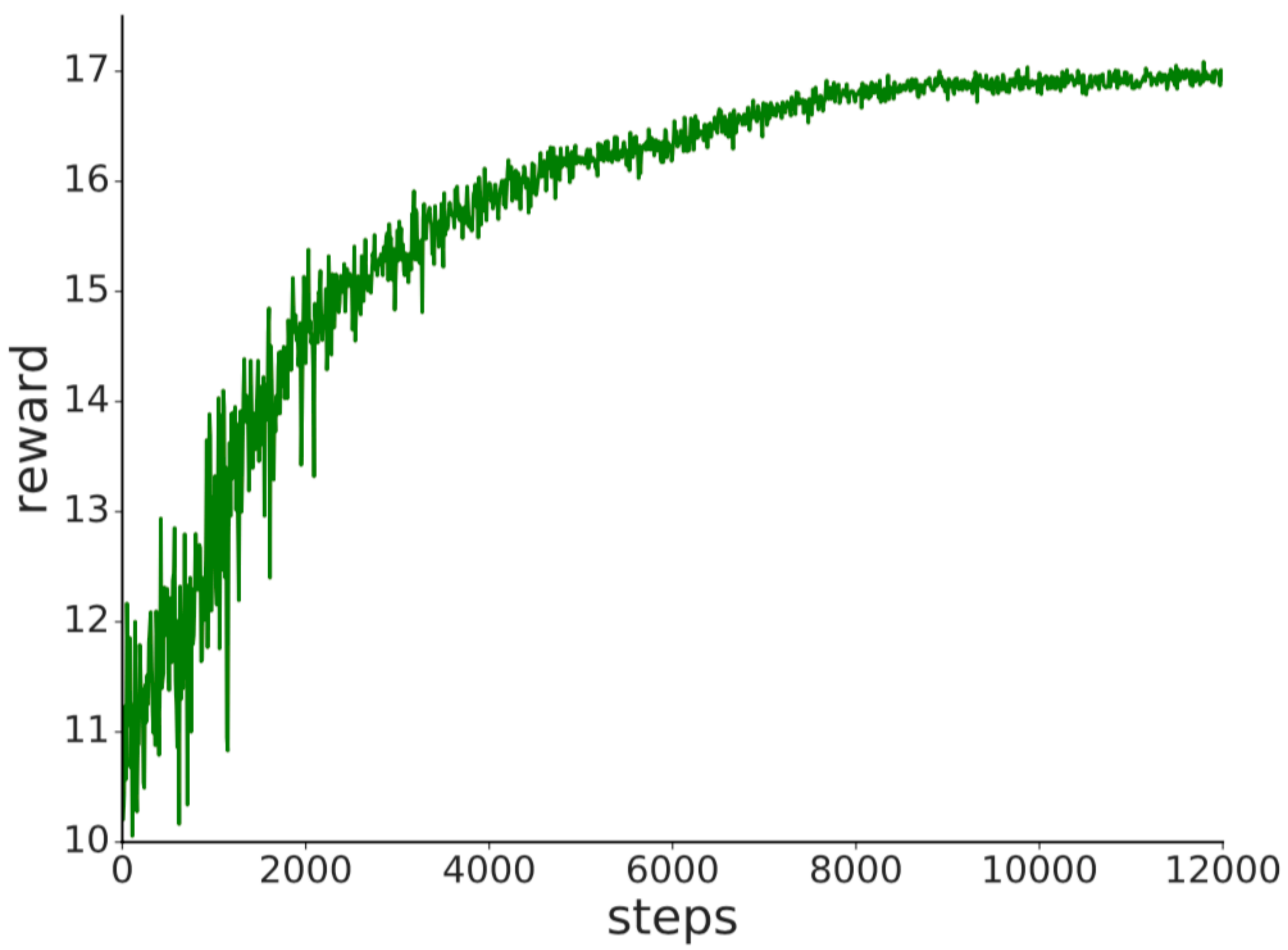


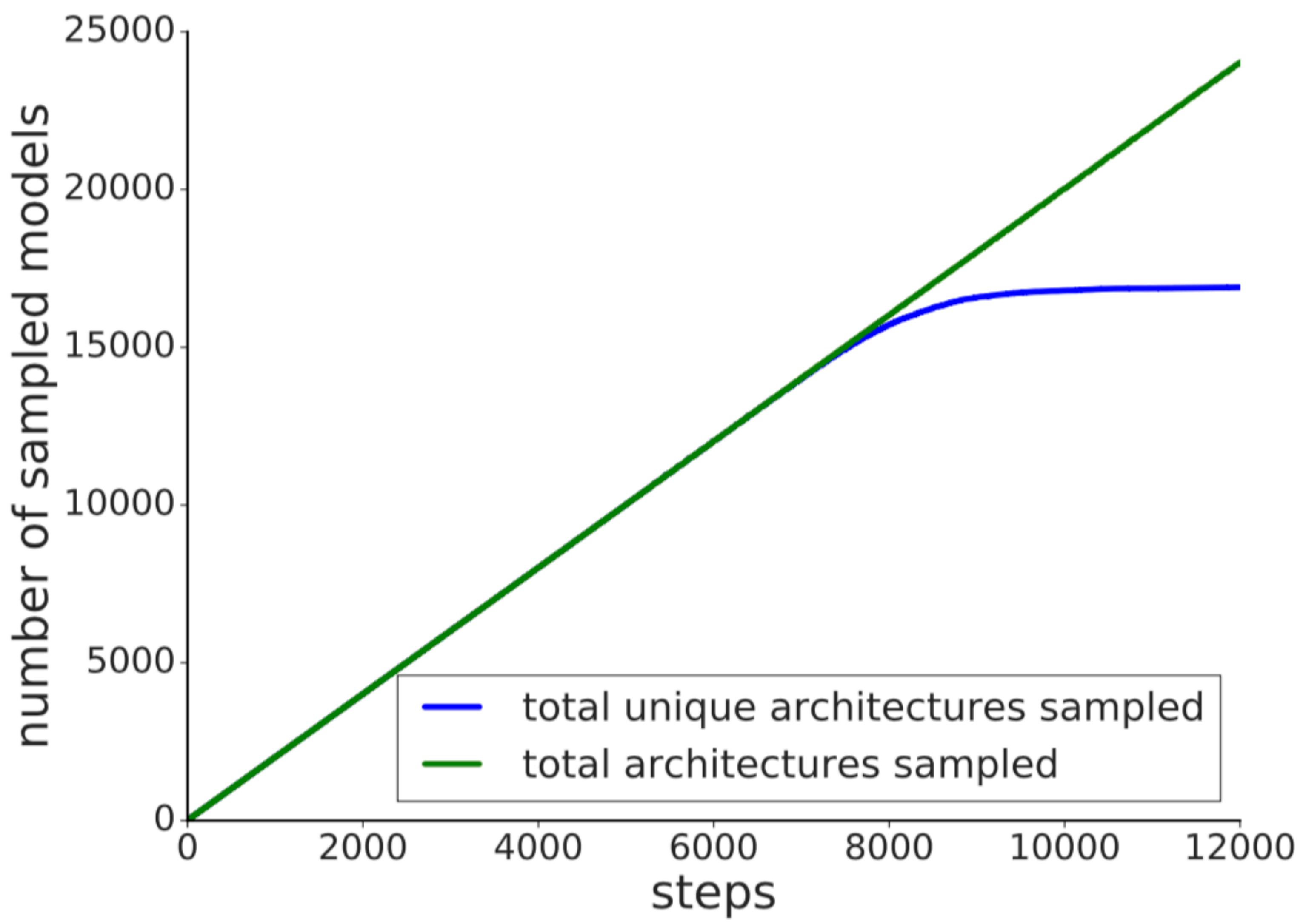
# Search Algorithm

MEGVII 旷视

- Controller
  - RNN-based controller
  - Search with Proximal Policy Optimization (PPO)

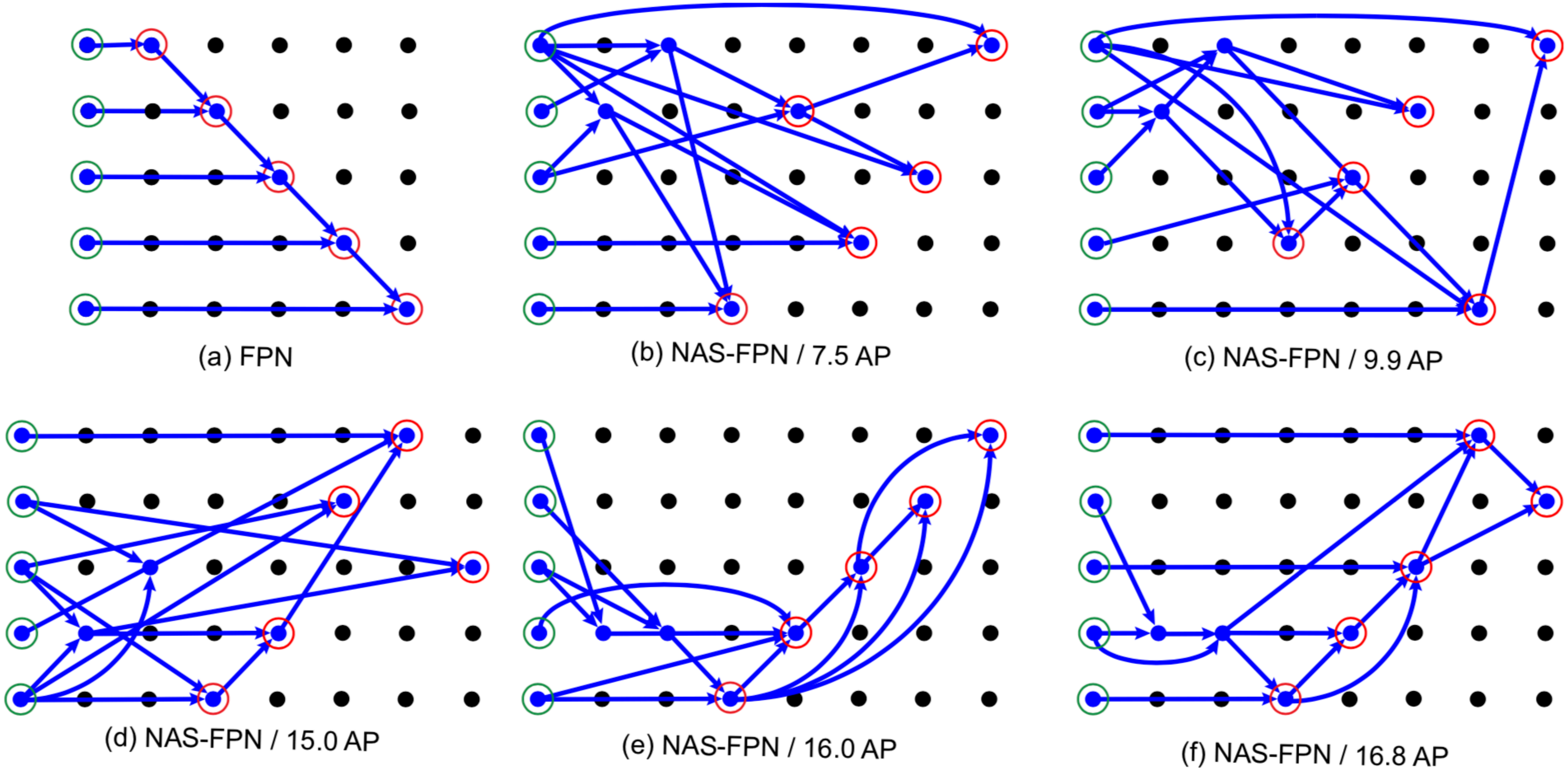
- Candidate evaluation
  - Training a light-weight proxy task



# Architectures During Search

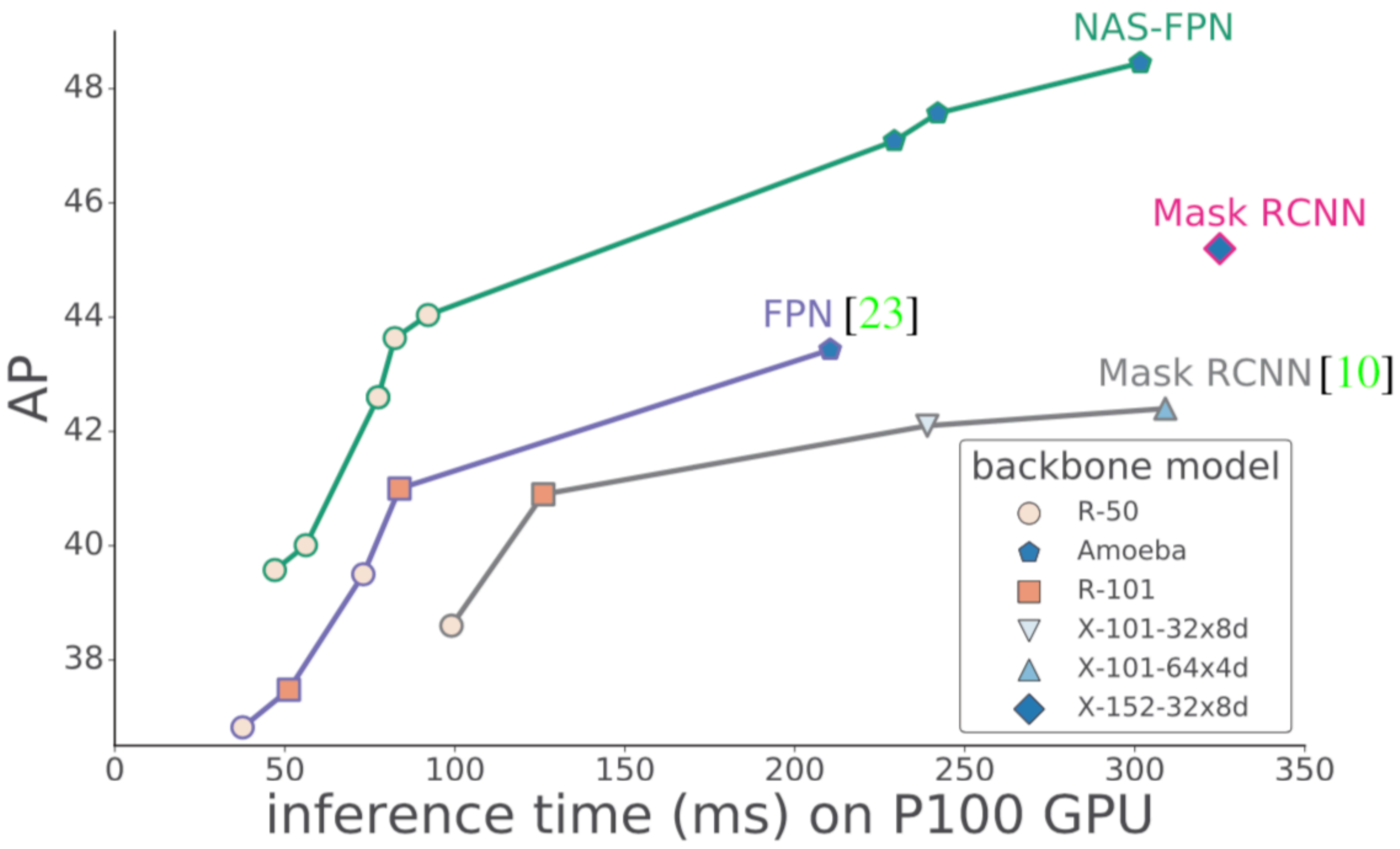
❖ Many downsamples and upsamples

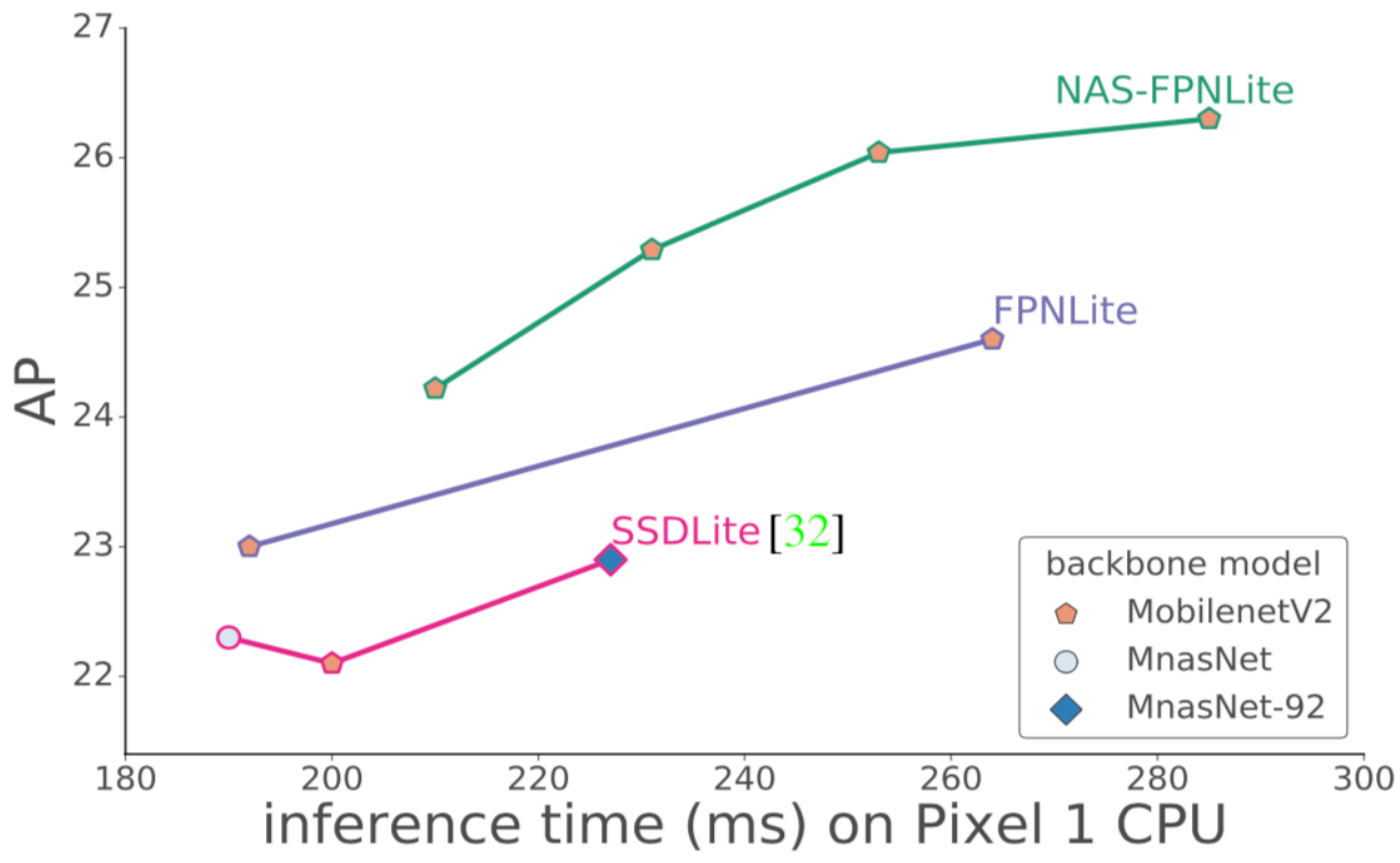


# Results

MEGVII 旷视

❖ State-of-the-art speed/AP trade-off



# Search for Detection Systems

MEGVII 旷视

**Backbone**

**Feature Fusion**

**Augmentation**

Auto-Augment for Detection

Zoph et al. Learning Data Augmentation Strategies for Object Detection

# Data Augmentation for Object Detection

- Augmentation pool
  - Color distortions
  - Geometric transforms
  - Random noise (e.g. cutout, drop block, …)
  - Mix-up

  …

- Search for the best augmentation configurations

# Search Space Design

❖ Mainly follows AutoAugment

❖ Randomly sampling from K sub-policies

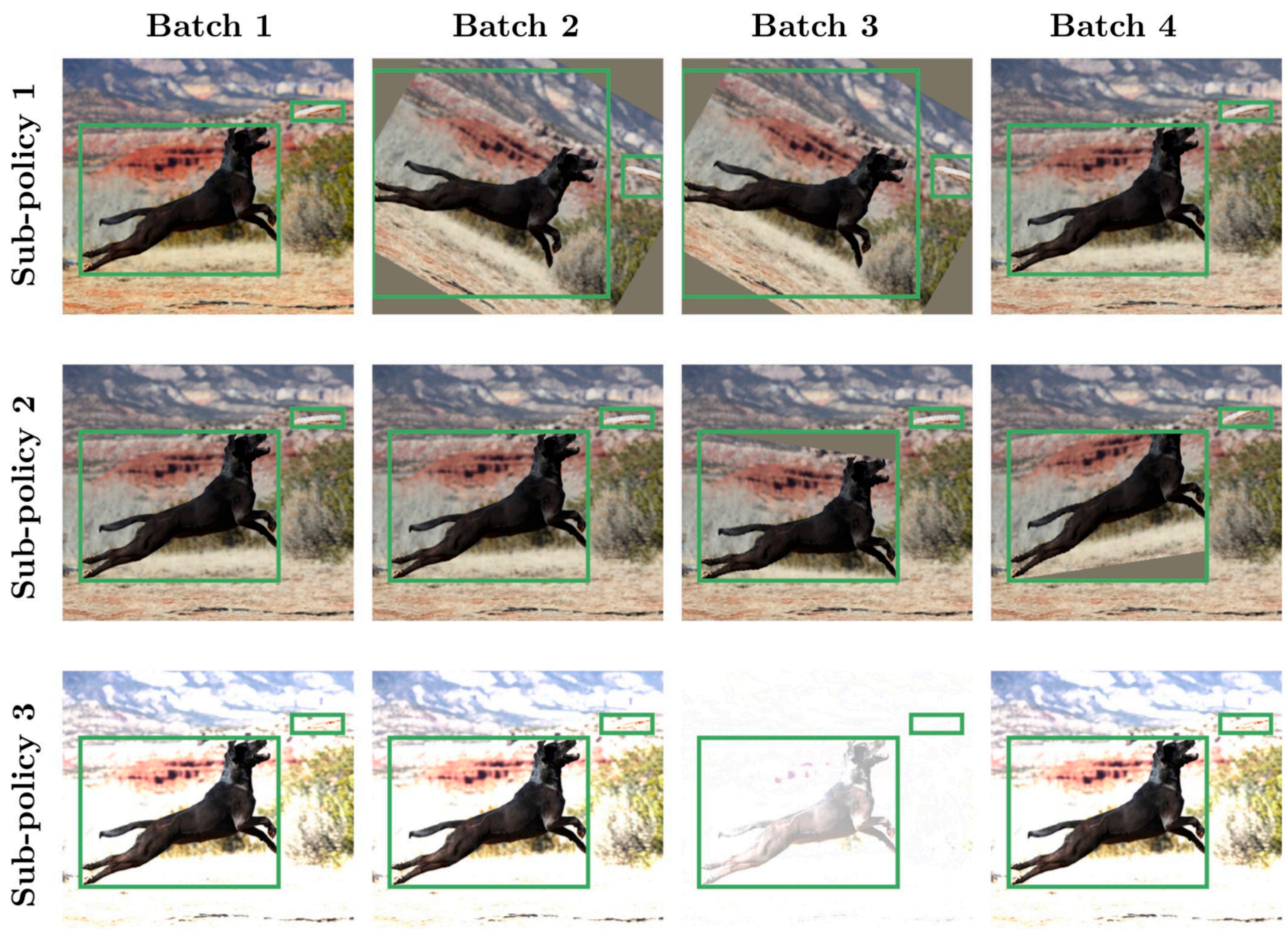
❖ For each sub-policy, N image transforms

❖ Each image transform selected from 22 operations:

- Color operations
- Geometric operations
- Bounding box operations

Cubuk et al. AutoAugment: Learning Augmentation Strategies from Data

# Search Space Design (cont’d)




Sub-policy 1. (Color, 0.2, 8), (Rotate, 0.8, 10)
Sub-policy 2. (BBox_Only_ShearY, 0.8, 5)
Sub-policy 3. (SolarizeAdd, 0.6, 8), (Brightness, 0.8, 10)
Sub-policy 4. (ShearY, 0.6, 10), (BBox_Only_Equalize, 0.6, 8)
Sub-policy 5. (Equalize, 0.6, 10), (TranslateX, 0.2, 2)

# Search Algorithm

- Very similar to NAS-FPN
- Controller
  - RNN-based controller
  - Search with Proximal Policy Optimization (PPO)

- Evaluation
  - A small proxy dataset
  - Short-time training

# Results

❖ Significantly outperforms previous state-of-the-arts

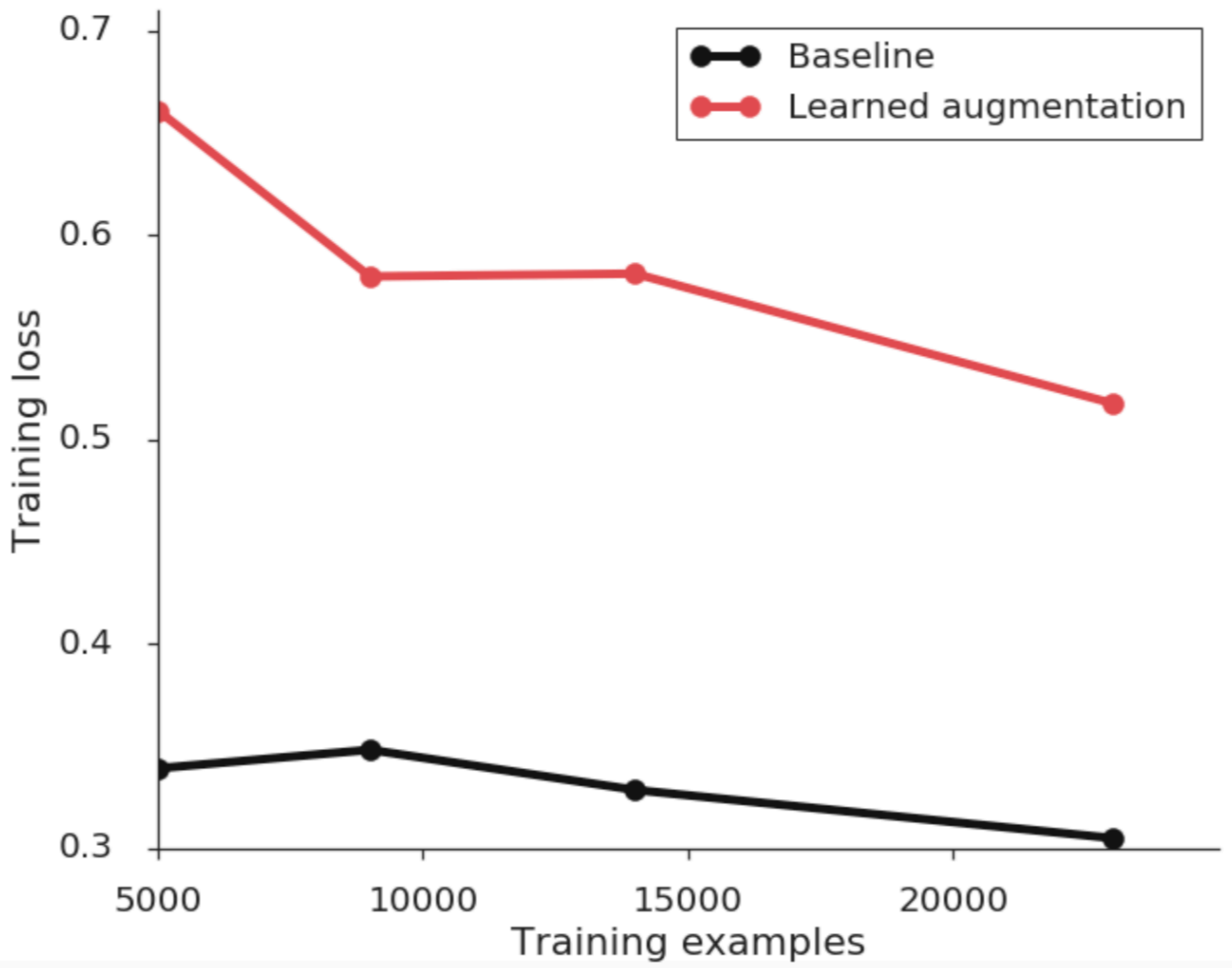
| Backbone | Baseline | Our result | Difference |
|---|---|---|---|
| ResNet-50 | 36.7 | 39.0 | +2.3 |
| ResNet-101 | 38.8 | 40.4 | +1.6 |
| ResNet-200 | 39.9 | 42.1 | +2.2 |

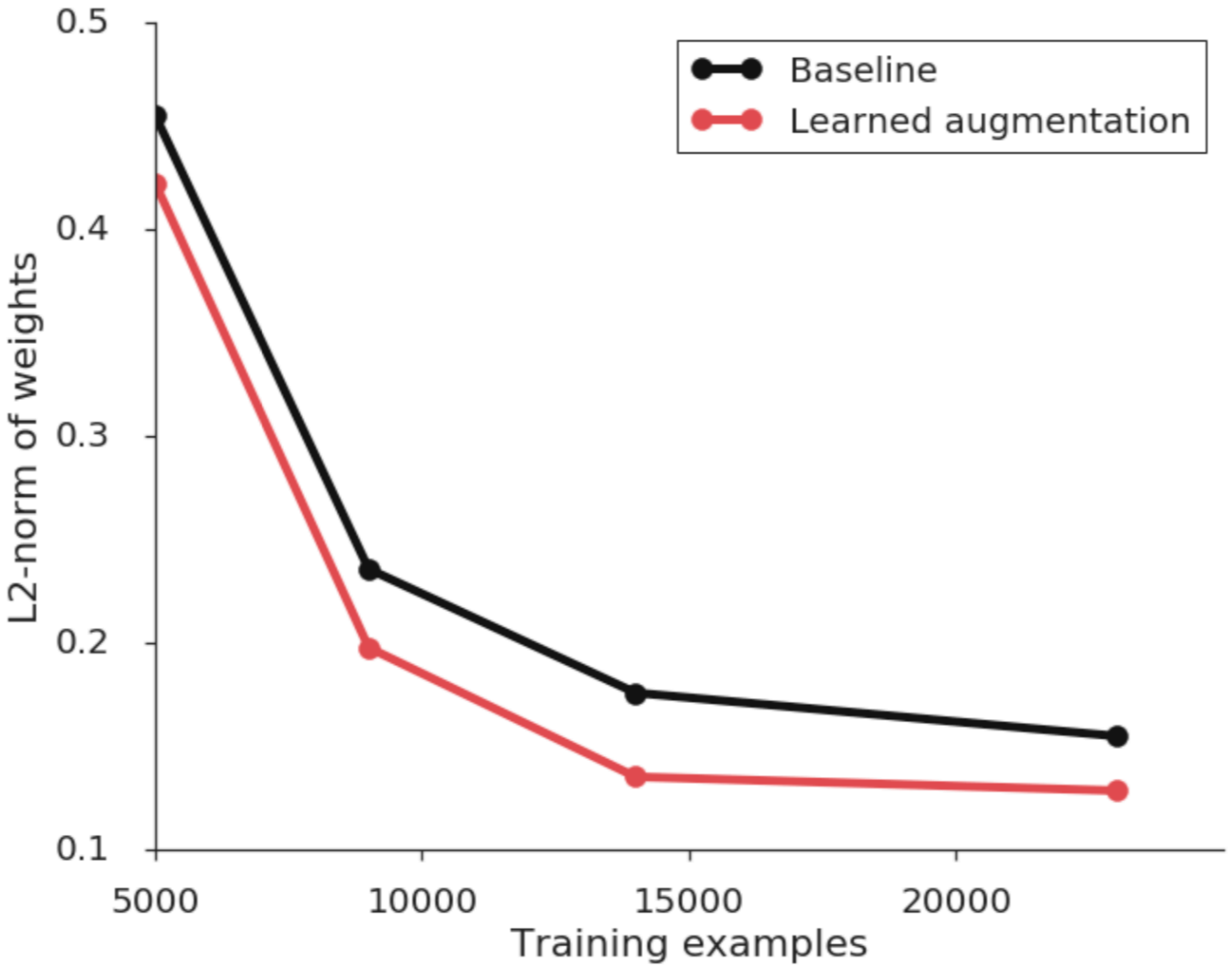
| Method | mAP |
|---|---|
| baseline | 36.7 |
| baseline + DropBlock [13] | 38.4 |
| Augmentation policy with color operations | 37.5 |
| + geometric operations | 38.6 |
| + bbox-only operations | **39.0** |

| Architecture | Change | # Scales | mAP | $mAP_S$ | $mAP_M$ | $mAP_L$ |
|---|---|---|---|---|---|---|
| MegDet [32] | | multiple | 50.5 | - | - | - |
| AmoebaNet + NAS-FPN | baseline [14] | 1 | 47.0 | 30.6 | 50.9 | 61.3 |
| | + learned augmentation | 1 | 48.6 | 32.0 | 53.4 | 62.7 |
| | + ↑ anchors, ↑ image size | 1 | **50.7** | **34.2** | **55.5** | **64.5** |

# Analysis

❖ Better regularization



# Future Work

MEGVII 旷视

- More search dimensions
  - E.g. loss, anchor boxes, assign rules, post-processing, …
- Reducing search cost
- Joint optimization

MEGVII 旷视

# Q & A