Security Level:

神经网络小型化方法

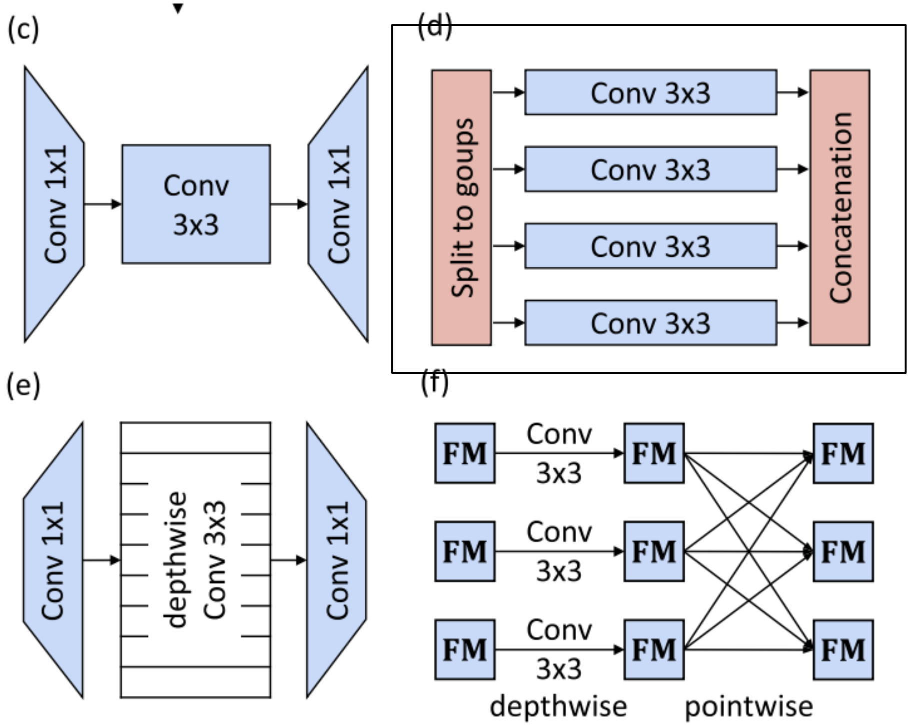
www.huawei.com

HUAWEI TECHNOLOGIES CO., LTD.

目录

- 设计紧凑网络
  - Heuristic Design
  - Neural Architecture Search (NAS)
- Tensor分解
- 量化
- 剪枝
- 知识蒸馏
- 总结

Heuristic Design Overview



分组卷积

思路：在通道上做信息分解。

优点：降低模型参数和计算量；

缺点：每组的卷积所能见到的信息比较固定，需要额外的模块做信息融合。

e.g. ResNext, ShuffleNet

MobileNet

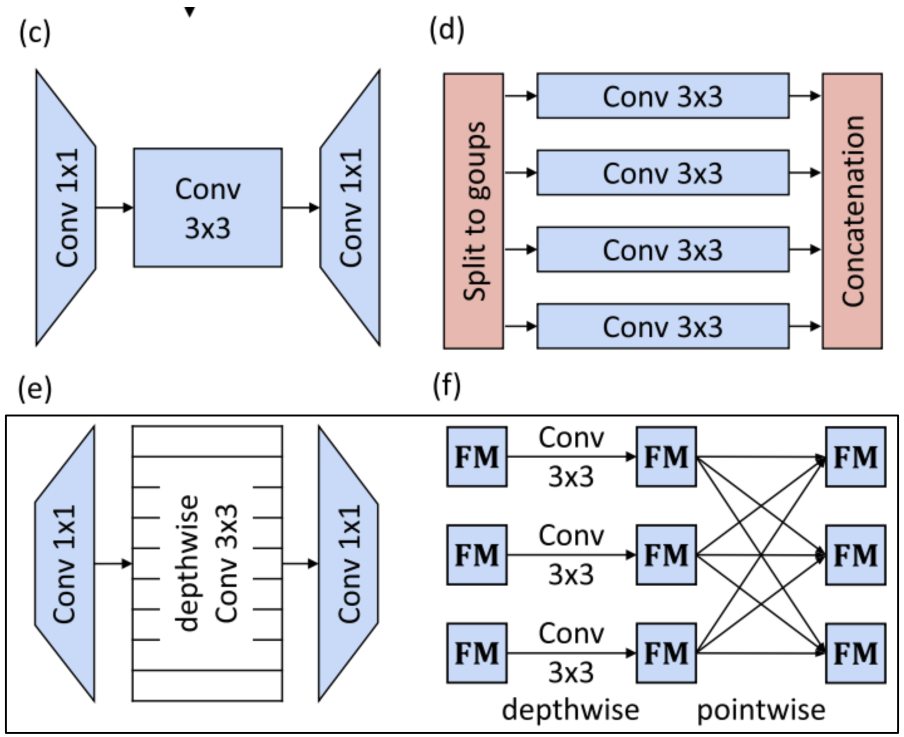
深度可分离卷积

做法：在通道上做信息分解

优点：参数量极大降低

缺点：depthwise卷积的计算的数据重用和locality 比较差，往往在没有memory bound的设备上计算速度比较慢。

e.g. MobileNet全家桶，EfficientNet



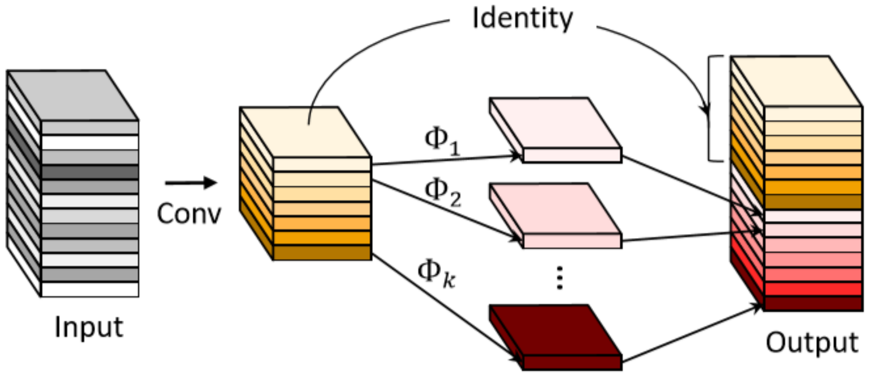
GhoseNet

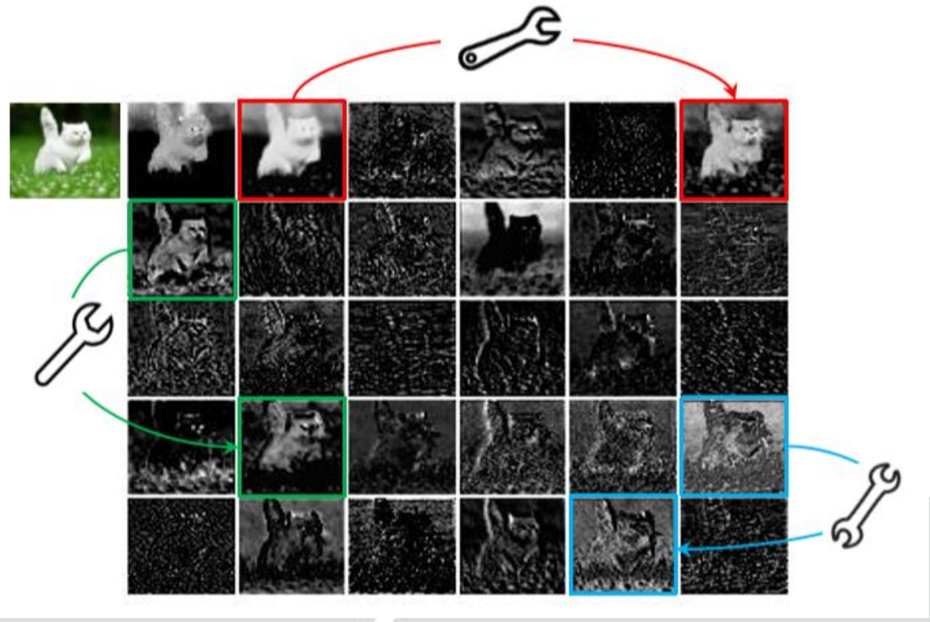
- 现状：卷积操作的输入、输出通道数比较大（256/512），计算量大

- 观察：有些feature map高度相似

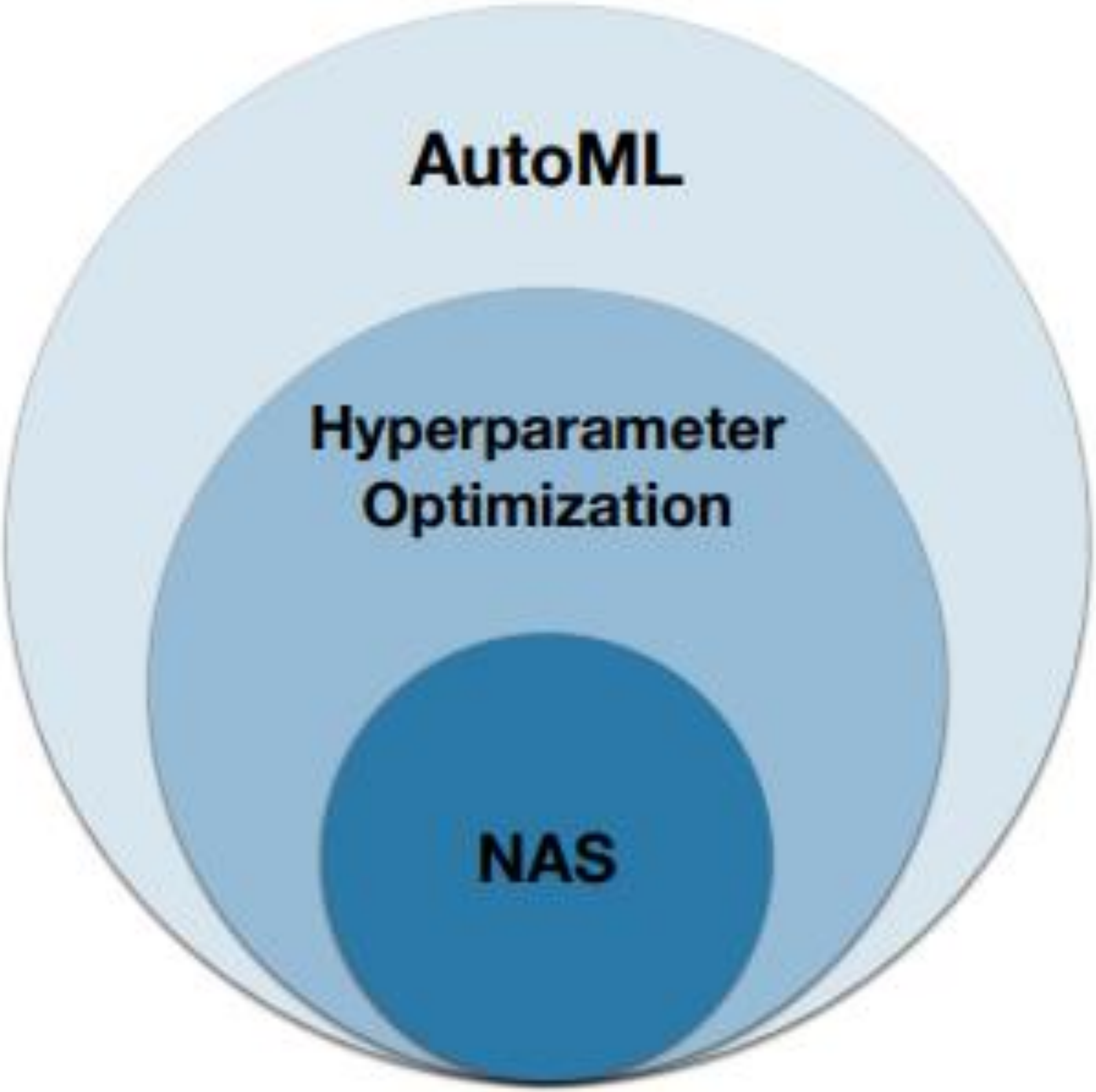
- 猜想：是否可以使用较少的feature map（基向量），通过低复杂度操作（cheap operations）增加feature map

- 实践：





AutoML Overview



Goal: Automate architecture design

Reality: Search through space of network architectures

NAS is a special case of HP Opt!

Search Space

Continuous
& Discrete

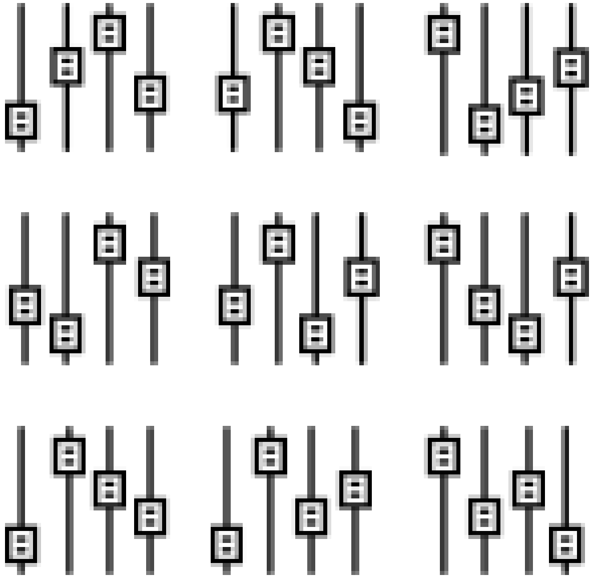
a: activation fct
u: nodes per layer
h: # hidden layers
r: regularization

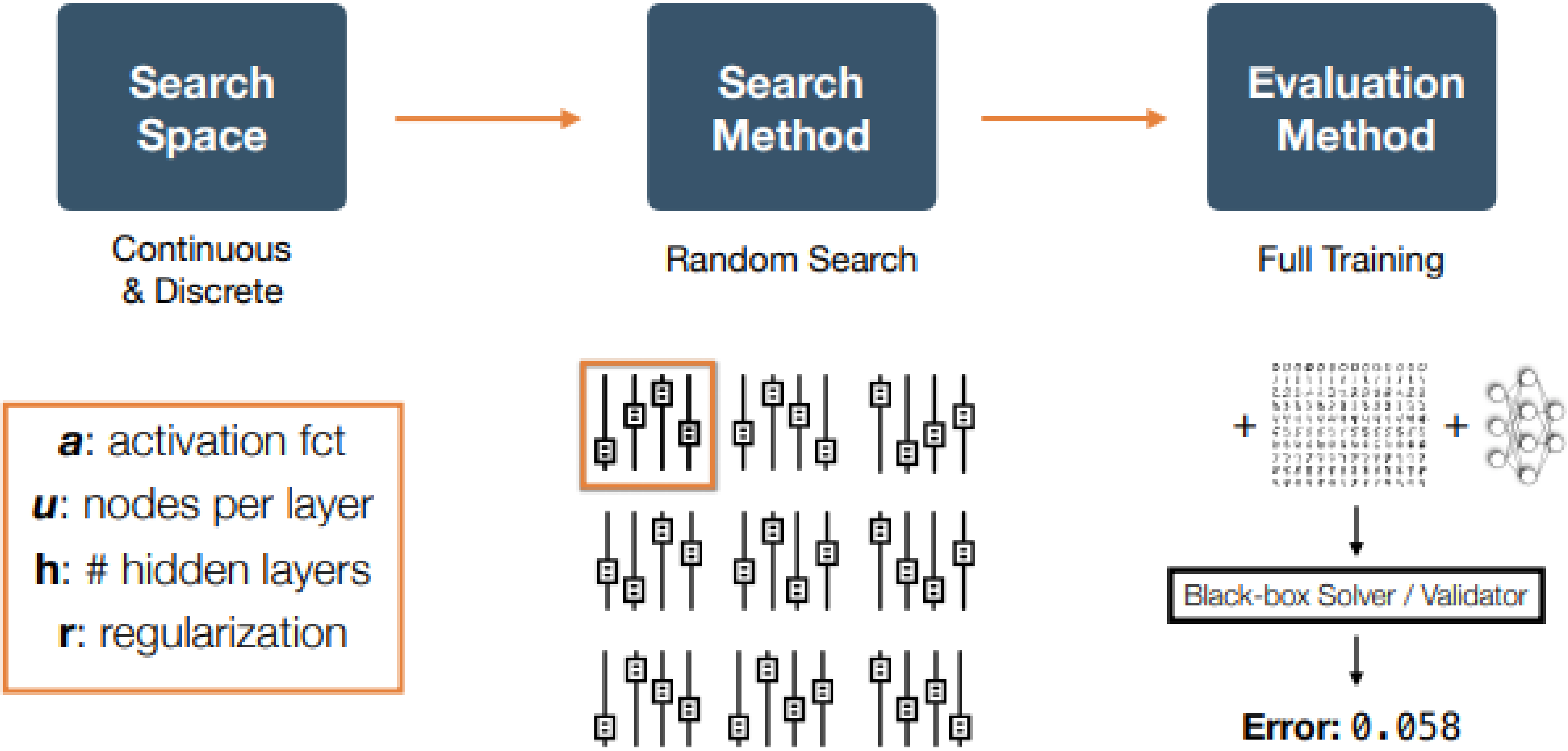
Search Space → **Search Method**

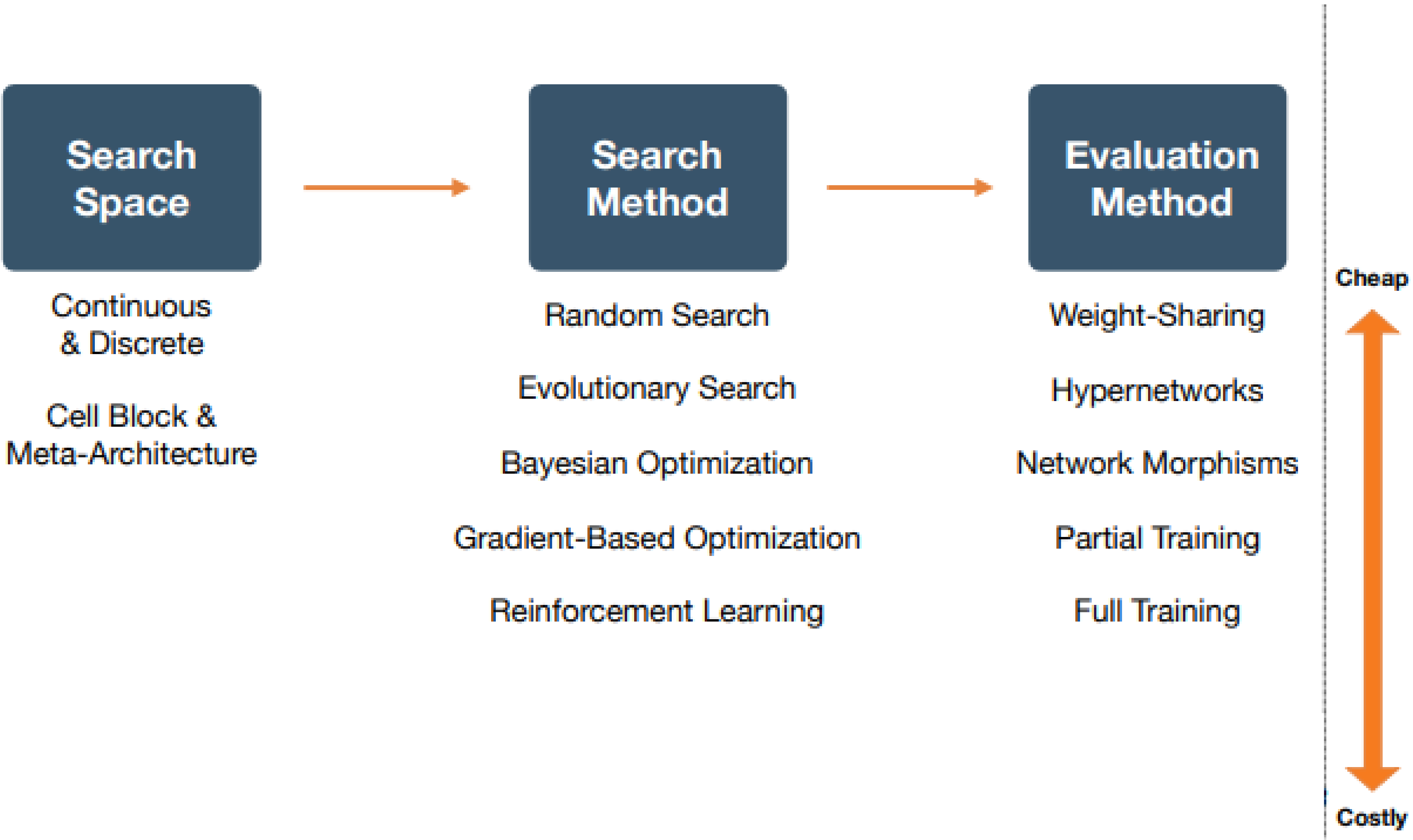
Continuous
& Discrete

Random Search

a: activation fct
u: nodes per layer
h: # hidden layers
r: regularization




Search Space
Search Method
Evaluation Method
Continuous & Discrete
Random Search
Full Training
a: activation fct
u: nodes per layer
h: # hidden layers
r: regularization
Black-box Solver / Validator
Error: 0.058


Search Space
Search Method
Evaluation Method
Continuous & Discrete
Cell Block & Meta-Architecture
Random Search
Evolutionary Search
Bayesian Optimization
Gradient-Based Optimization
Reinforcement Learning
Weight-Sharing
Hypernetworks
Network Morphisms
Partial Training
Full Training
Cheap
Costly

Traditional Hyperparameter Optimization

Search Space → **Search Method** → **Evaluation Method**

Search Space

Continuous
& Discrete

Search Method

Random Search

Evolutionary Search

Bayesian Optimization

Gradient-Based Optimization

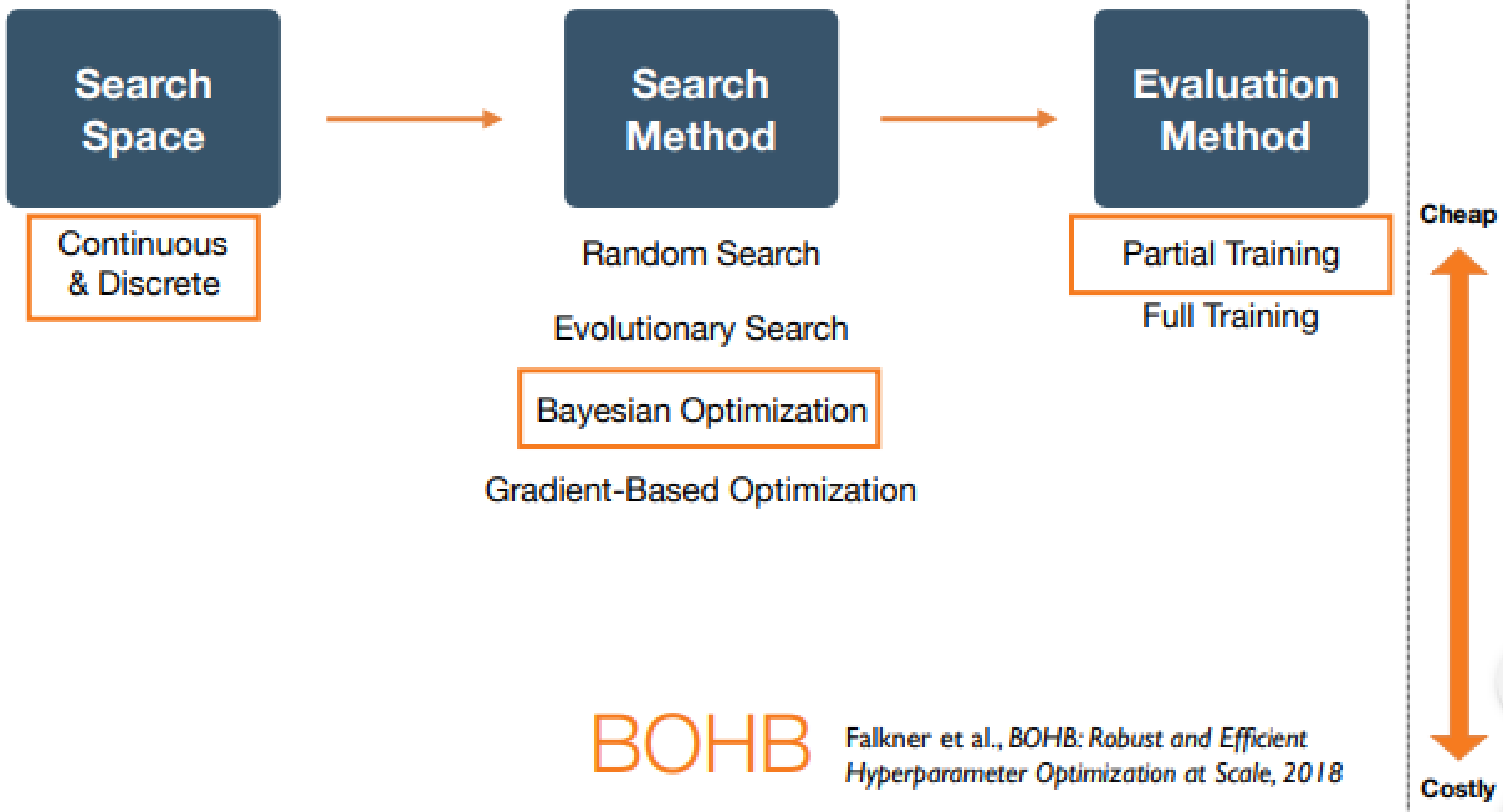
Evaluation Method

Partial Training

Full Training

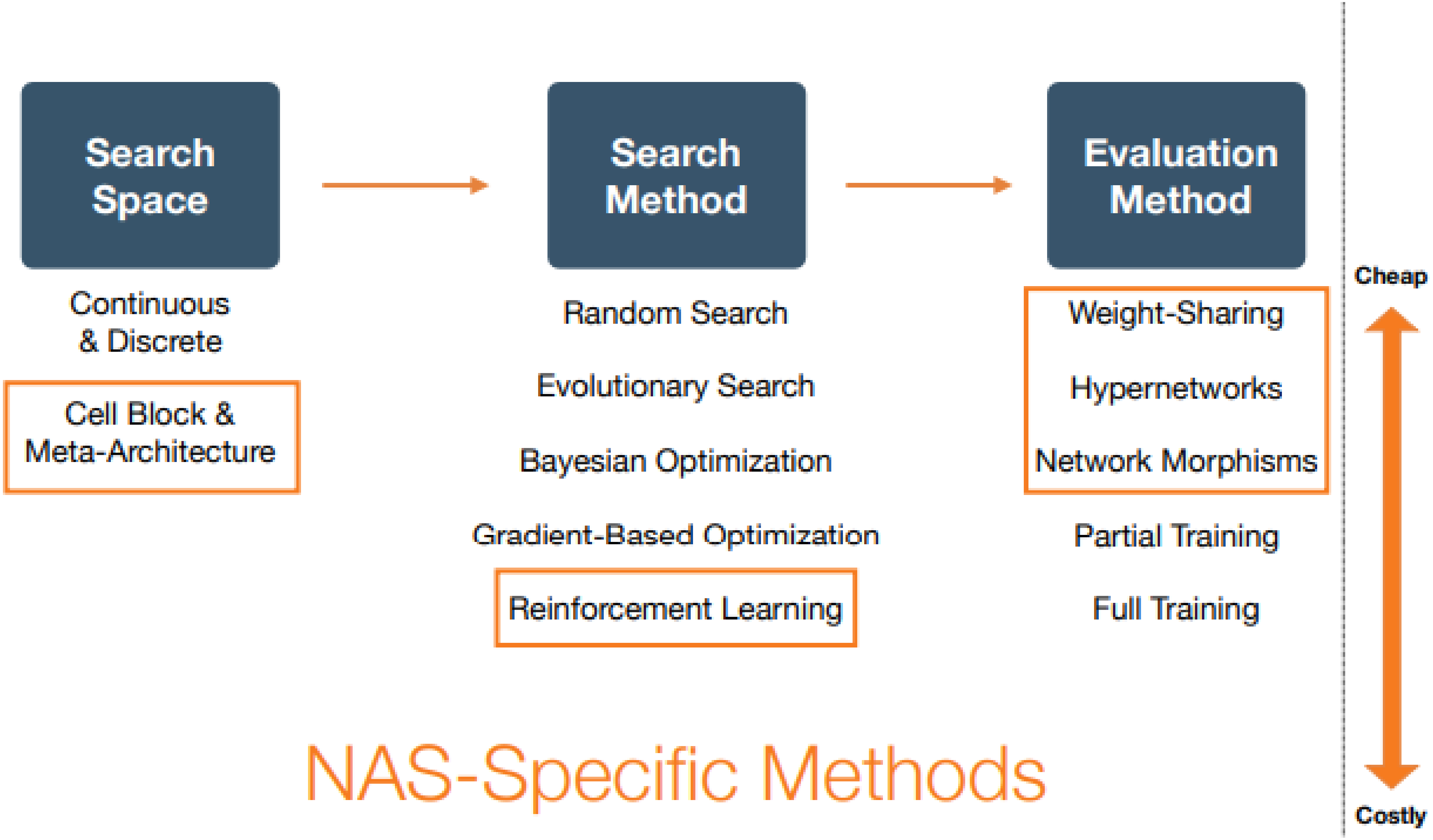
Cheap ↕ Costly

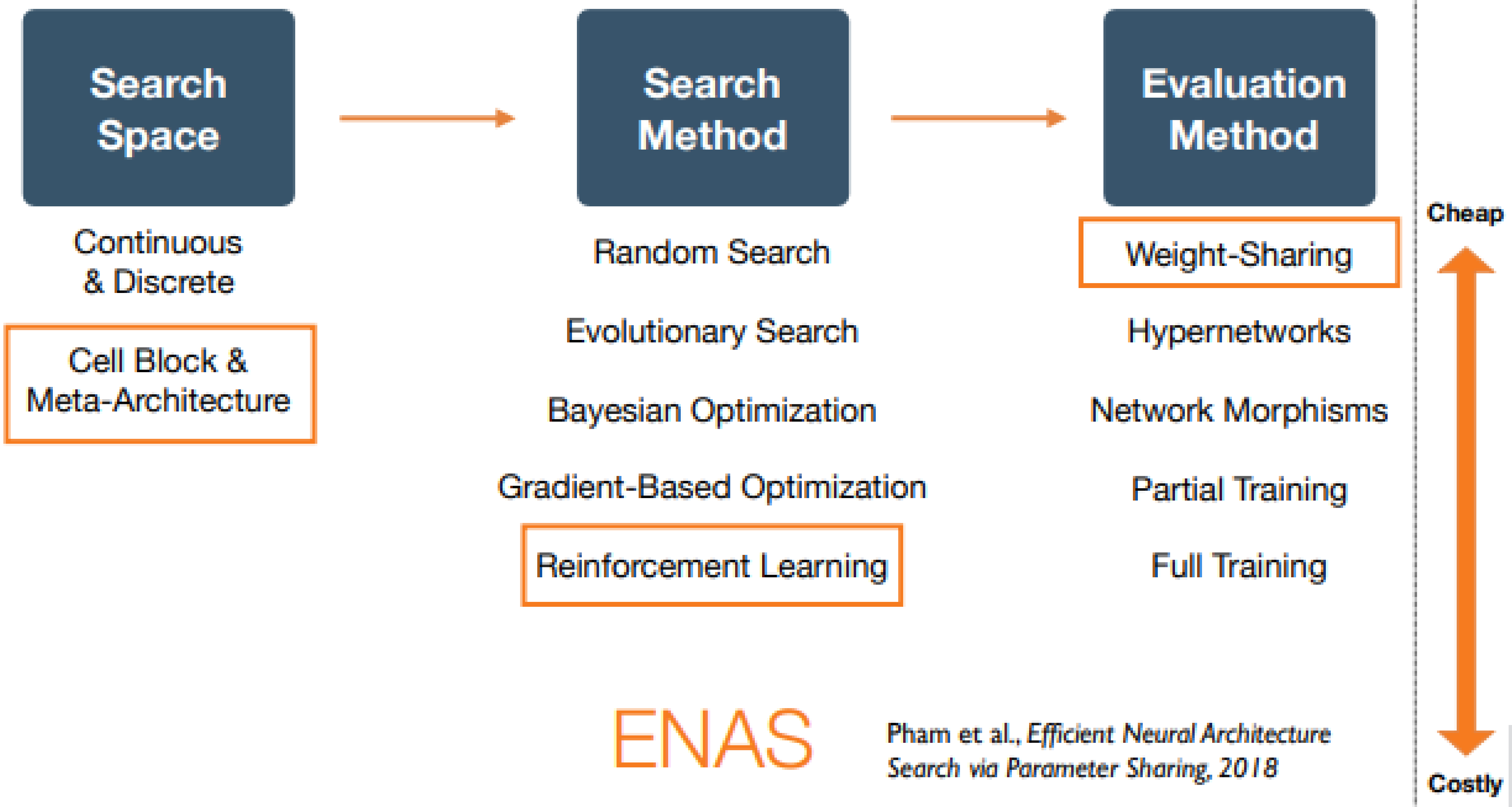
Elsken et al., *Neural Architecture Search: A Survey, 2018*

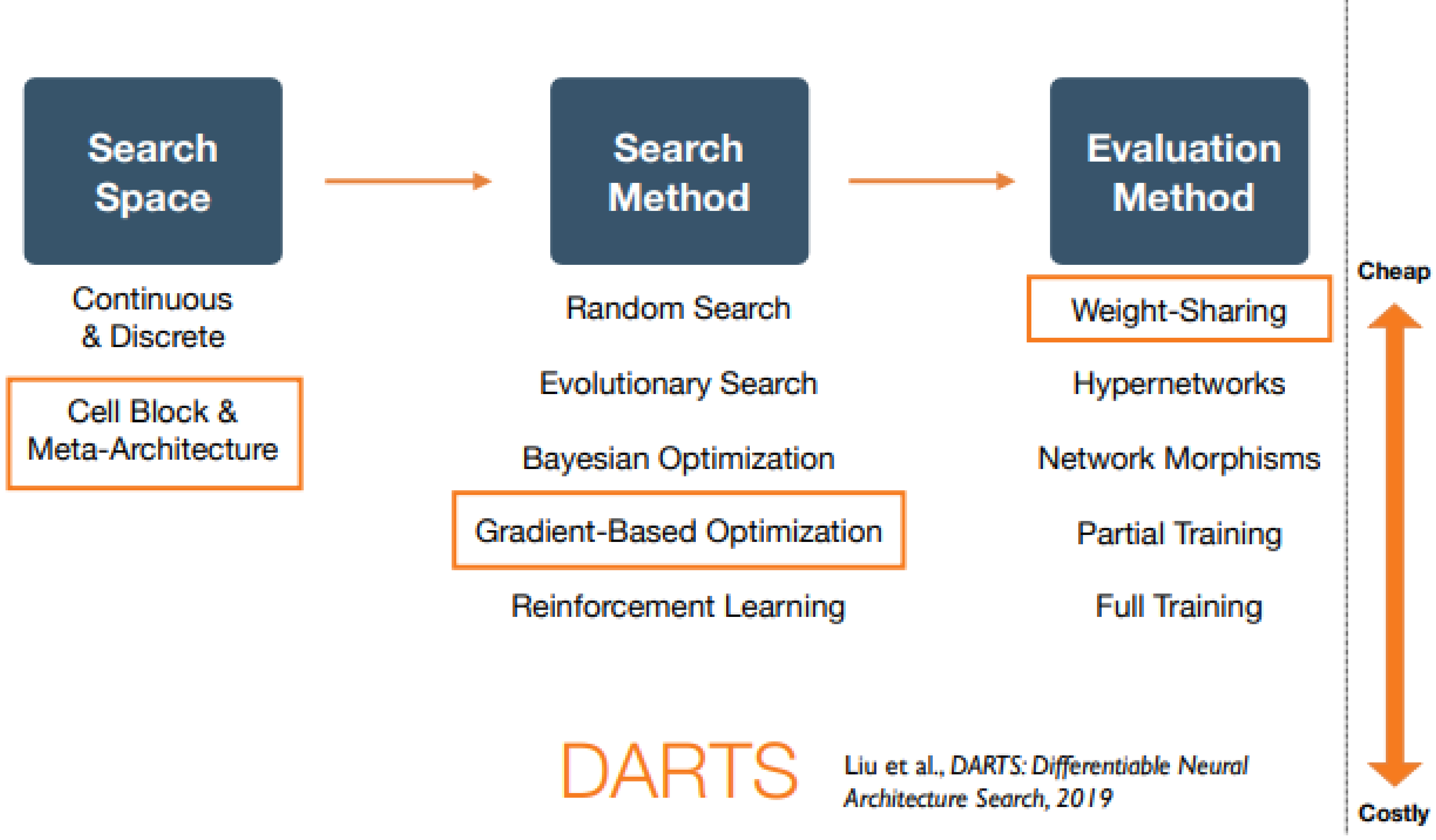


Falkner et al., *BOHB: Robust and Efficient Hyperparameter Optimization at Scale*, 2018

NAS-Specific Methods




Search Space
Search Method
Evaluation Method
Continuous & Discrete
Cell Block & Meta-Architecture
Random Search
Evolutionary Search
Bayesian Optimization
Gradient-Based Optimization
Reinforcement Learning
Weight-Sharing
Hypernetworks
Network Morphisms
Partial Training
Full Training
Cheap
Costly
ENAS
Pham et al., *Efficient Neural Architecture Search via Parameter Sharing*, 2018


Search Space
Search Method
Evaluation Method
Continuous & Discrete
Cell Block & Meta-Architecture
Random Search
Evolutionary Search
Bayesian Optimization
Gradient-Based Optimization
Reinforcement Learning
Weight-Sharing
Hypernetworks
Network Morphisms
Partial Training
Full Training
Cheap
Costly
DARTS
Liu et al., DARTS: Differentiable Neural Architecture Search, 2019

Search Space → **Search Method** → **Evaluation Method**

Search Space

- Continuous & Discrete
- Cell Block & Meta-Architecture

Search Method

- Random Search
- Evolutionary Search
- Bayesian Optimization
- Gradient-Based Optimization
- Reinforcement Learning

Evaluation Method

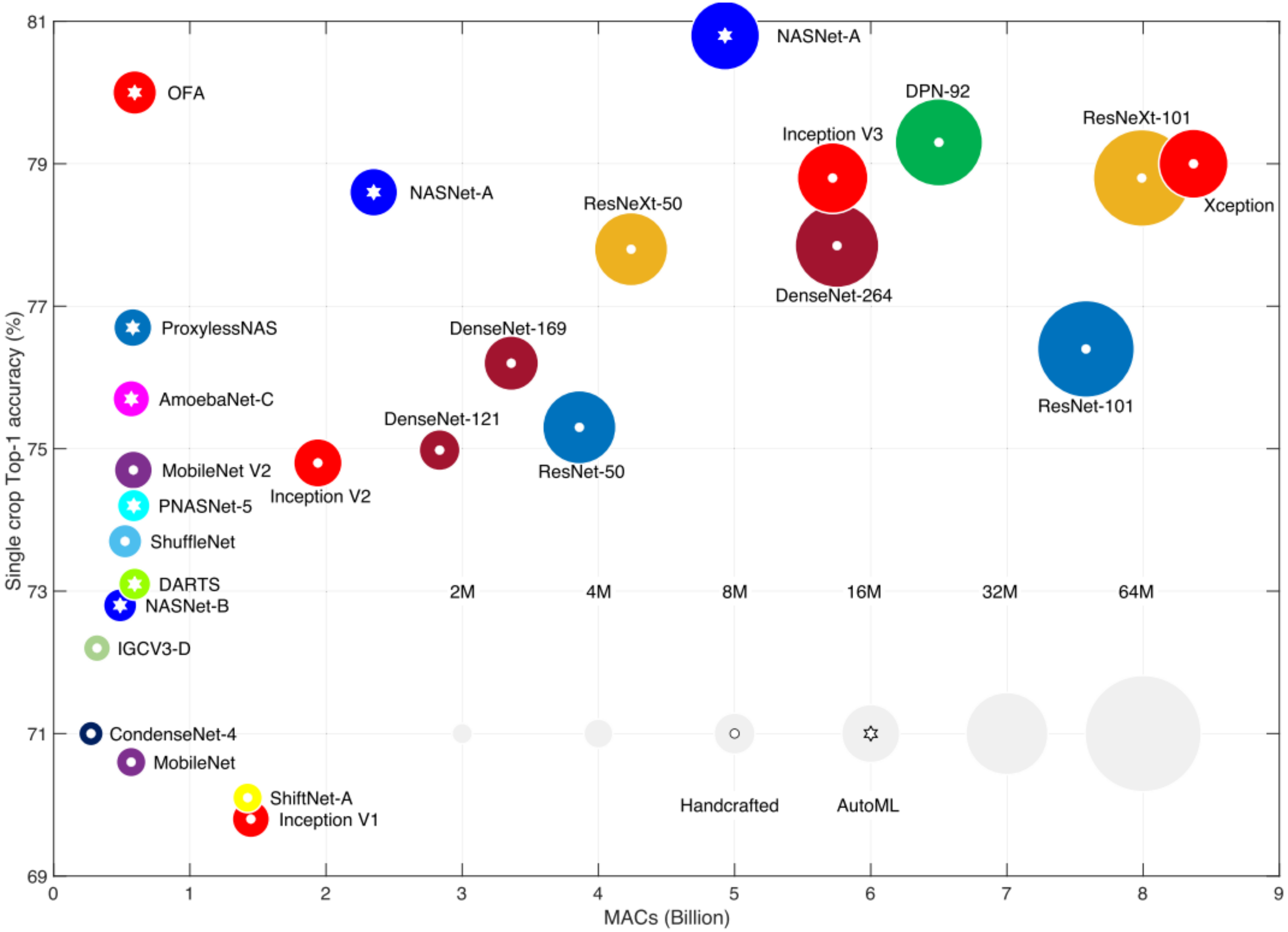
Cheap

- Weight-Sharing
- Hypernetworks
- Network Morphisms
- Partial Training
- Full Training

Costly

AmoebaNet

Real et al., *Regularized Evolution for Image Classifier Architecture Search*, 2018

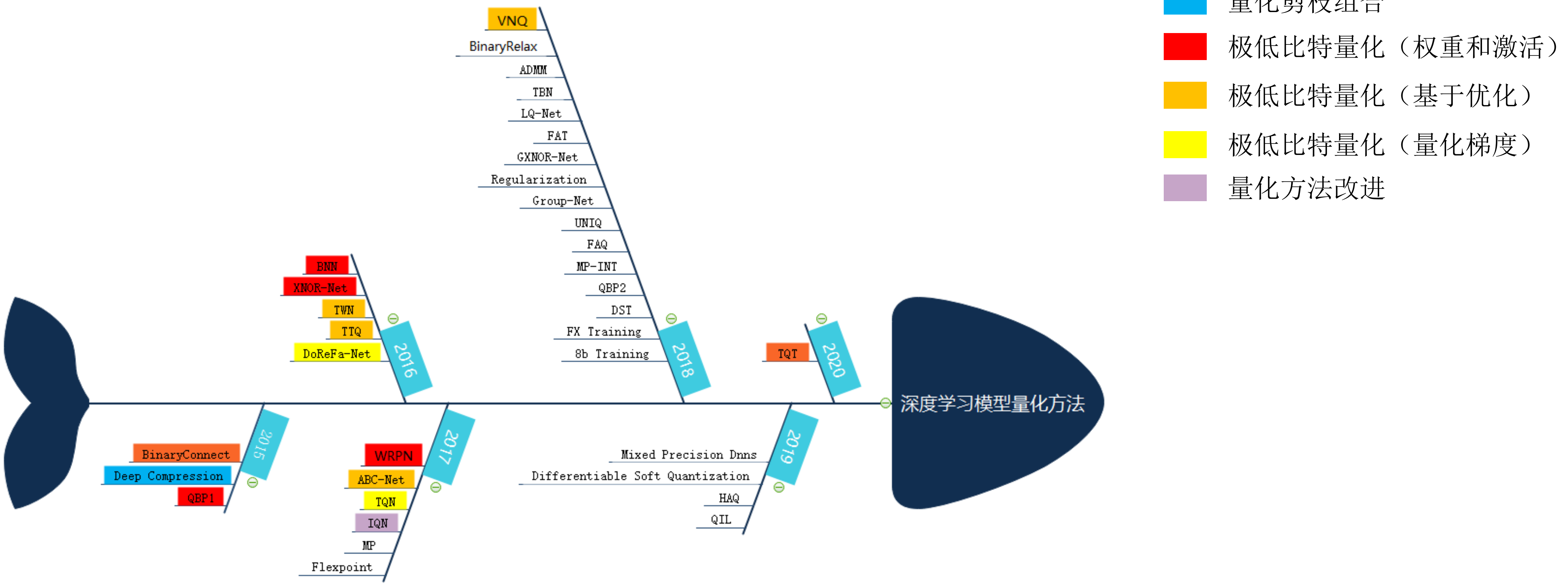


各模型在ImageNet上精度和复杂度

结论：NAS相关方法往往可以在一个搜索空间中找到更好的模型结构，常用于模型小型化和硬件亲和。

量化Overview

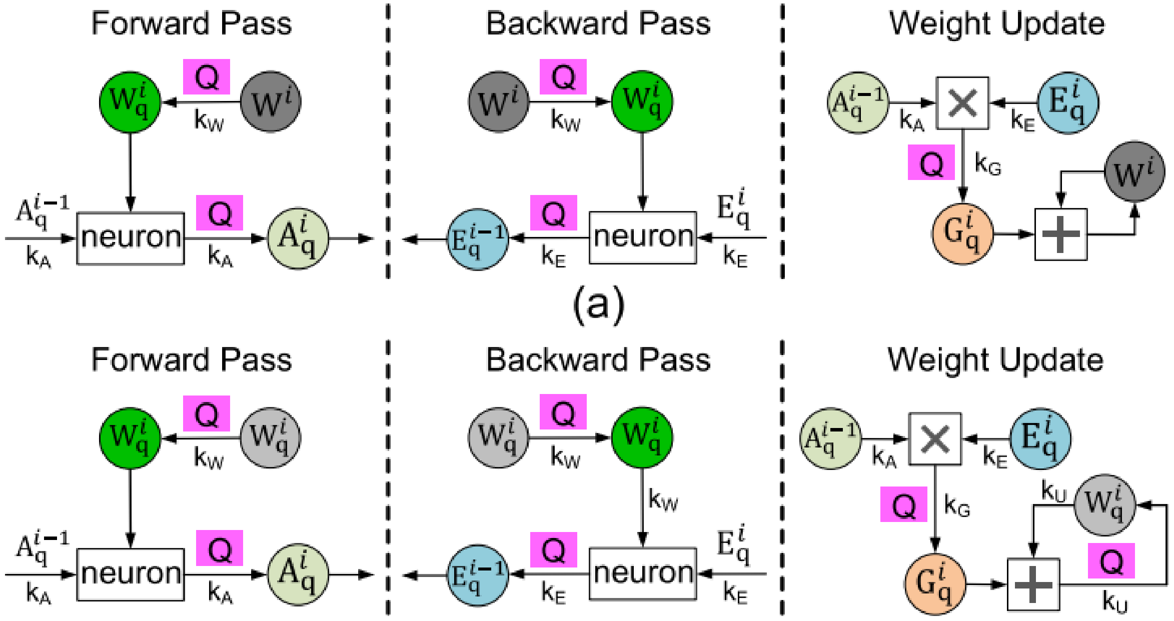
目的：
- 减少模型大小
- 特定芯片上的低比特计算更快速
- 训练、推理加速



Different Views of Quantization

Quantization Data Object

W: Weight
A: Activation
E: Activation Gradient
G: Parameters Gradient



Different Views of Quantization

Problem formulation

$$Q(x) = \Delta \cdot \mathrm{round}\left(\frac{x}{\Delta}\right)$$

v.s.

$$\min_{Q} \|\boldsymbol{X} - Q(\boldsymbol{X})\|_2^2, \quad \text{s.t. } Q_i \in X_Q \quad \text{for all } i$$

$$\Delta = c/(2^{bits} - 1)$$

$$z = min_{qtarget} - round(\frac{min_{val}}{\Lambda})$$

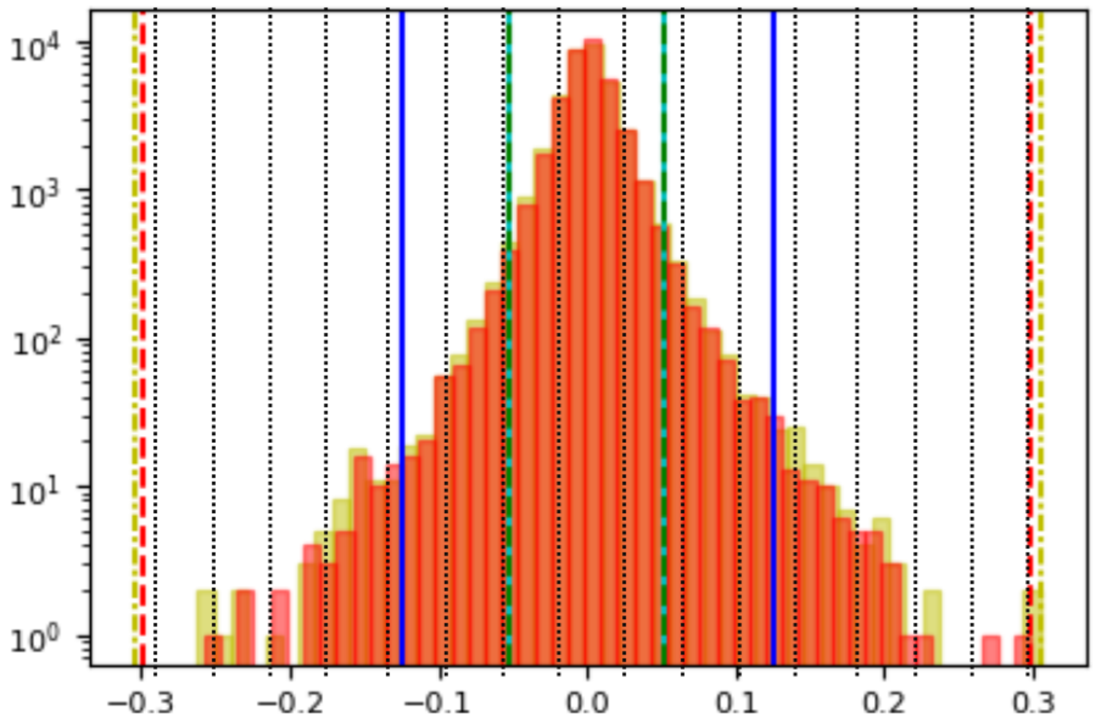
$$c = max_{val} - min_{val}$$

不同层有不同的bit位数

不同层有不同的c

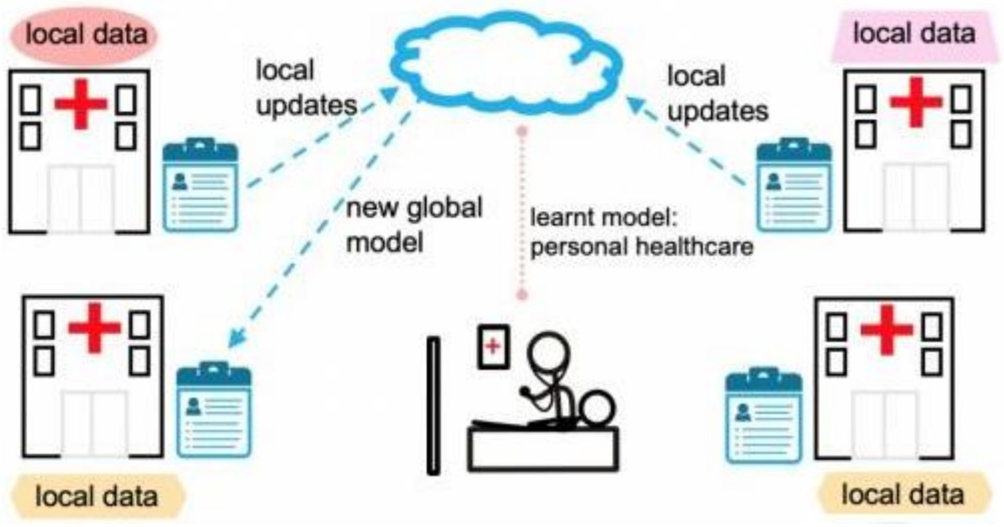
Different Views of Quantization

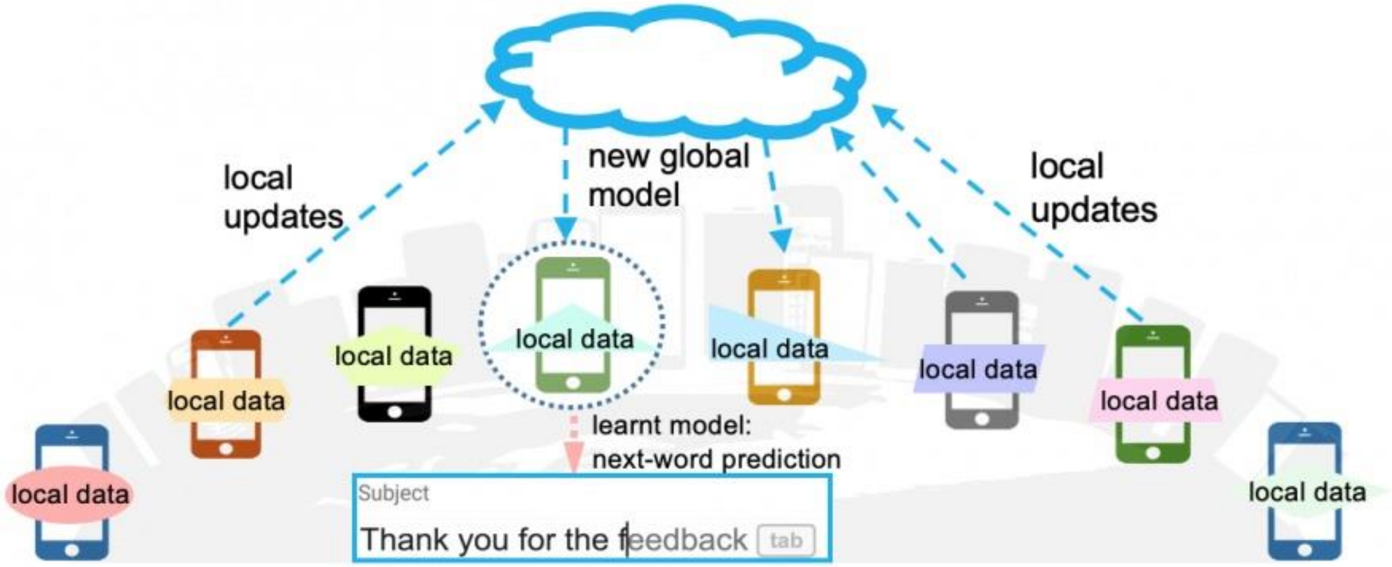
Dynamic Range/Threshold Calculation



量化方法的应用

- 推理加速
  - 2比特推理可以将乘法变成加法
  - 减少模型大小
- 训练加速
  - 对梯度做量化减少梯度的网络传输
  - 联邦学习





量化结论

使用启发式量化方法在小数据集上能够有比较好的表现，但是在大数据集上，往往优化的方法更有效；
对于CNN来说，不小于8bit的量化能够保证甚至提升精度，不大于4bit的量化会导致明显的精度下降；

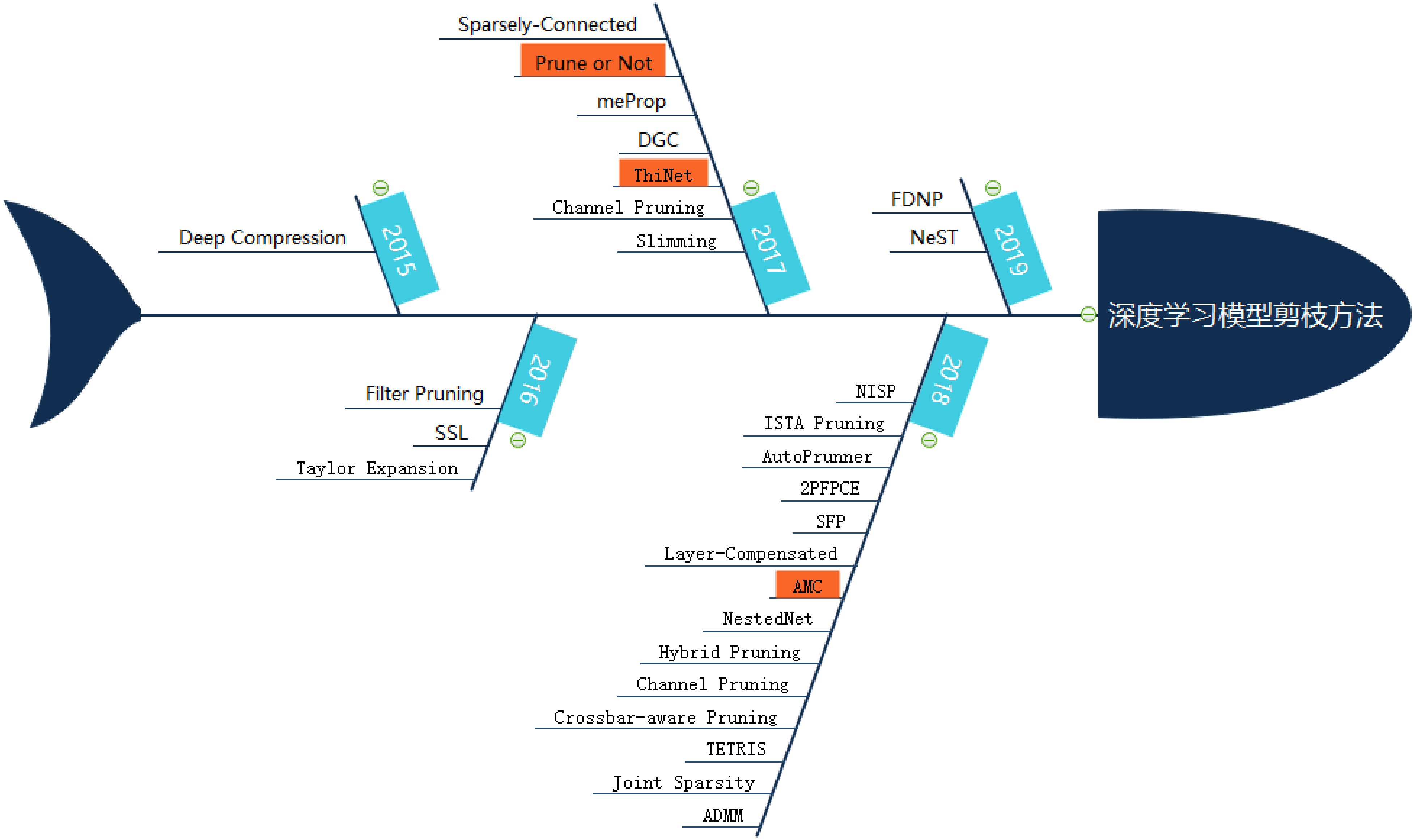
对于RNN的量化来说，很少有能成功量化到低bit的工作；

在权重、激活以及权重梯度上的量化会容易一些，在激活的梯度以及权重的更新操作上做量化会导致模型恶化；所以量化梯度后进行分布式训练，减小带宽是可能的；

比较冗余的网络结构能够有比较好的量化效果，例如VGG、AlexNet，也是很多论文的目标；

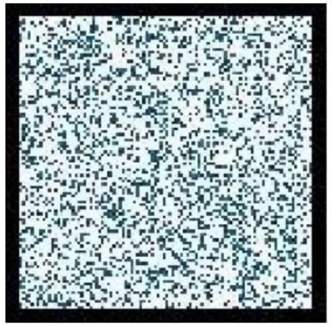
Reference	Sensitivity	Configuration and Accuracy
TernGrad (2017) [167]/ADMM (2018) [165]/TTQ (2016) [162]	CNN: G≤W	ImageNet-AlexNet, **G(ternary)**: top1-↑0.28% [167]; **W(ternary)**: top1-↓1.8% [165]; **W(ternary)**: top1-↑0.3% [162]
WRPN (2017) [170]/HWGQ (2017) [160]	CNN: W<A	ImageNet-AlexNet, **W(2b)**: top1-↑0.3%, **A(2b)**: top1-↓4.5% [170]; **W(binary)**: top1-↓0.3%, **A(ternary)**: top1-↓6.2% [160]
		ImageNet, ResNet18, **W(binary)**: top1-↓5%, **A(ternary)**: top1-↓28.8% [160]; VGG-Variant, **W(binary)**: top1-↓3.1%, **A(ternary)**: top1-↓20.3% [160]
DoReFa-Net (2016) [147]	CNN: A<E	**W(2b)** on SVHN, **A(2b)/E(4b)**: ↑0%; **A(4b)/E(2b)**: ↓16%
WAGE (2018) [182]	CNN: E<U	**W(ternary)/A(8b)/E(8b)** on CIFAR10-VGG8, **G(8b)/U(8b)**: ↓1.07%; **G(4b)/U(4b)**: ↓22.51%
		W(ternary)/A(8b) on ImageNet-AlexNet, **E(8b)/G(12b)/U(12b)**: top5-↓7.59%; **E(12b)/G(8b)/U(8b)**: top5-↓8.77%
	CNN: BN matters	**W(ternary)/A(8b)**, **BN**: top5-↓1.38%; **Linear Scaling**: top5-↓4.85%
Neuron Increase (2017) [190]	RNN: A<W	PTB-LSTM300×1, **A(4b)**: ↑0.5% PPW; **W(4b)**: ↑5.6 PPW; **A(2b)**: ↑2.7% PPW; **W(2b)**: ↑32.4 PPW
		PTB-LSTM450/1000×1, **W(4b)/A(2b)**: 111.7/113.1 PPW; **W(2b)/A(4b)**: 130.6/128.4 PPW

剪枝（模型稀疏化）Overview

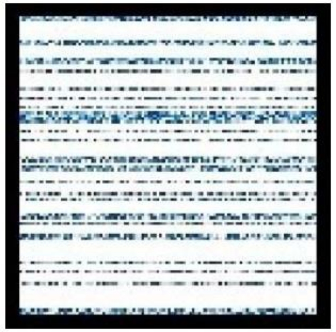


Different Views of Pruning

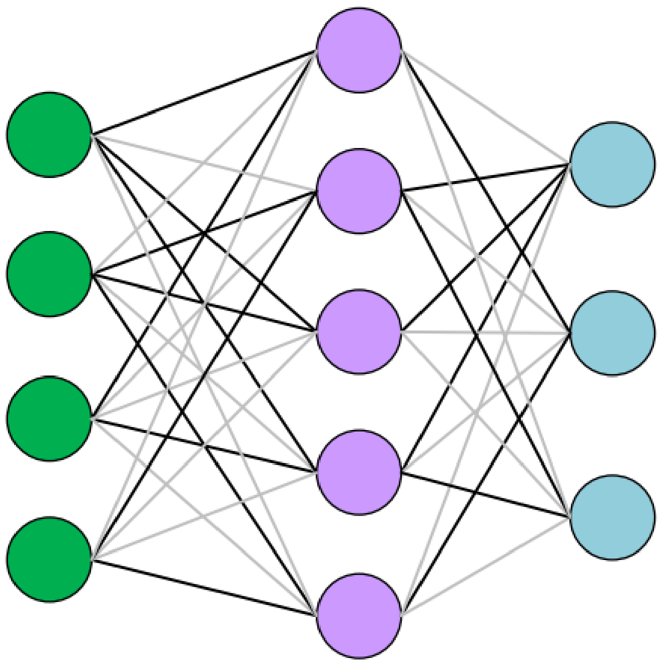
Pruning Data Object



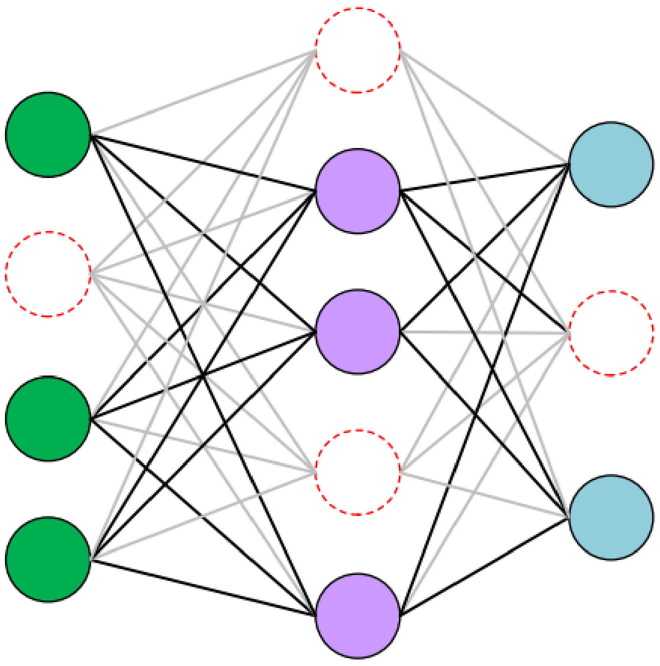
ElementWise Sparsity



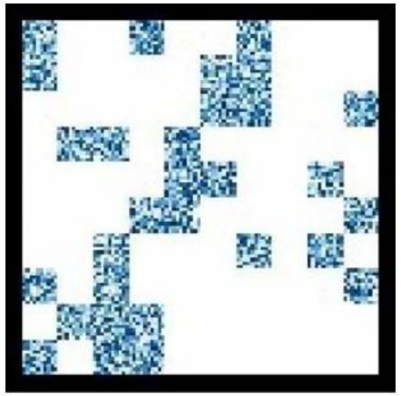
VectorWise Sparsity



Weight Pruning



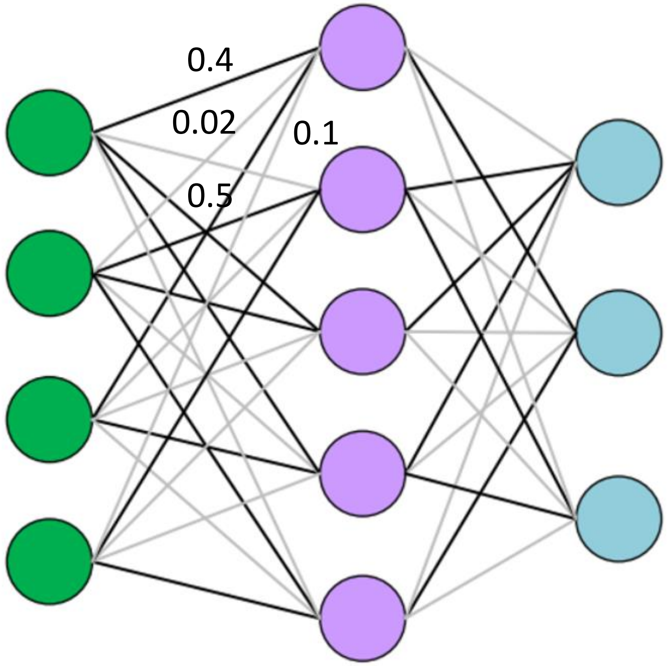
Neuron Pruning



Block Wise Sparsity

Different Views of Pruning

Pruning Methods



Heuristic Choice of Smallest Ln-Norm

$$\min_{\boldsymbol{W}} L = L_0(\boldsymbol{W}) + \lambda \sum_{g=1}^{G} \|\boldsymbol{W}^{(g)}\|_2$$

Add Regularization to Weight

$$\min_{S} \sum_{i=1}^{m} \left(\hat{y}_i - \sum_{j \in S} \hat{x}_{ij} \right)^2, \quad S \subset \{1, 2, \ldots, C\}$$

$$y = \sum_{c=1}^{C} \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} w_{ck_1k_2} x_{ck_1k_2} + b$$

The Least Effective Weight

$$\min_{\boldsymbol{W},\, \boldsymbol{\gamma}} L = L_0(\boldsymbol{W}) + \lambda \|\boldsymbol{\gamma}\|_1$$

BN's Gamma Help

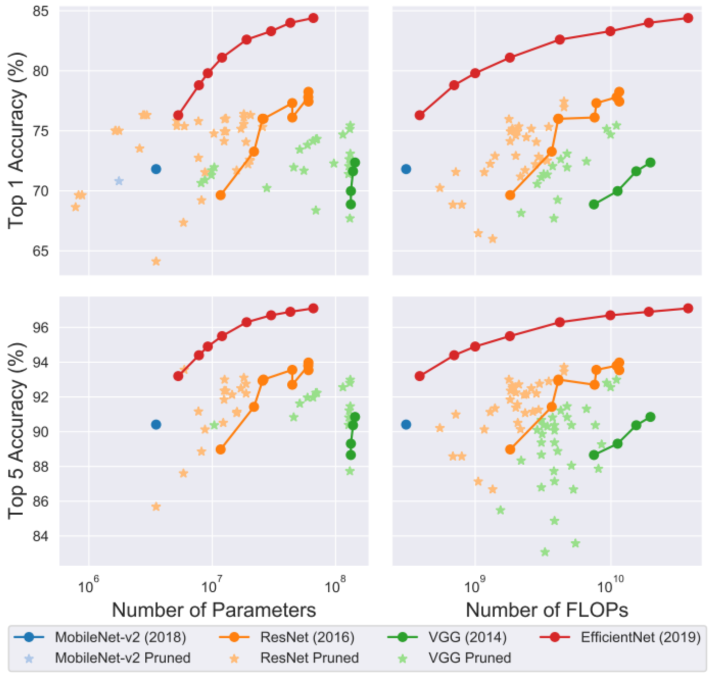
剪枝（稀疏化）方法的应用

推理加速

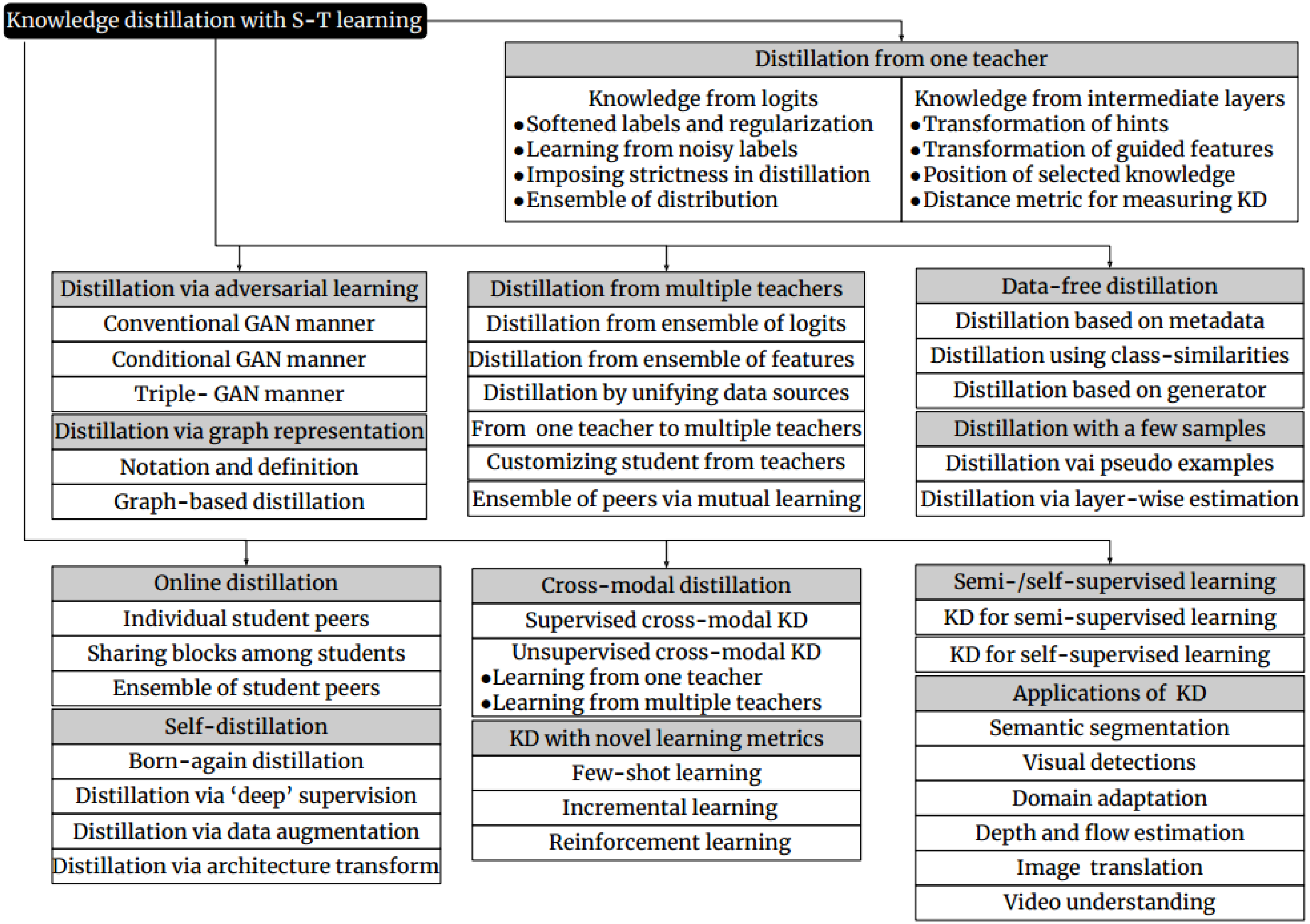
与量化方法结合进一步提升压缩率

剪枝结论

- 模型剪枝是有效的，在图像分类任务上，往往能减少80%+的参数量保证精度不变，但是往往不如一个新的模型簇；
- 不同的剪枝方式影响到不同的稀疏化模式， e.g. blocked的稀疏化对硬件是友好的；
- 迭代式的剪枝方式能够在保证精度的情况下极大提升性能，但是过程冗长；
- 对于不同的模型，事先得到Lottery Ticket，剪枝模板可能可以复用；
- 当前学术界/业界的文章主要focus在分类任务上，对于其他任务的精度有待考证；
- 对于比较小的数据集，启发式地剪枝即可，比较大的数据集需要基于优化的方法；
- 使用RL方法可以在众多剪枝参数中找到比较好的参数，同时可以用到不同的信息作为反馈；



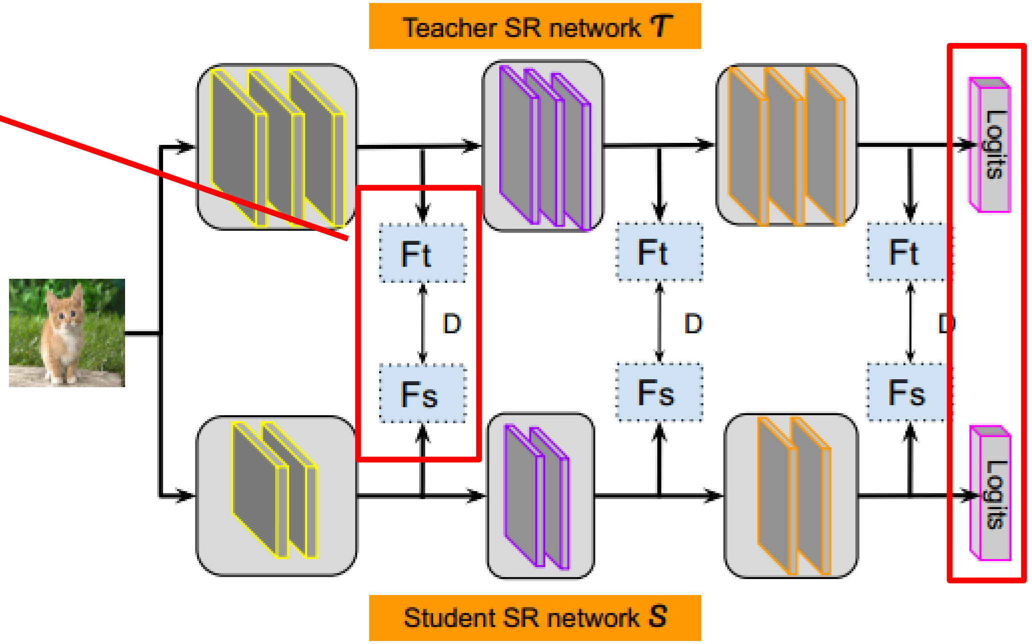
知识蒸馏（Knowledge Distillation） Overview



Different Views of KD

Position

Intermedia Feature Map



Logits

Different Views of KD

Losses

$$p_i = \frac{\exp(\frac{z_i}{T})}{\sum_j \exp(\frac{z_j}{T})}$$

Temperature Softmax/SoftLabel

$$L_n = ||x_s - x_t||_n$$

L1, L2 …

$$cosine = \frac{x_s \cdot x_t}{||x_s||||x_t||}$$

Cosine similarity

$$\min_G \max_D J(G, D) = \mathbb{E}_{x \sim p(x)}[log(D(x))] + \mathbb{E}_{z \sim p(z)}[log(1 - D(G(z)))]$$

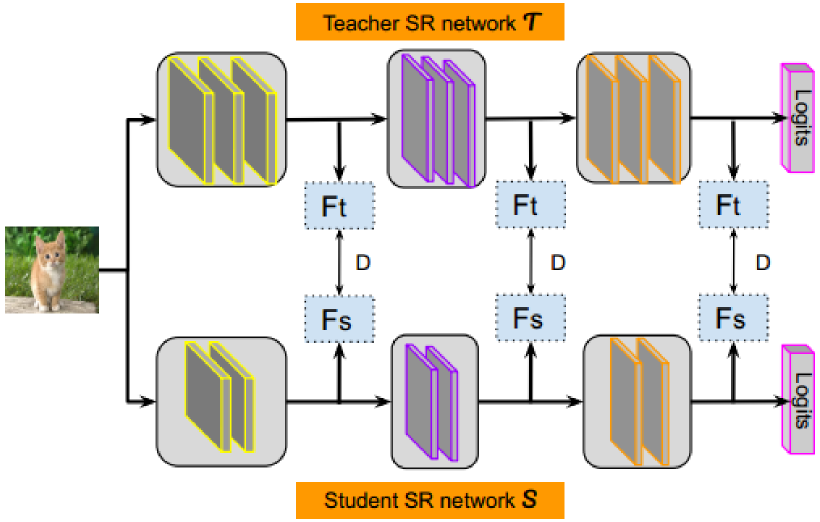
GAN Loss

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

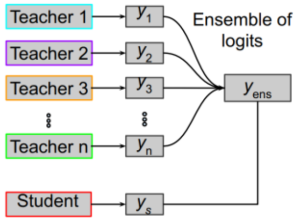
KL Divergence

Different Views of KD

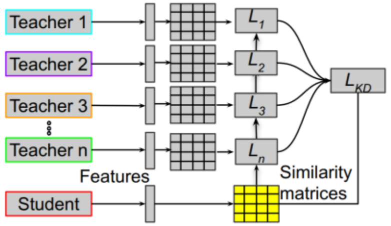
Who is Teacher



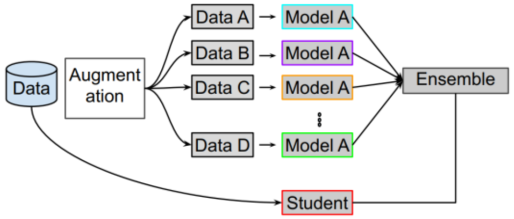
Single Teacher



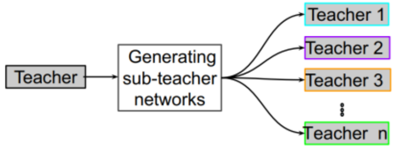
(a) ensemble of logits

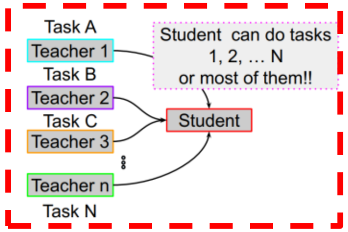


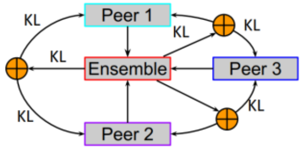
(b) ensemble of features



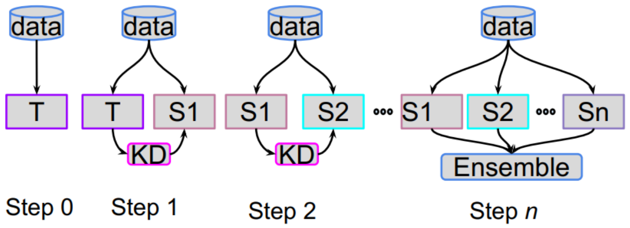
(c) unifying data sources



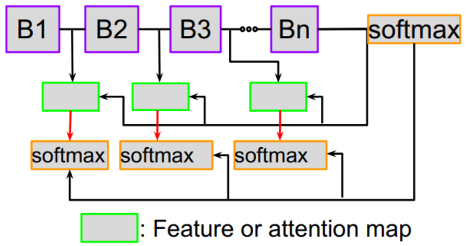




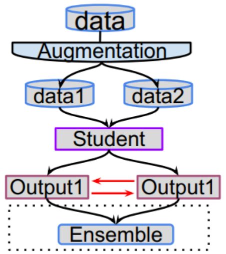
Multiple Teachers



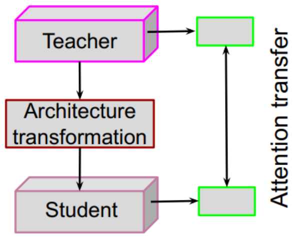
(a) Born-again KD



(b) KD via 'deep' supervision



(c) data augmented KD

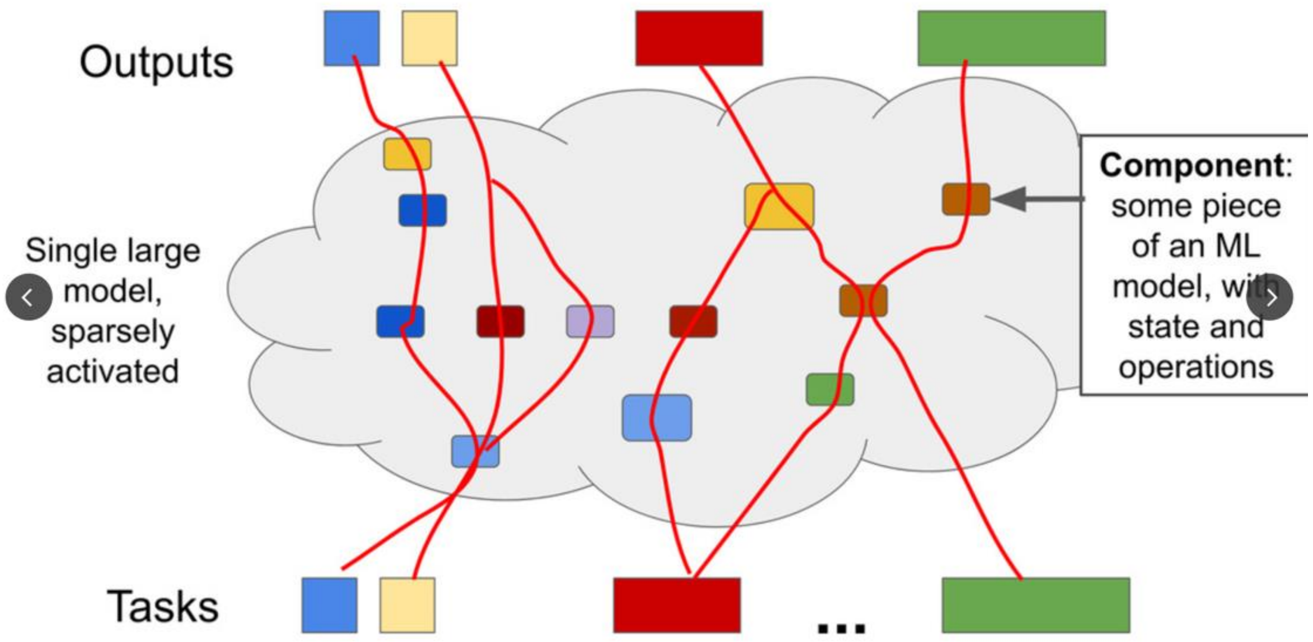


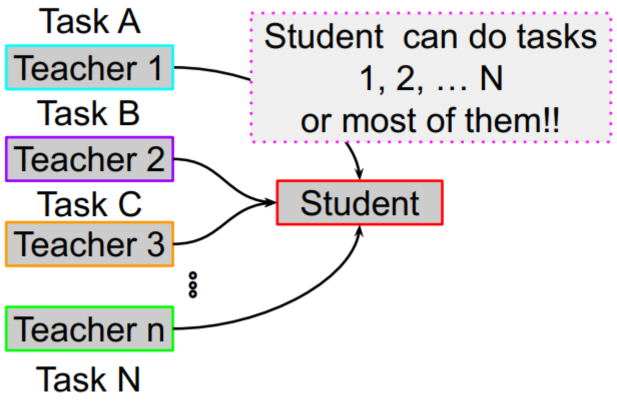
(d) KD with architecture transform

Self-Teaching

蒸馏方法的应用

- 提升小模型的精度；
- 综合利用不同数据域的知识；
- 域迁移；
- 跨膜态学习





知识蒸馏总结

- 知识蒸馏作为模型小型化的方法，往往和其他的方法结合一起使用；
- 虽然知识蒸馏不要求Teacher和Student结构上的相似性，但往往结构越相似的模型，蒸馏效果越好；
- 知识蒸馏可以分为数据蒸馏和特征蒸馏，数据蒸馏在一些弱监督领域应用广泛；
  - 数据蒸馏：使用教师模型给没标签的数据打上标签；
  - 特征蒸馏：使用教师模型的特征层/logits进行蒸馏；
- 知识蒸馏往往用在目标识别领域，对于其他领域也渐渐有所探索，但是有很大空间。

总结

- 量化（当前框架）-> 操作空间
  - Common
    - 所有层的量化比特一致 -> 自适应地根据不同层的重要程度调整量化比特数
    - 都采用Uniform Quantize -> 自适应地根据不同层的分布选择量化的分布
  - Post Training Quantization
    - 不同统计量化Threshold的方法
  - Quantization Aware Training
    - Dynamic Range根据统计量得到 -> 通过损失函数压缩模型权重分布，更容易量化
    - 纯训练 -> 加入原始模型，引入蒸馏

总结

- 剪枝（当前框架）-> 操作空间
  - 模型稀疏率在全局或者每层上调整 -> 自适应根据每层的重要性调整稀疏率
  - 所有模型都需要迭代地进行稀疏化 -> 根据模型结构对应的稀疏化结构进行稀疏化
  - 基于启发式的剪枝策略 -> 在损失上进行模型稀疏化的约束
  - 冗长的迭代剪枝策略 -> One-shot的方法的探索
  - 纯训练 -> 剪枝+蒸馏训练
- 蒸馏（当前框架）-> 操作空间
  - 无 -> 常见蒸馏范式的实现

总结

端到端模型推理/训练加速(高校课题/合作)

One-shot
量化剪枝

低比特模型训练、联邦学习

提升精度+硬件感知

优化的剪枝、量化算法

自适应根据硬件调整量化、剪枝策略

基础能力组件

低比特算子
卷积、矩阵乘

稀疏计算算子
卷积、矩阵乘

PTQ相关量化策略实现

蒸馏相关逻辑

Thank you

www.huawei.com