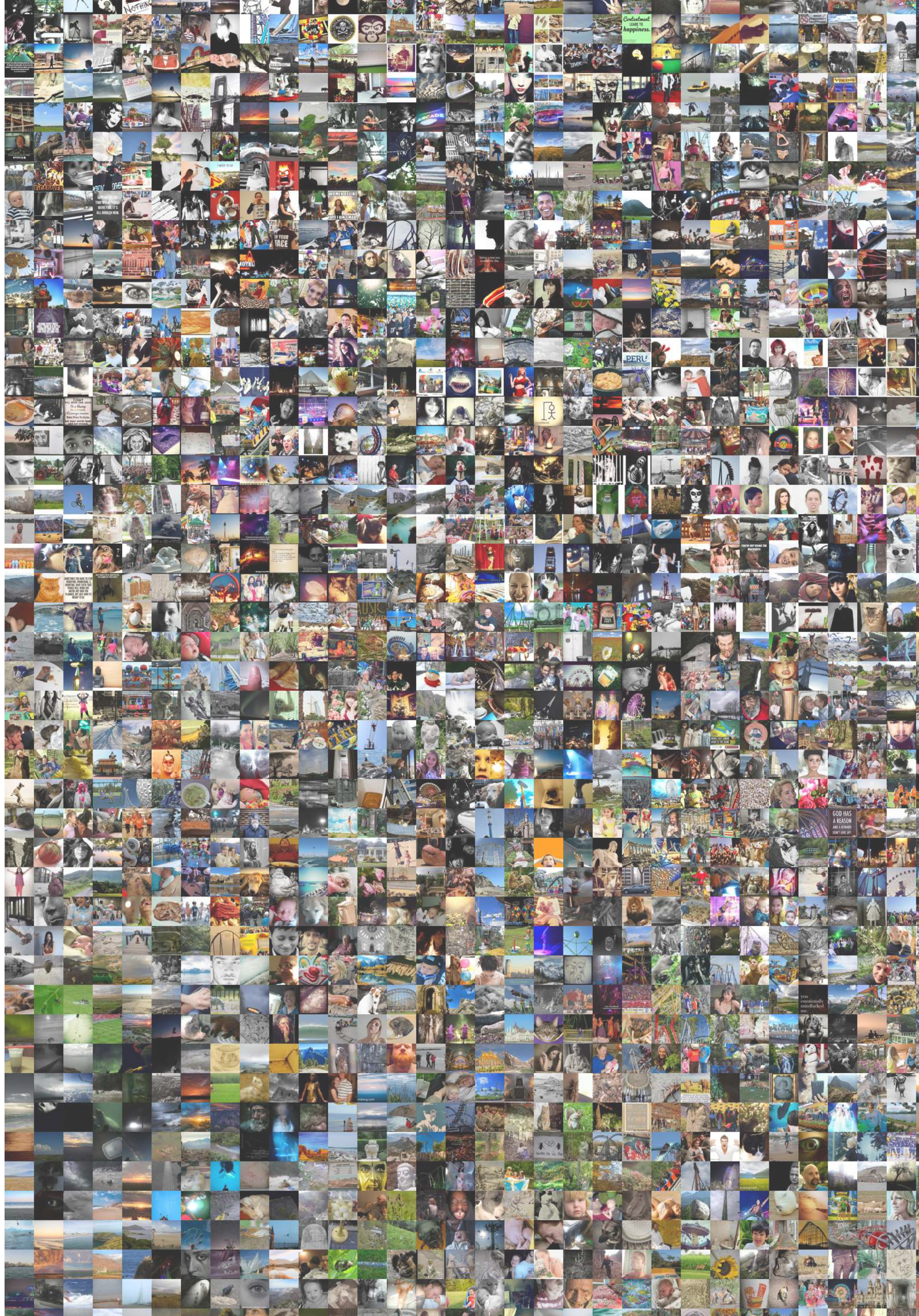
Neural Architecture Search

CS 4803 / 7643 Deep Learning

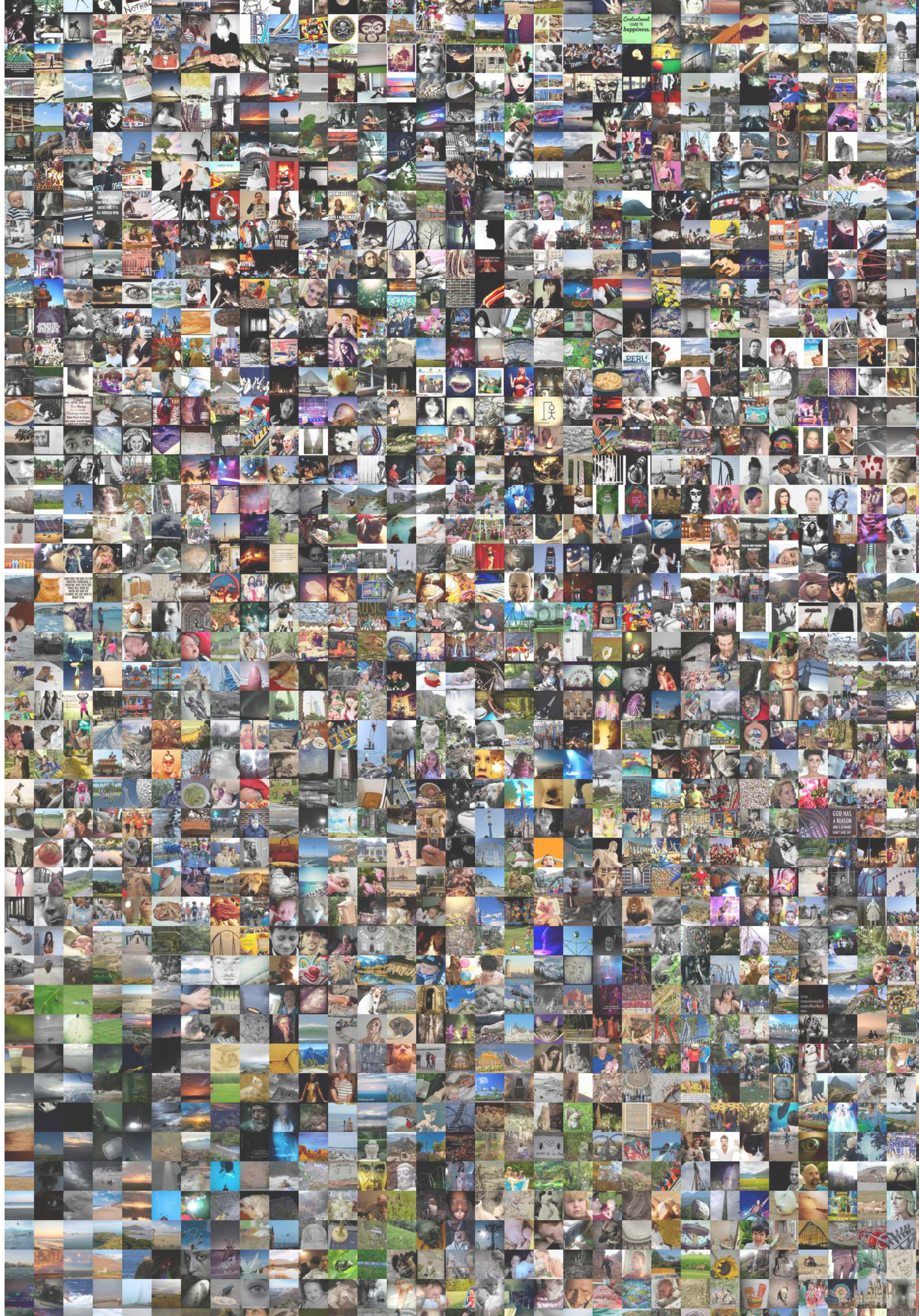
Erik Wijmans, 10/29/2020

Background



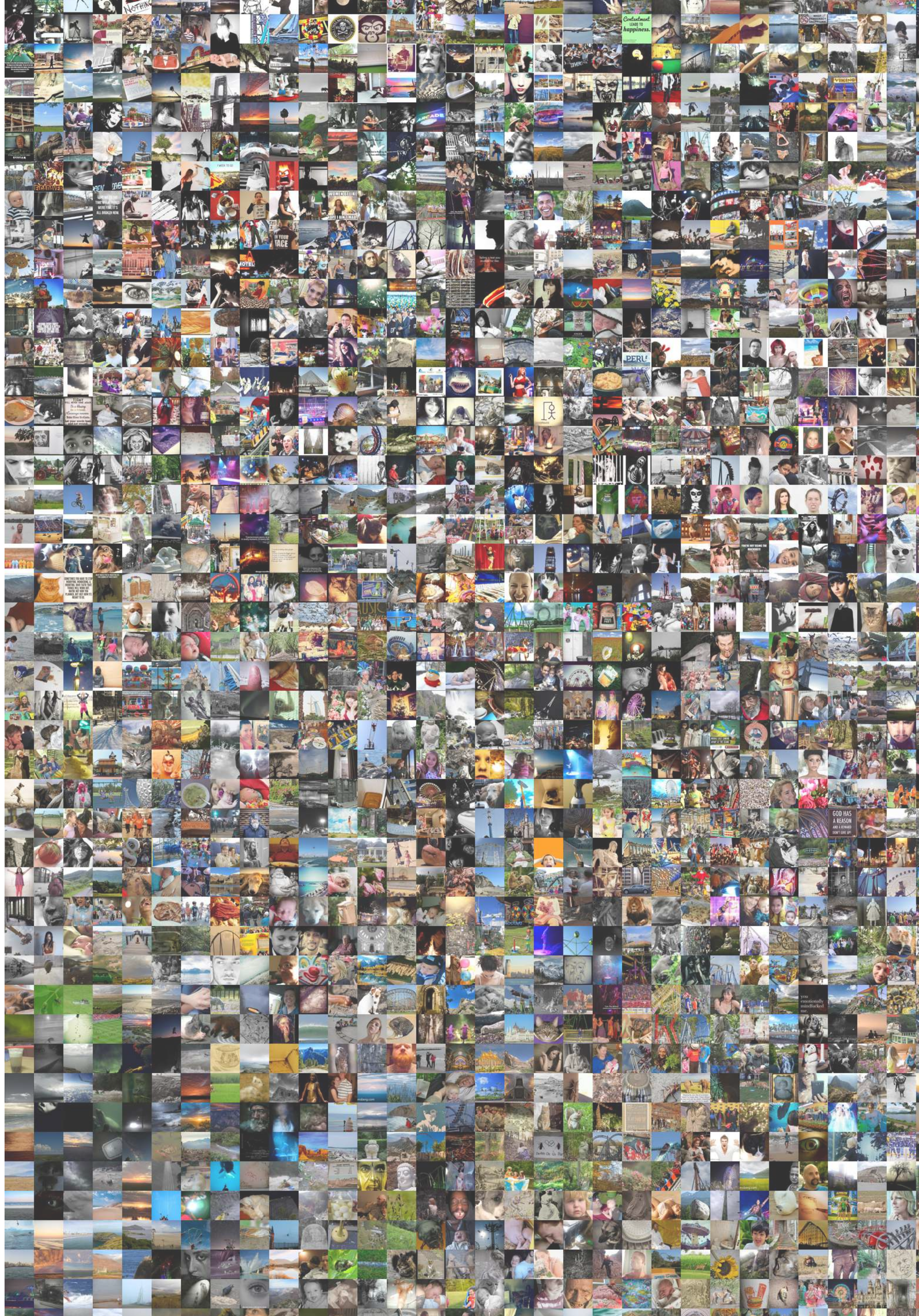
Background

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}\left(f\left(x;\theta \right), y \right) \right]$$

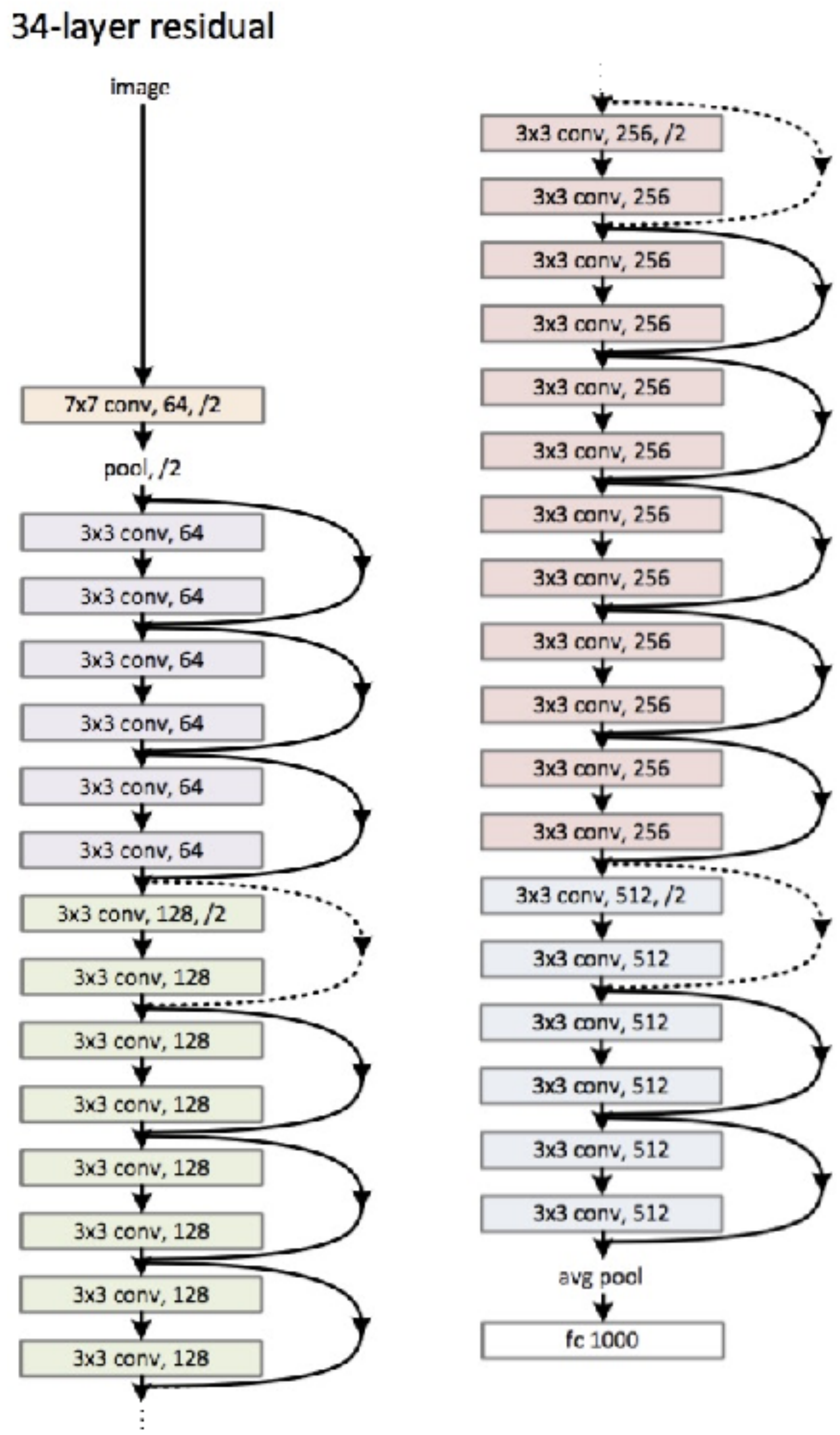


Background

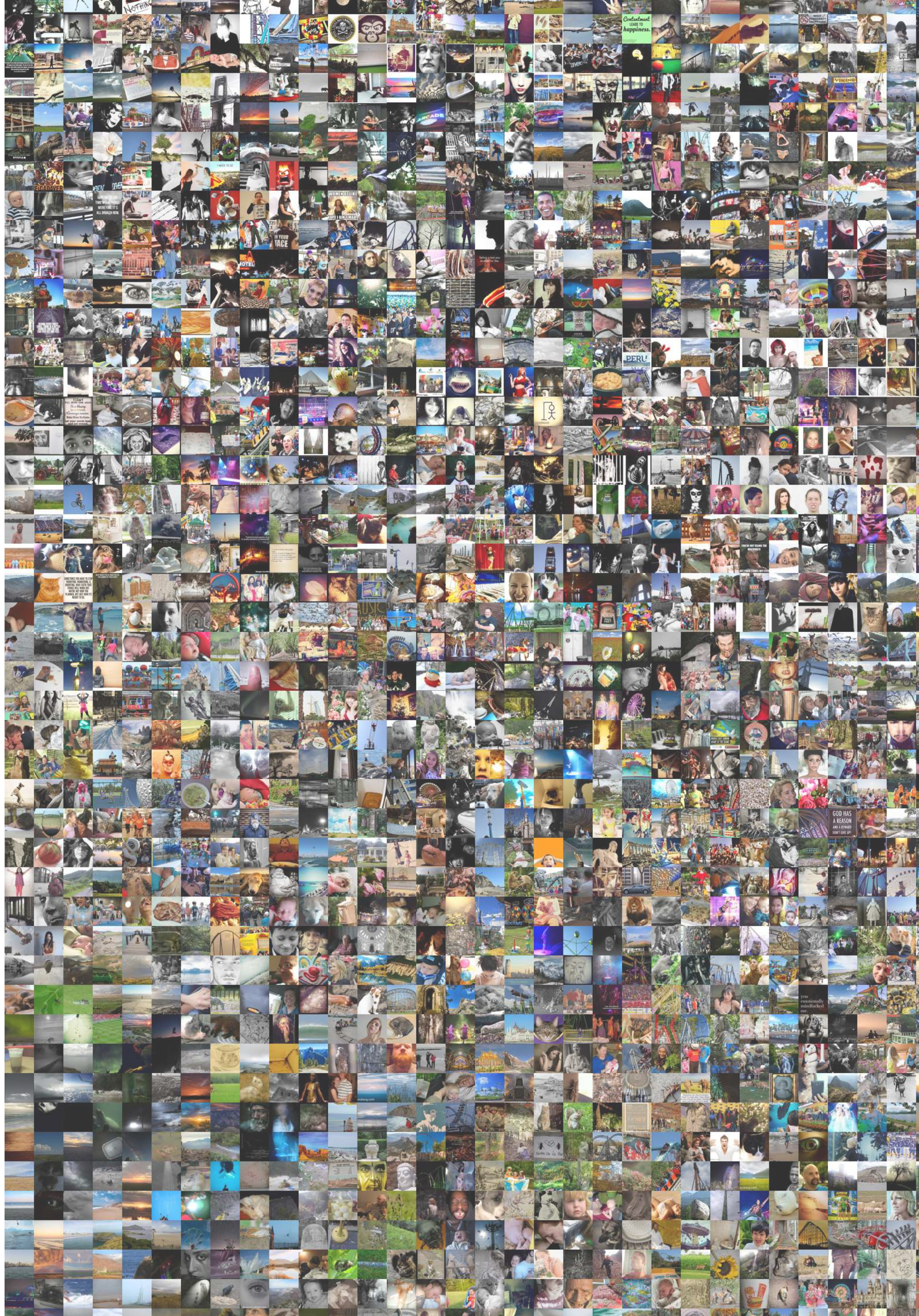
$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L} \left(f \left(x; \theta \right), y \right) \right]$$



Background

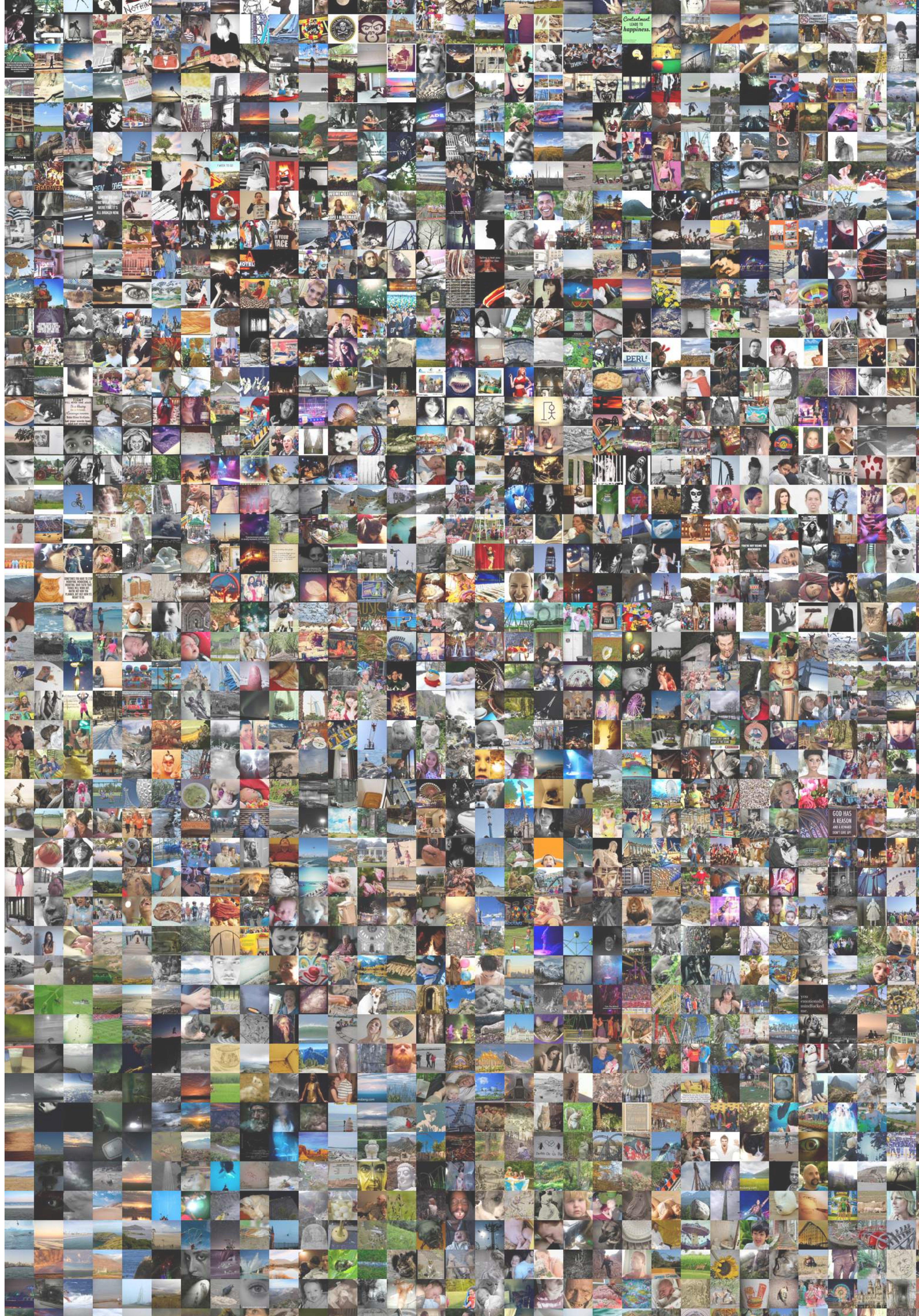


34-layer residual
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
avg pool
fc 1000



Background

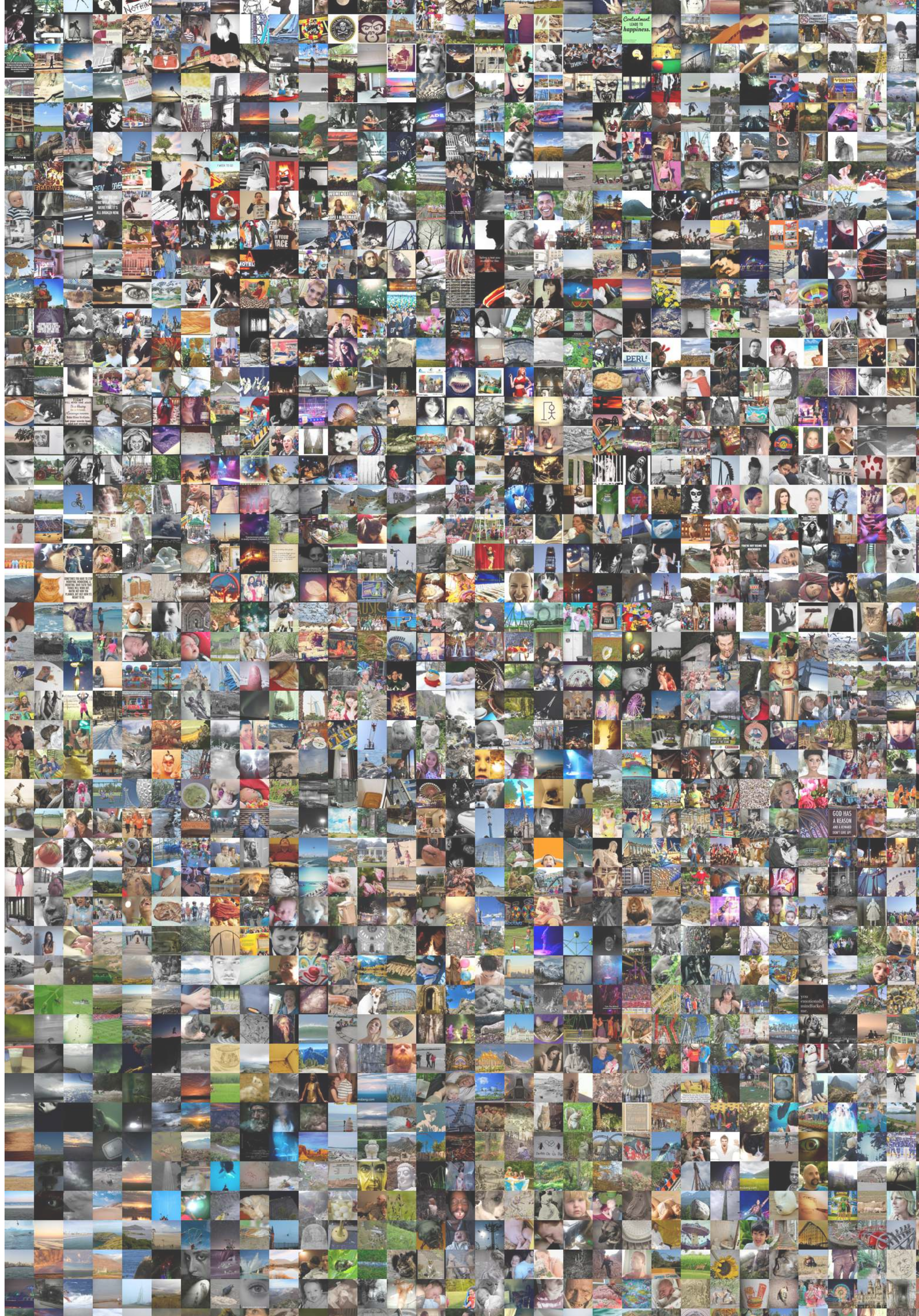
$$\min_{f \in \mathcal{F}} \min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L} \left(f \left(x; \theta \right), y \right) \right]$$



Background

$$\min_{f \in \mathcal{F}} \min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}\left(f\left(x;\theta\right), y \right) \right]$$

Set of networks



Neural Architecture Search

Neural Architecture Search

High Level Overview

Neural Architecture Search

High Level Overview

Search Space

Neural Architecture Search

High Level Overview

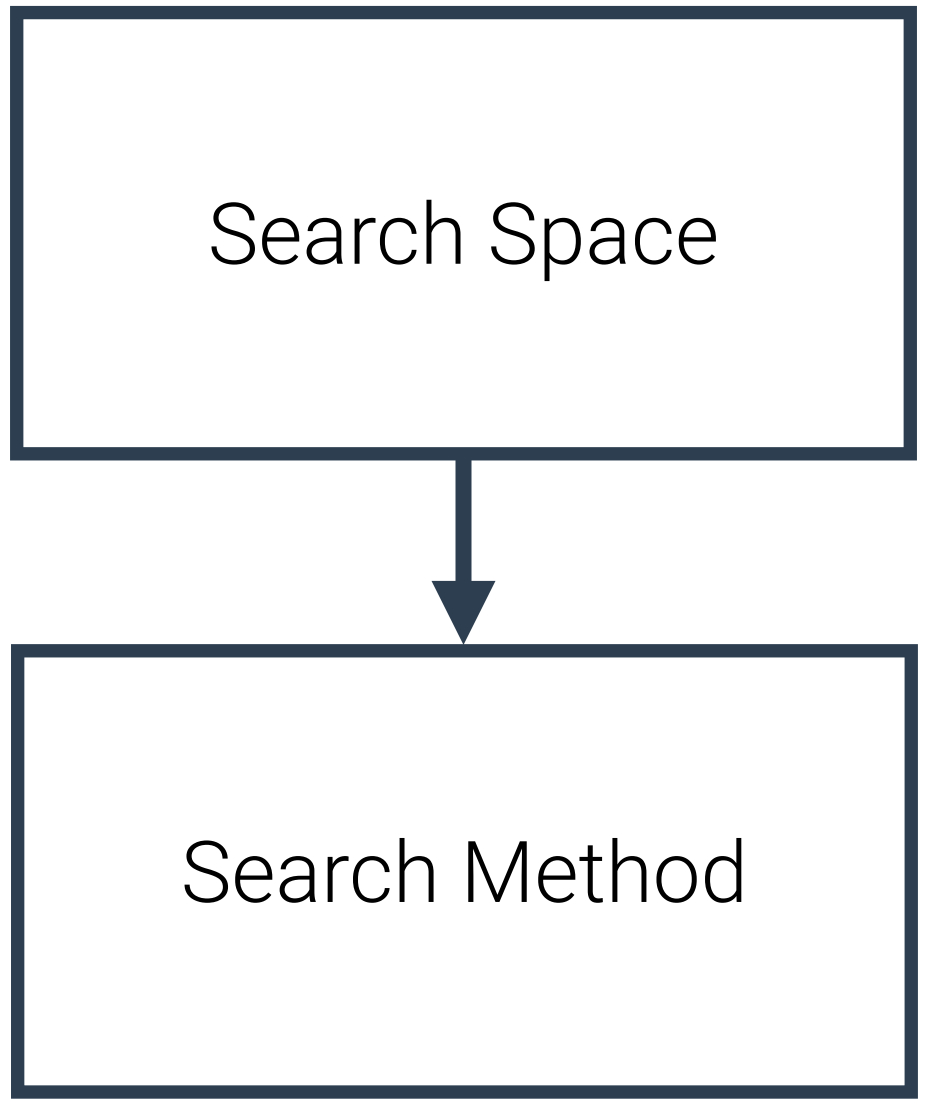
Search Space

$$\min_{f \in \mathcal{F}} \min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}\left(f\left(x;\theta\right), y\right)\right.$$

Set of networks

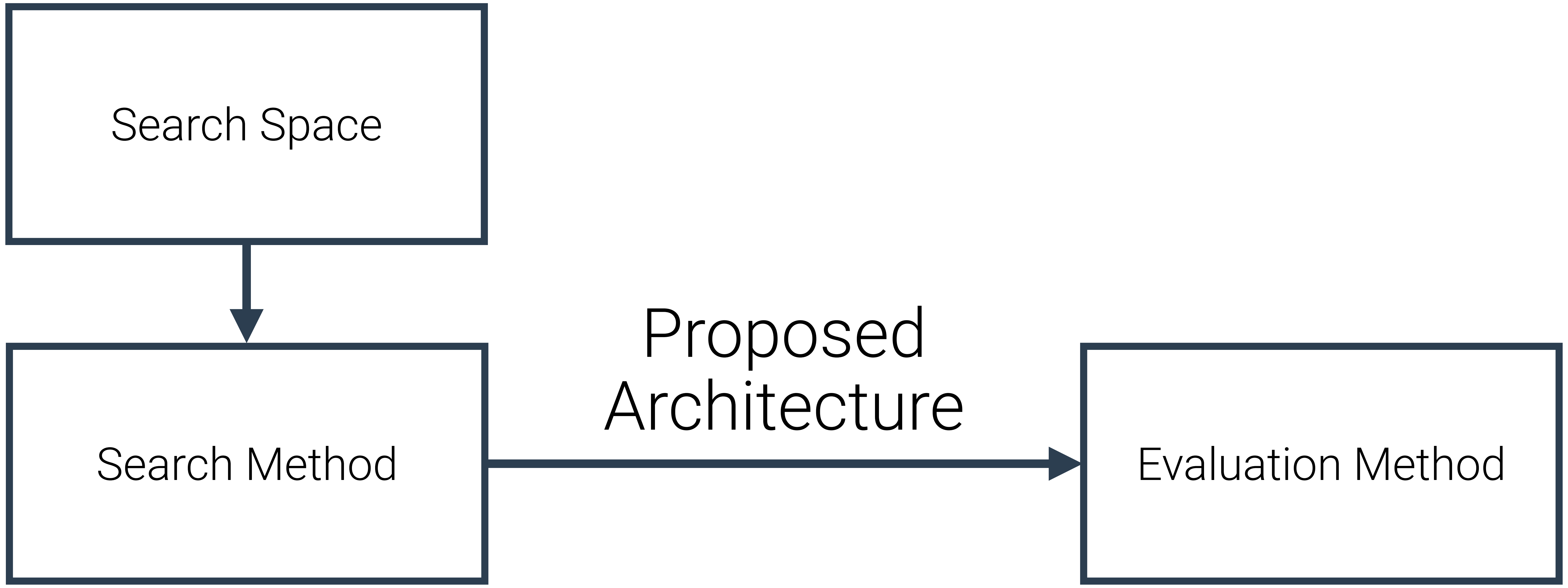
Neural Architecture Search

High Level Overview



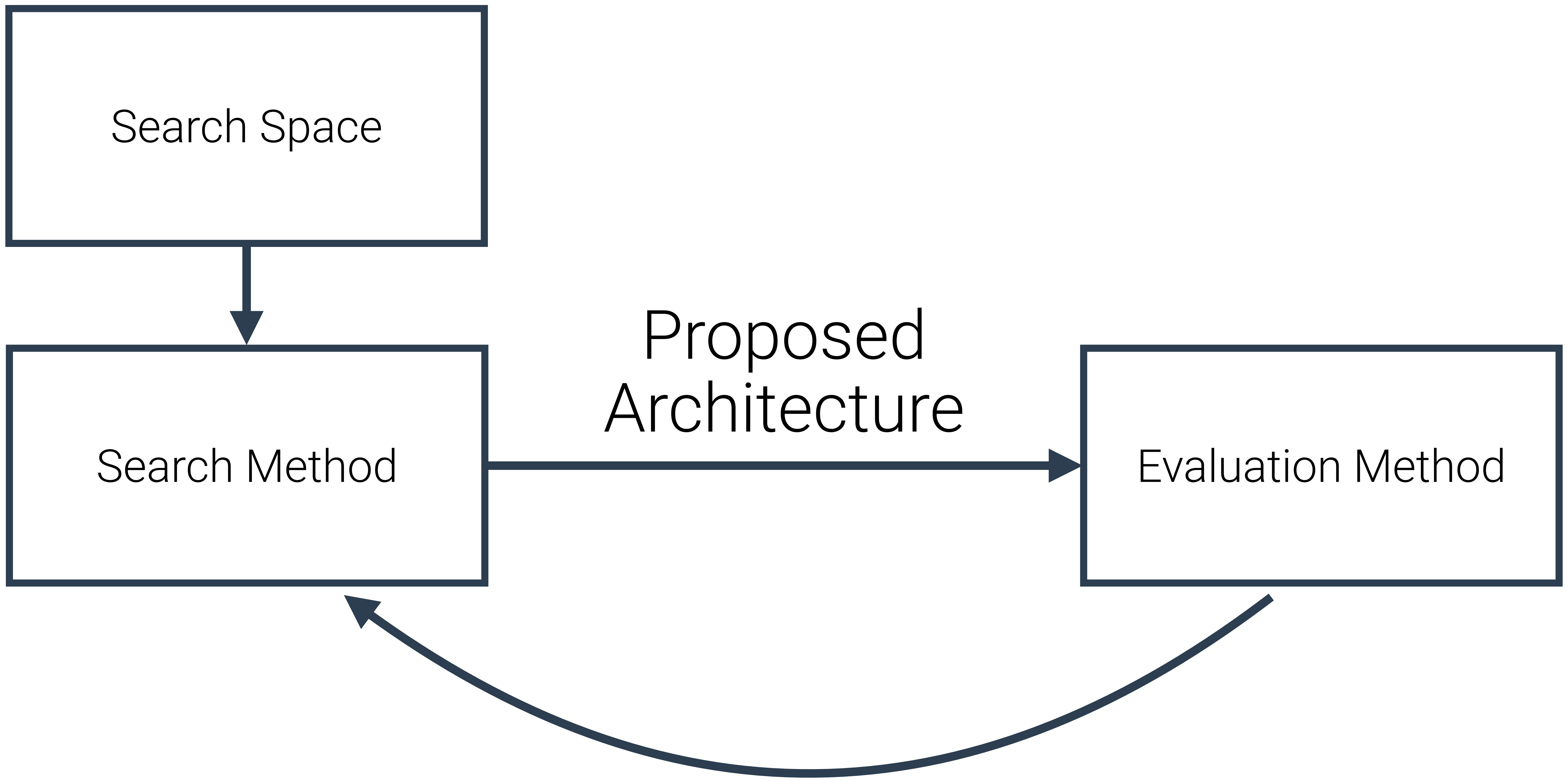
Neural Architecture Search

High Level Overview



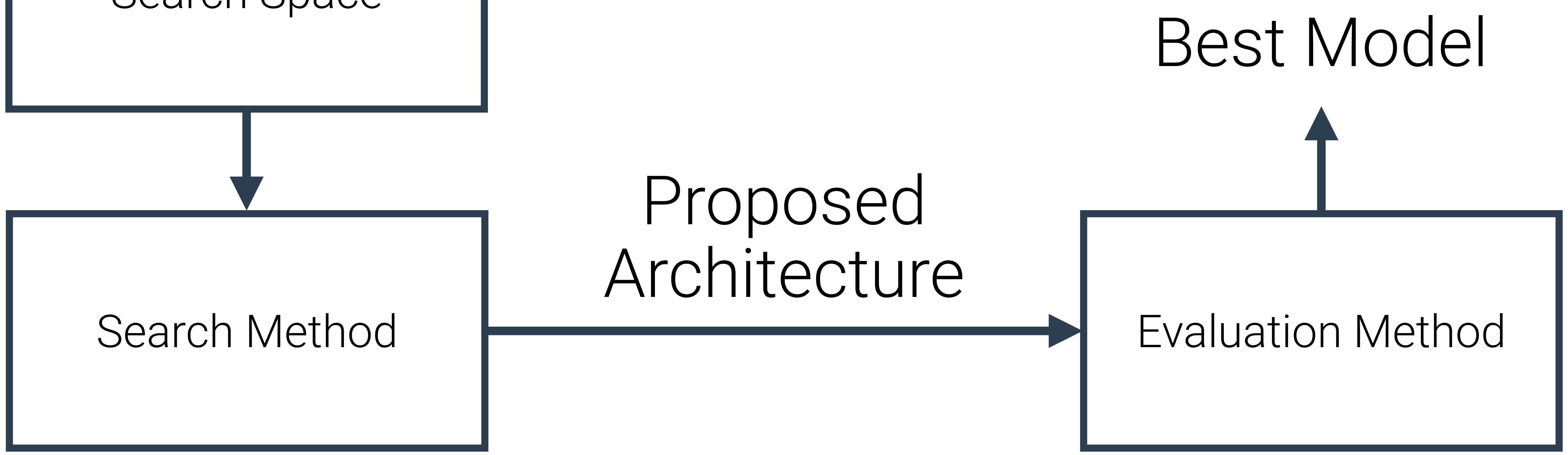
Neural Architecture Search

High Level Overview



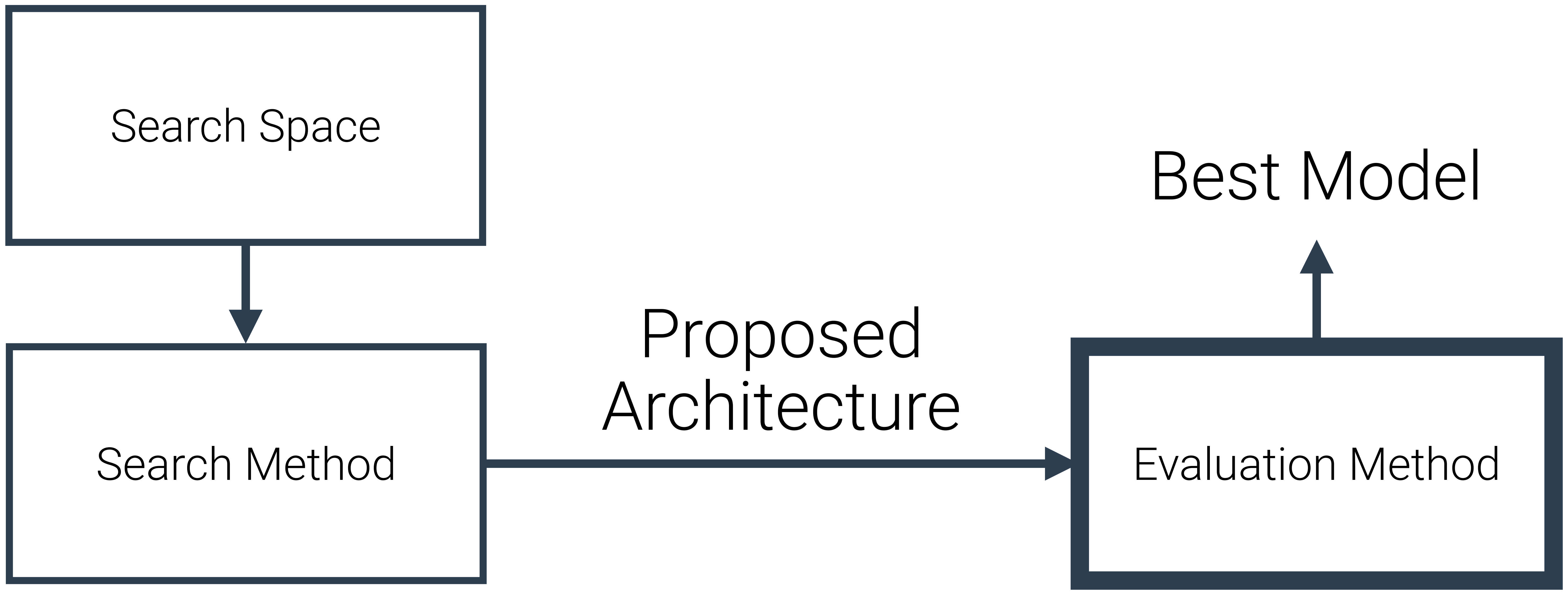
Neural Architecture Search

High Level Overview



Neural Architecture Search

High Level Overview



Neural Architecture Search
Evaluation Method

Neural Architecture Search
Evaluation Method

- Generally, this is performance on held-out data.

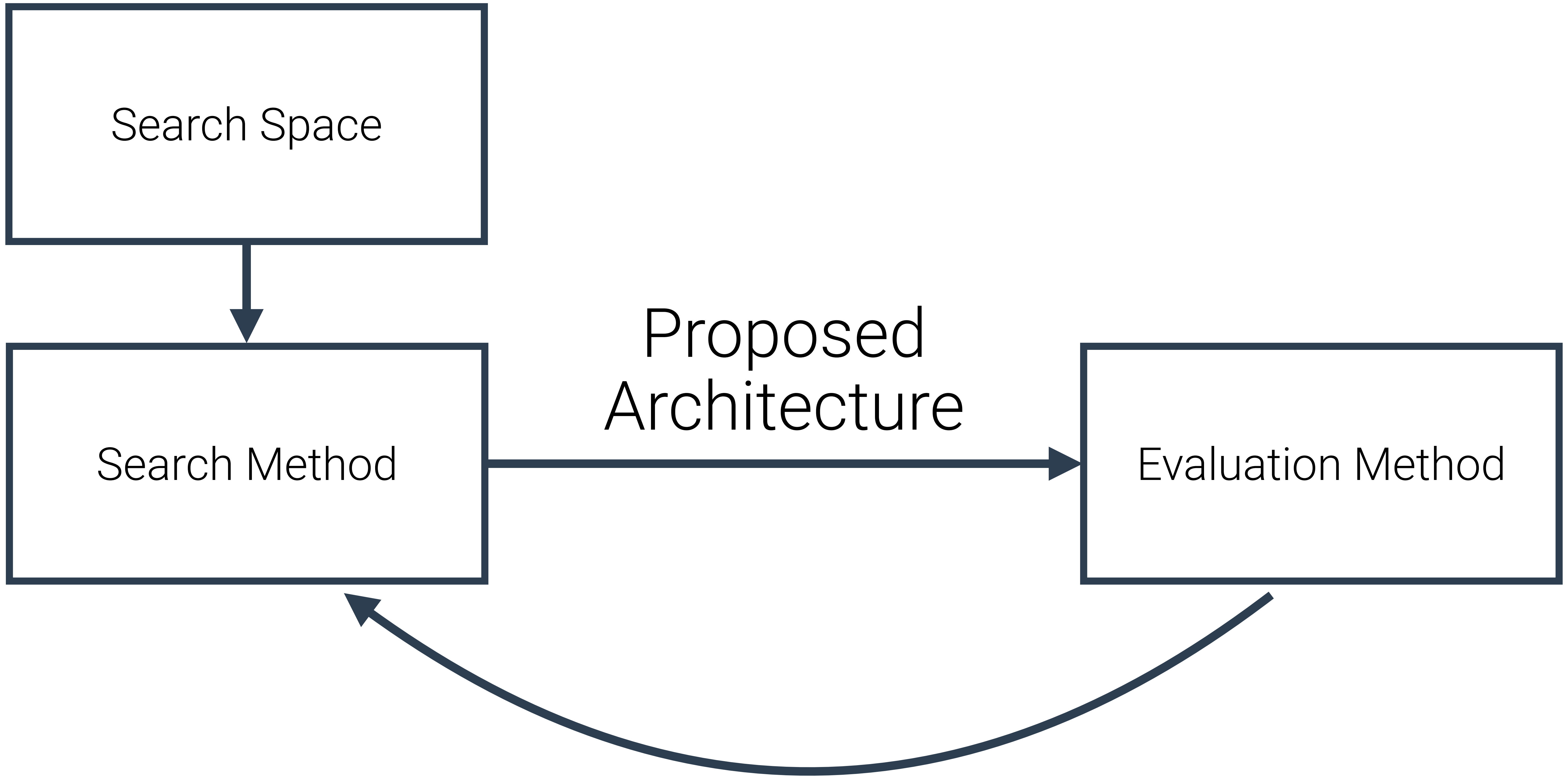
Neural Architecture Search

Evaluation Method

- Generally, this is performance on held-out data.
- Evaluation is typically done by (partially) training the network and evaluating its performance on held-out data.

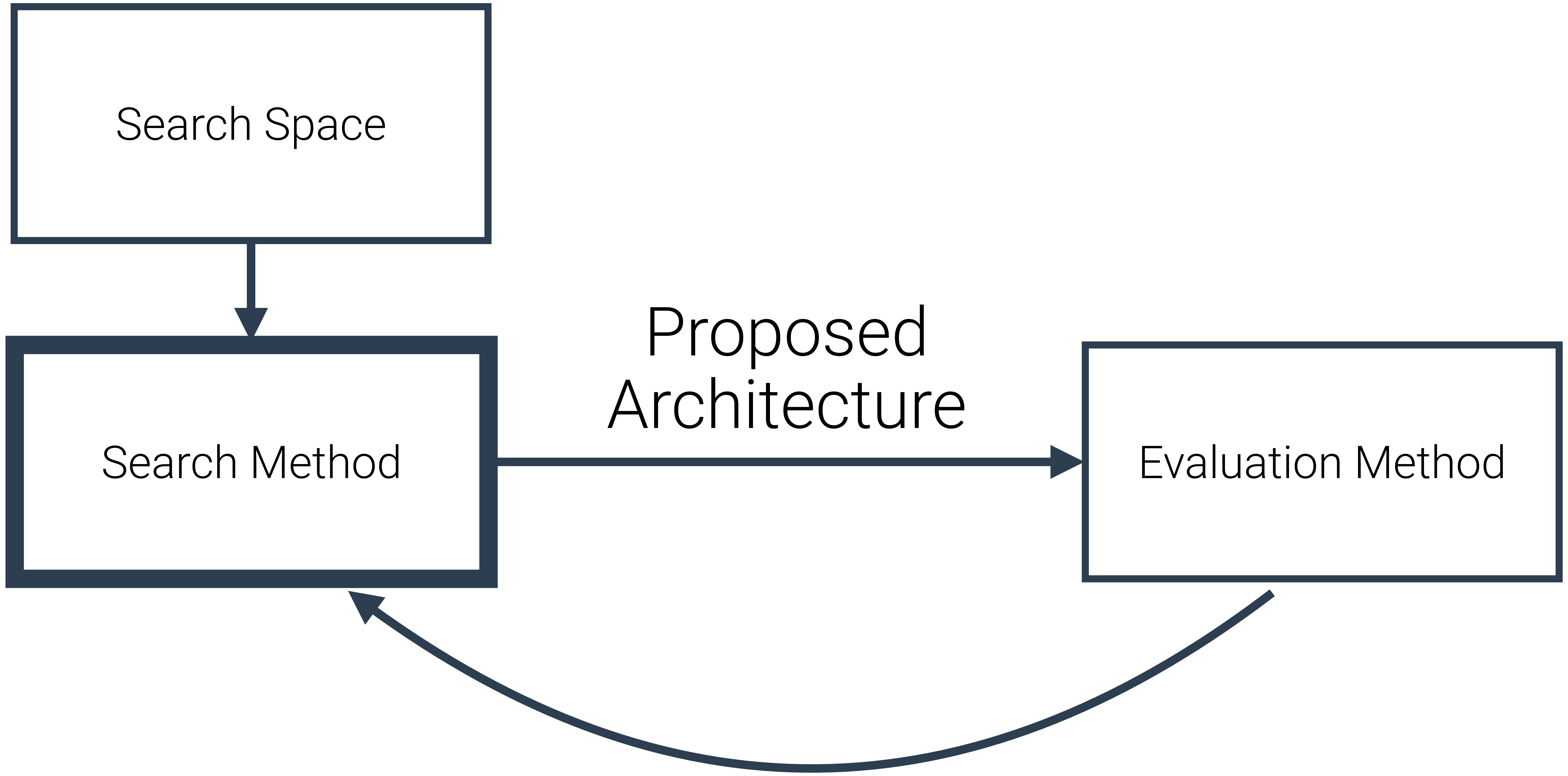
Neural Architecture Search

High Level Overview



Neural Architecture Search

High Level Overview



Search via Reinforcement Learning

Search via Reinforcement Learning

NAS-RL

Search via Reinforcement Learning

NAS-RL

- Motivated by the observation that a DNN architecture can be specified by a string of variable length (i.e. Breadth-first traversal of their DAG)

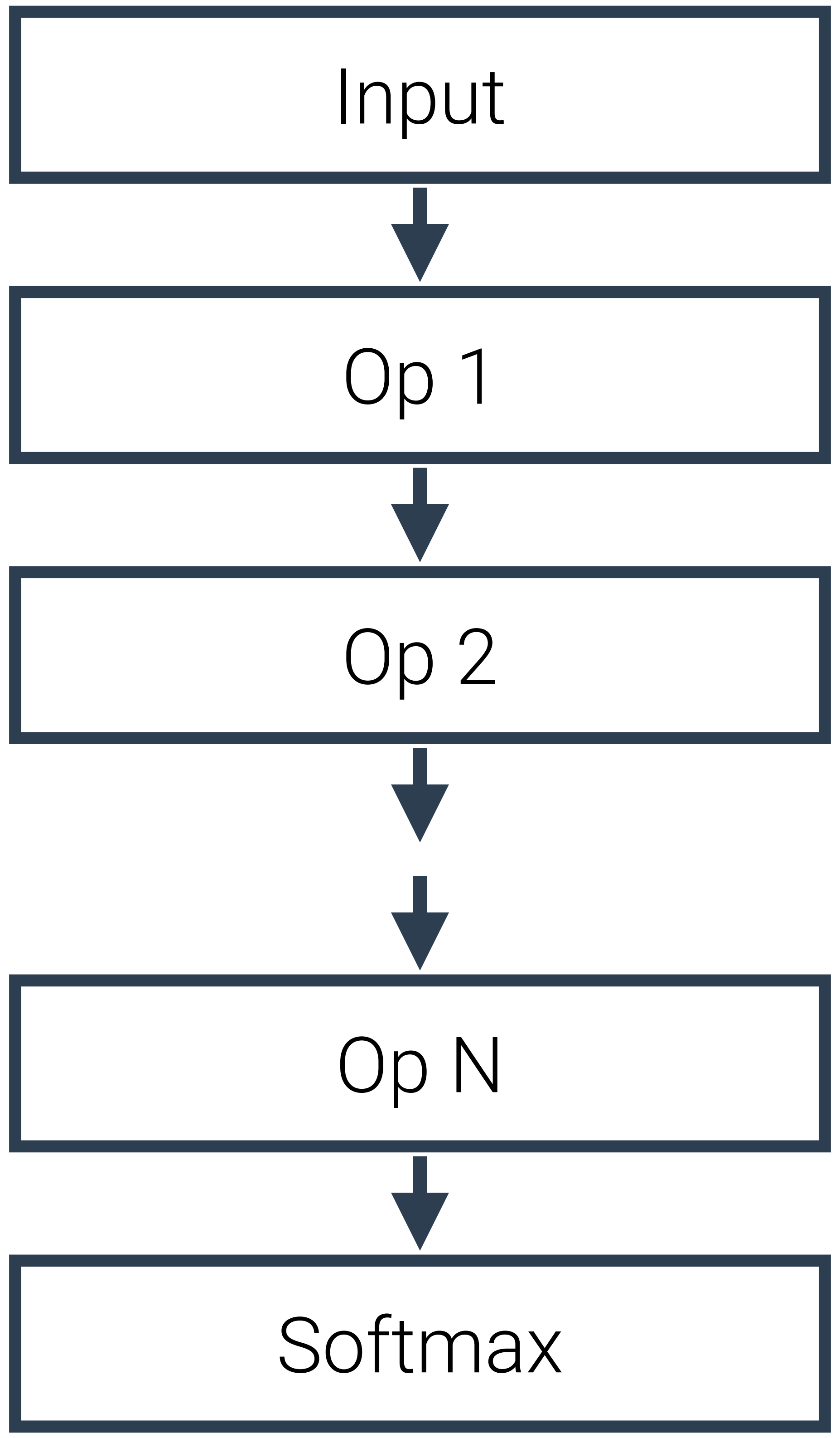
Search via Reinforcement Learning

NAS-RL

- Motivated by the observation that a DNN architecture can be specified by a string of variable length (i.e. Breadth-first traversal of their DAG)
- Use reinforcement learning to train an RNN that builds the network

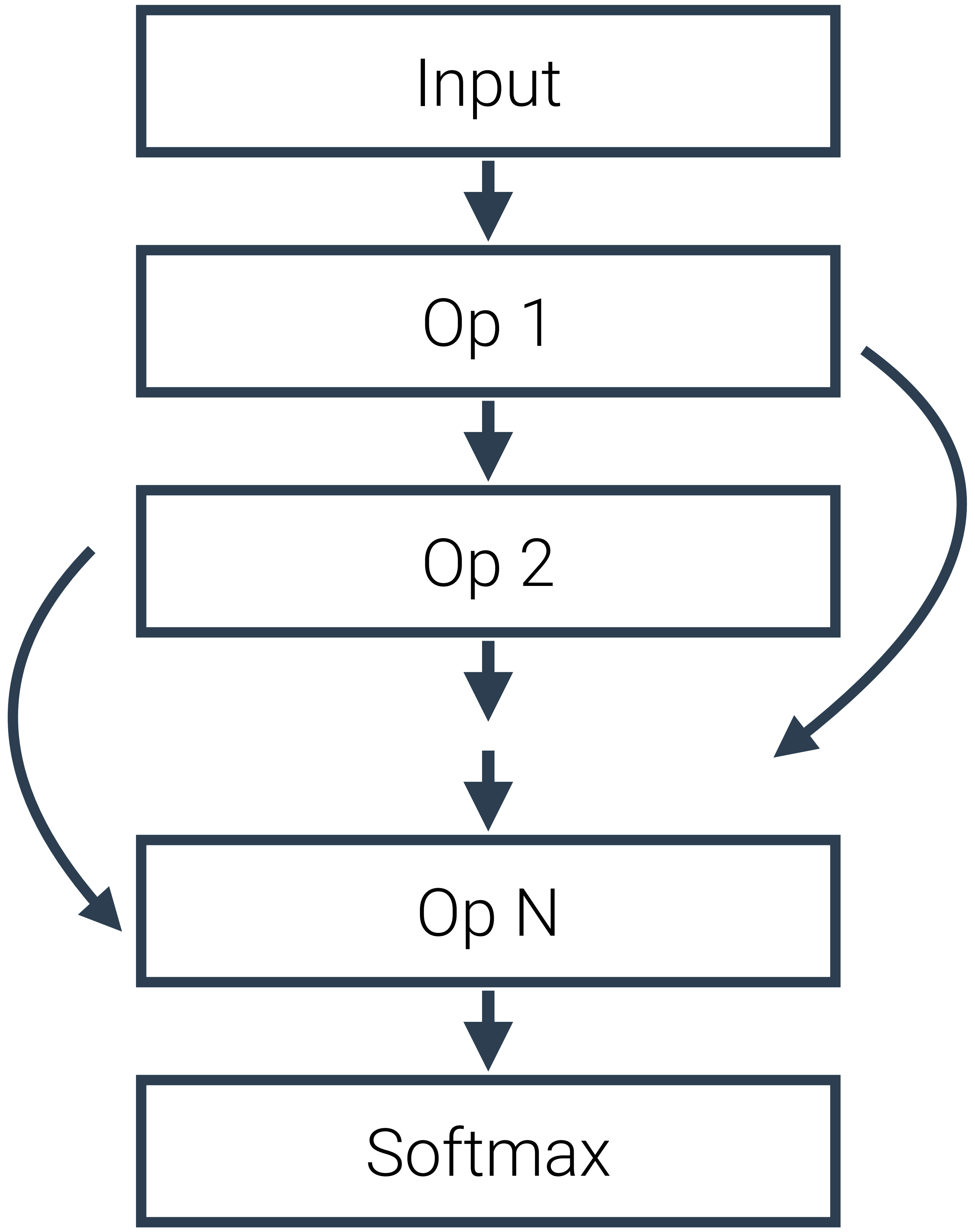
Search via Reinforcement Learning

NAS-RL



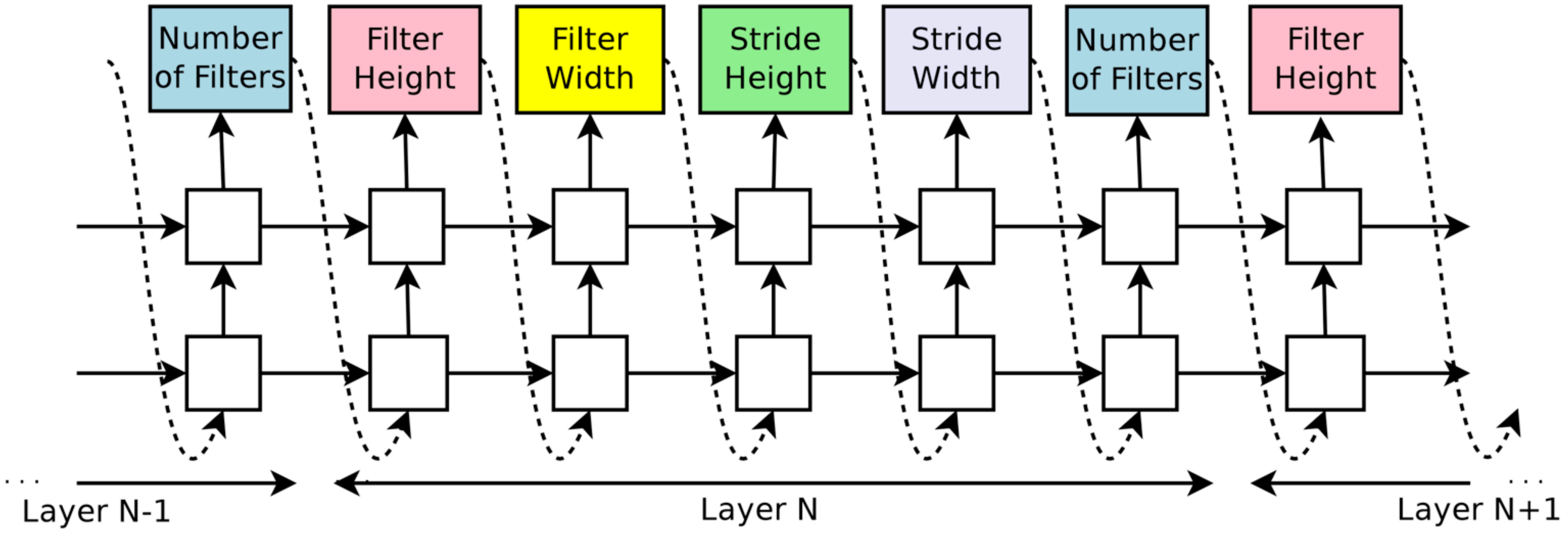
Search via Reinforcement Learning

NAS-RL



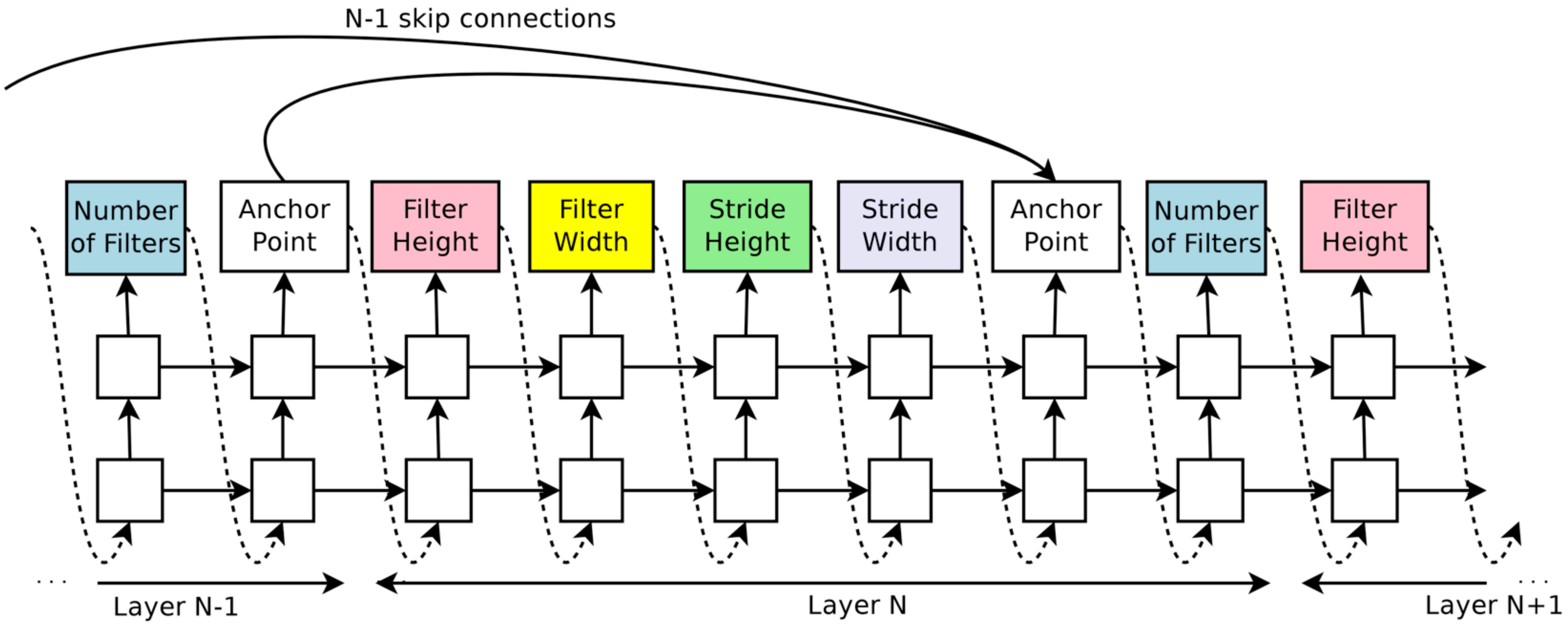
Search via Reinforcement Learning

NAS-RL



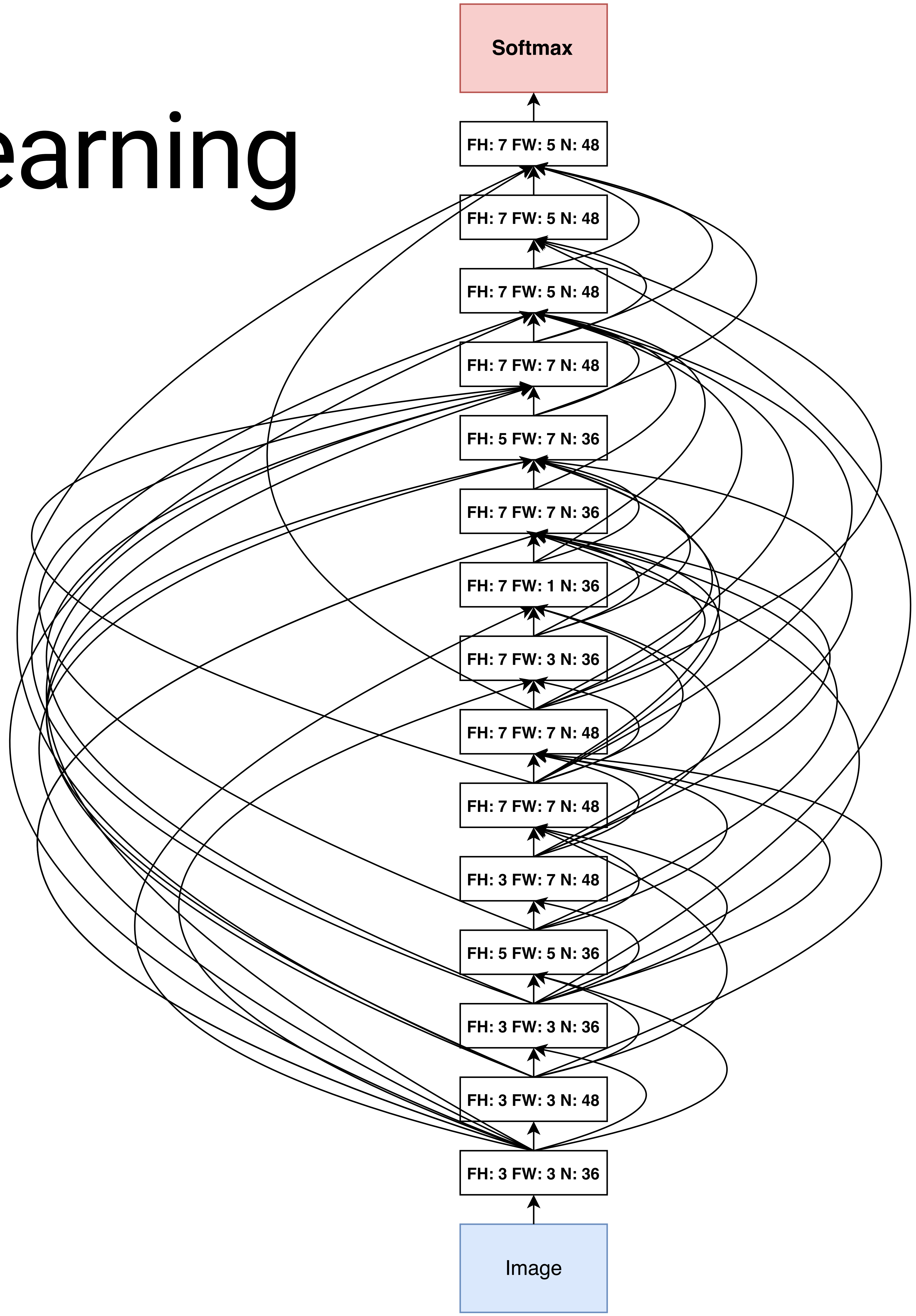
Search via Reinforcement Learning

NAS-RL



Search via Reinforcement Learning

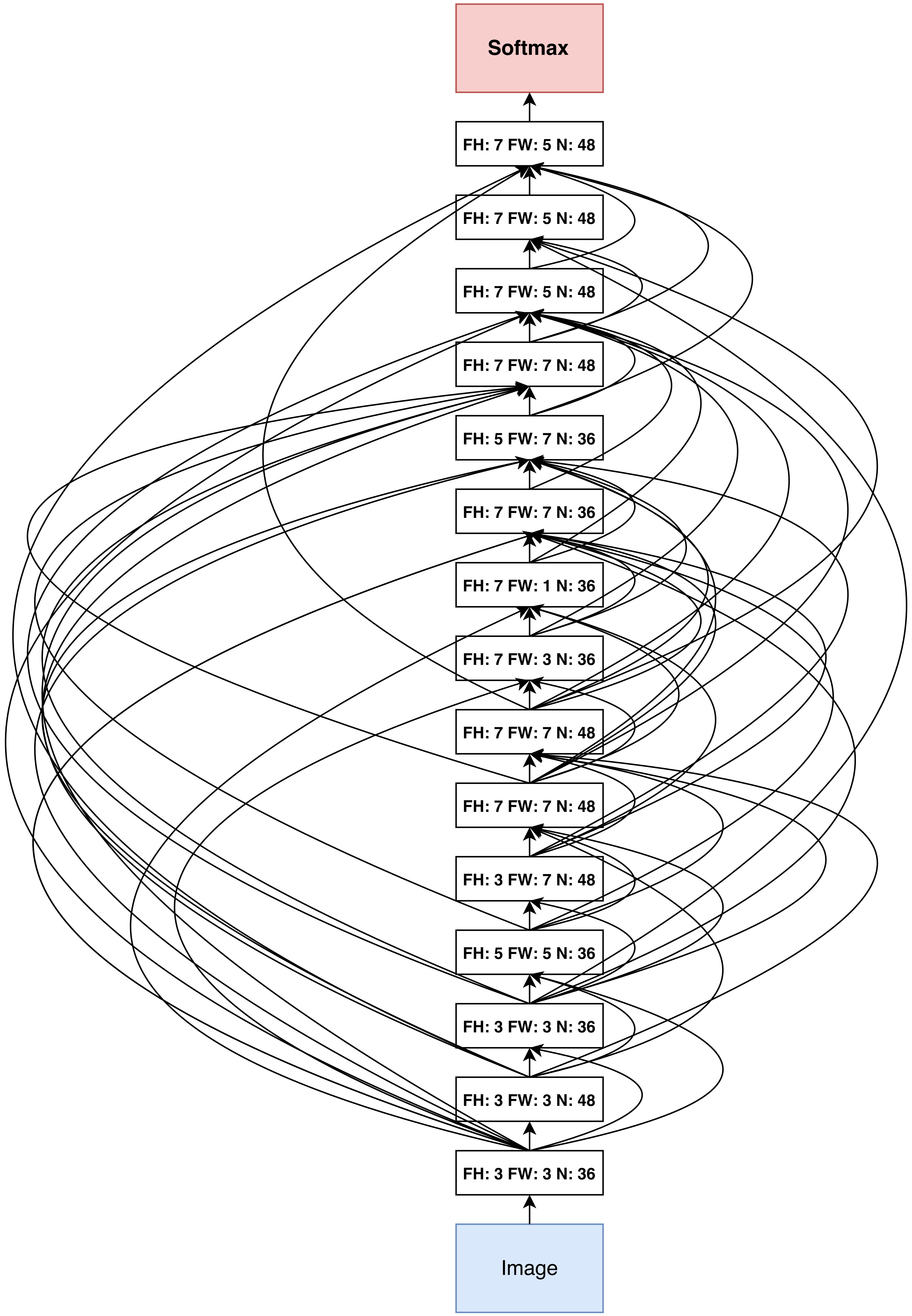
NAS-RL



Search via Reinforcement L

NAS-RL

- Performance is on-par with other CNNs of the time



Search via Reinforcement Learning

NAS-RL

- This is a very general method

Search via Reinforcement Learning

NAS-RL

- This is a very general method
- The cost of that is compute: This used 800 GPUs (for an unspecified amount of time) and trained >12,000 candidate architectures

Search via Reinforcement Learning

NASNet

- Instead, limit the search space with “blocks”

Search via Reinforcement Learning

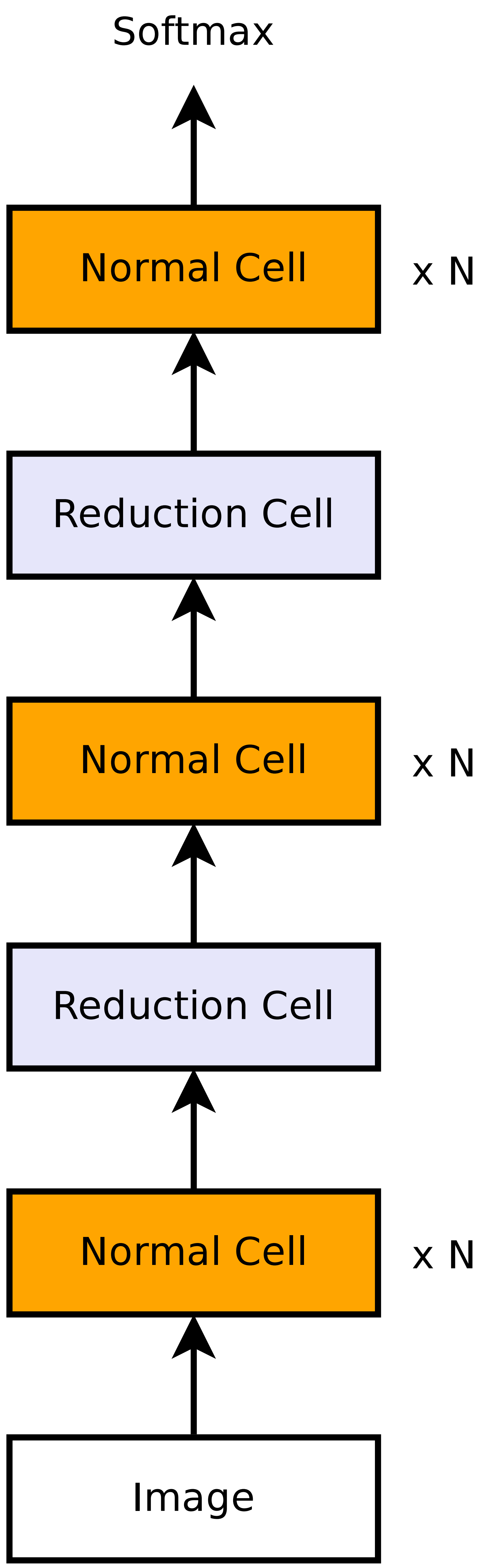
NASNet

- Instead, limit the search space with “blocks”
- This is similar to “Human Neural Architecture Search”

Search via Reinforcement Learning

NASNet

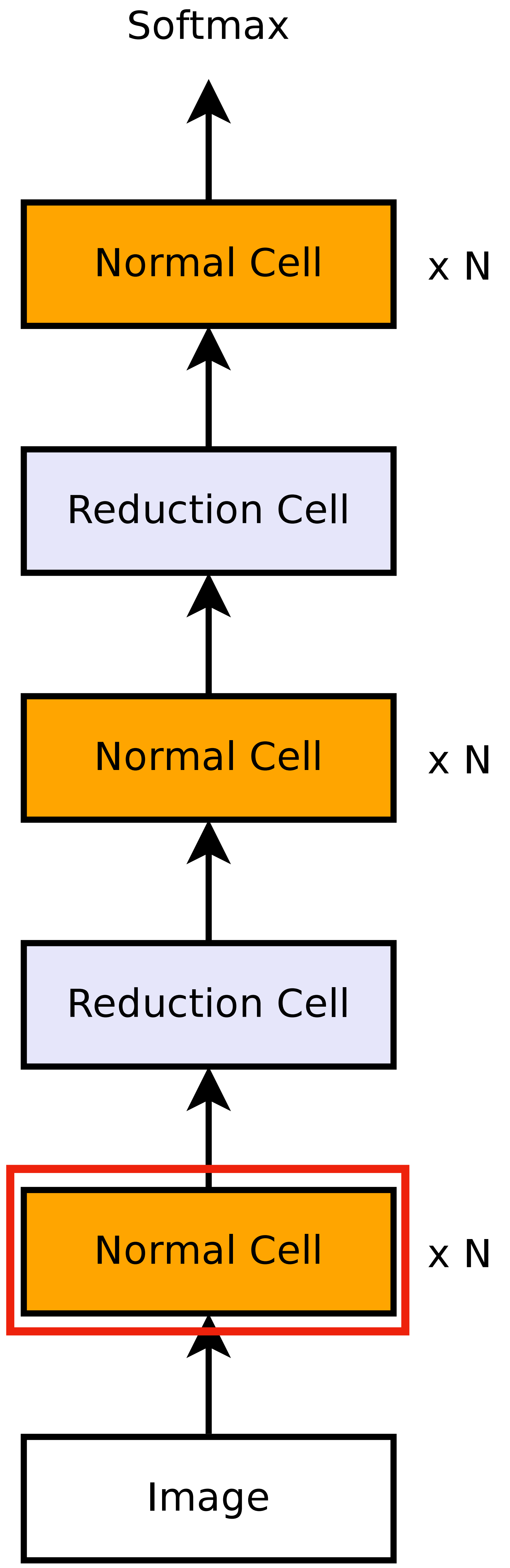
- Instead, limit the search space with “blocks”



Search via Reinforcement Learning

NASNet

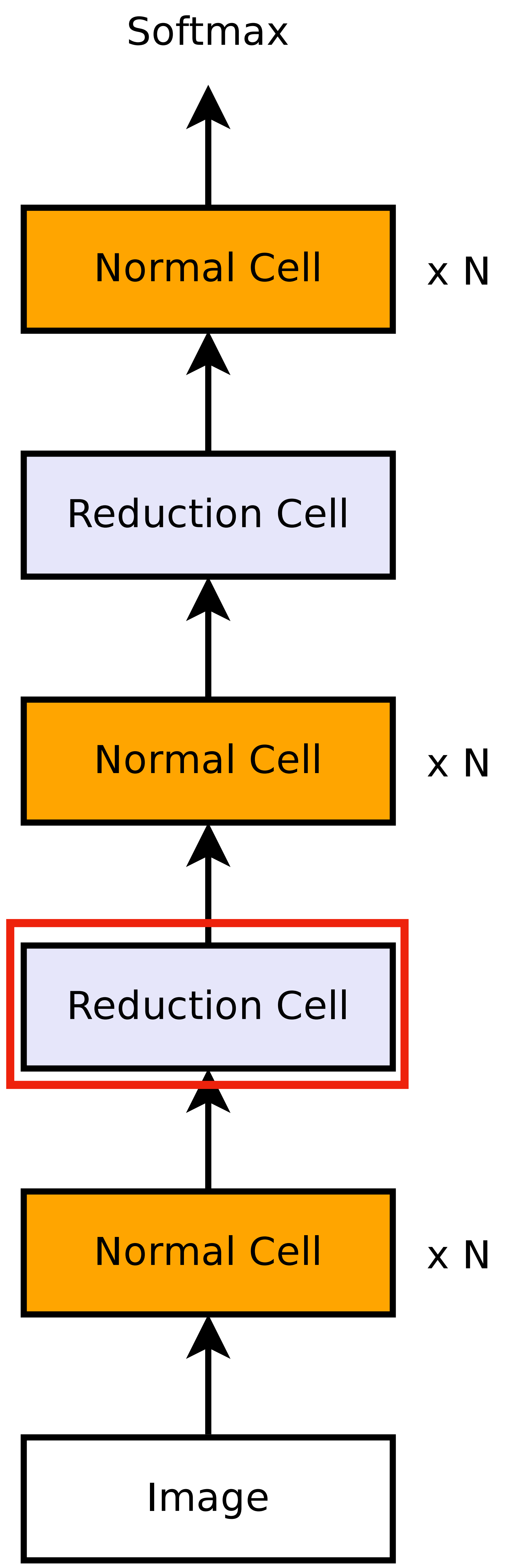
- Instead, limit the search space with “blocks”



Search via Reinforcement Learning

NASNet

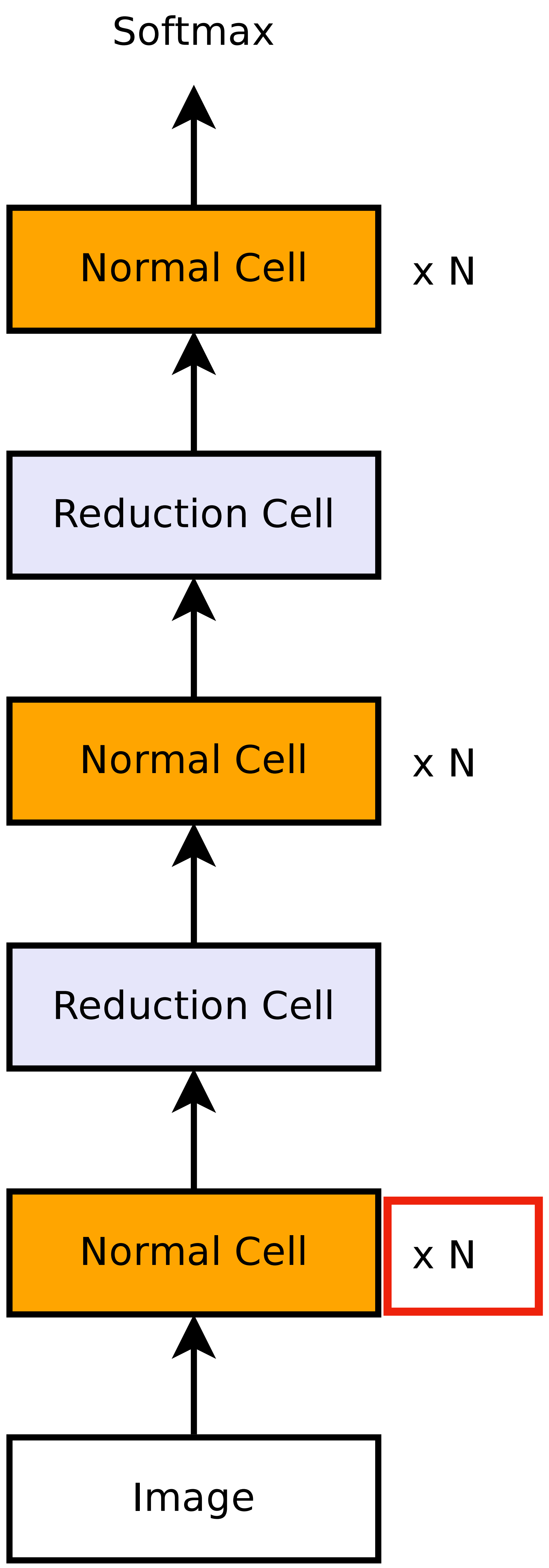
- Instead, limit the search space with “blocks”



Search via Reinforcement Learning

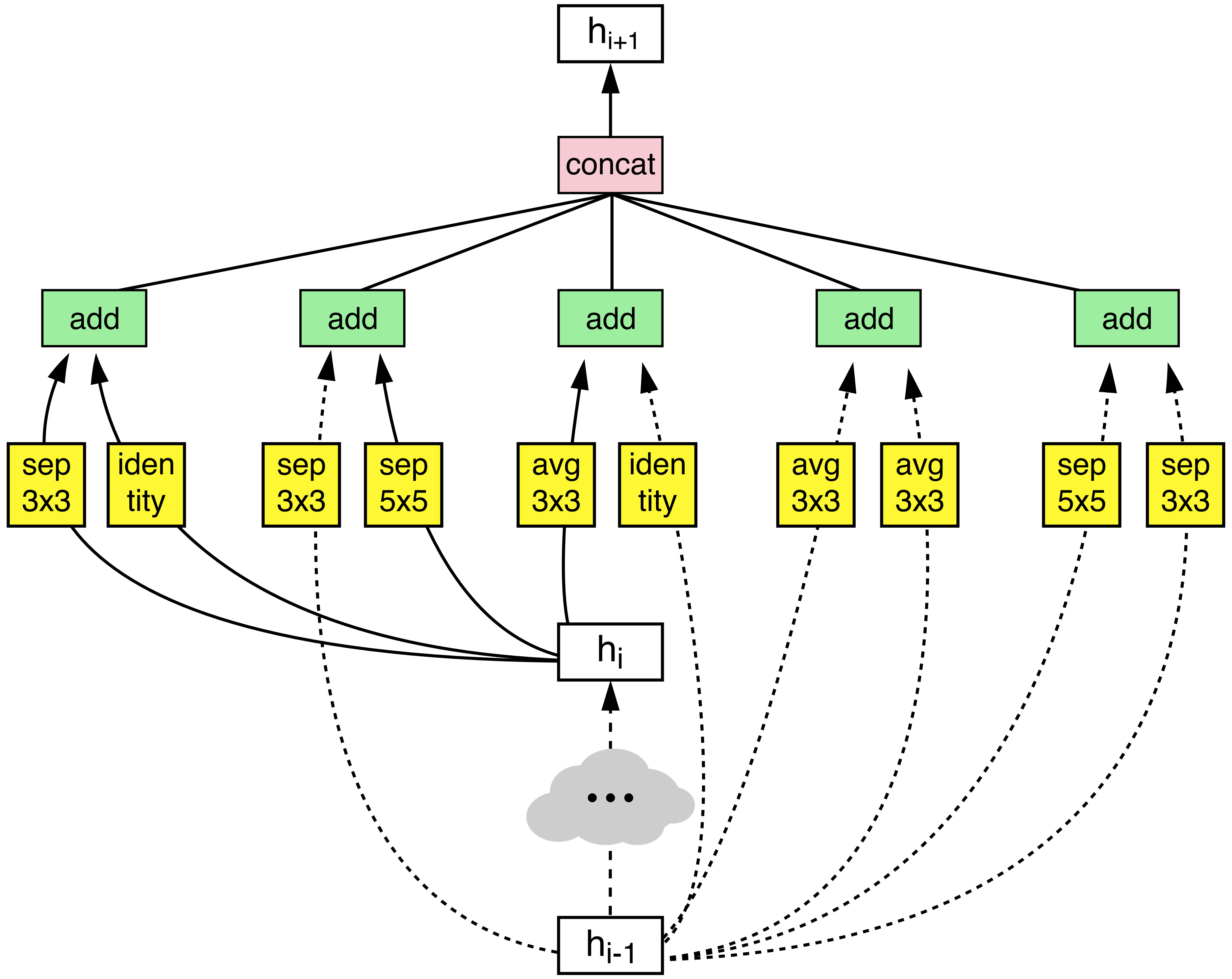
NASNet

- Instead, limit the search space with “blocks”

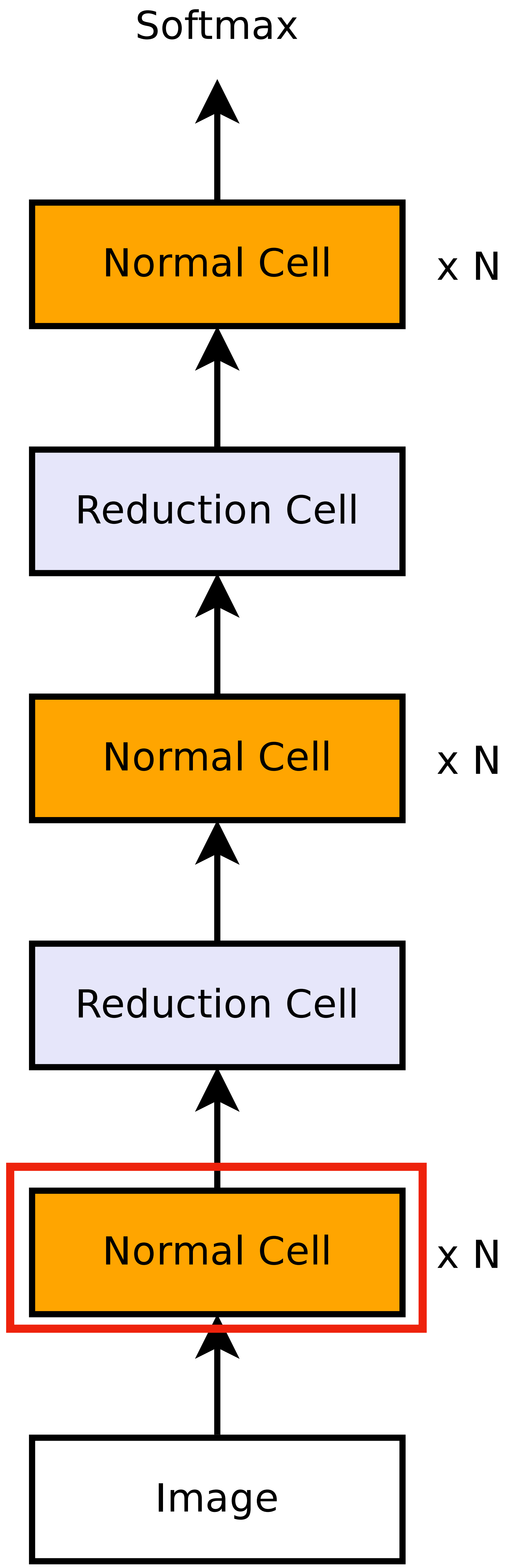


Search via Reinforcement Learning

NASNet

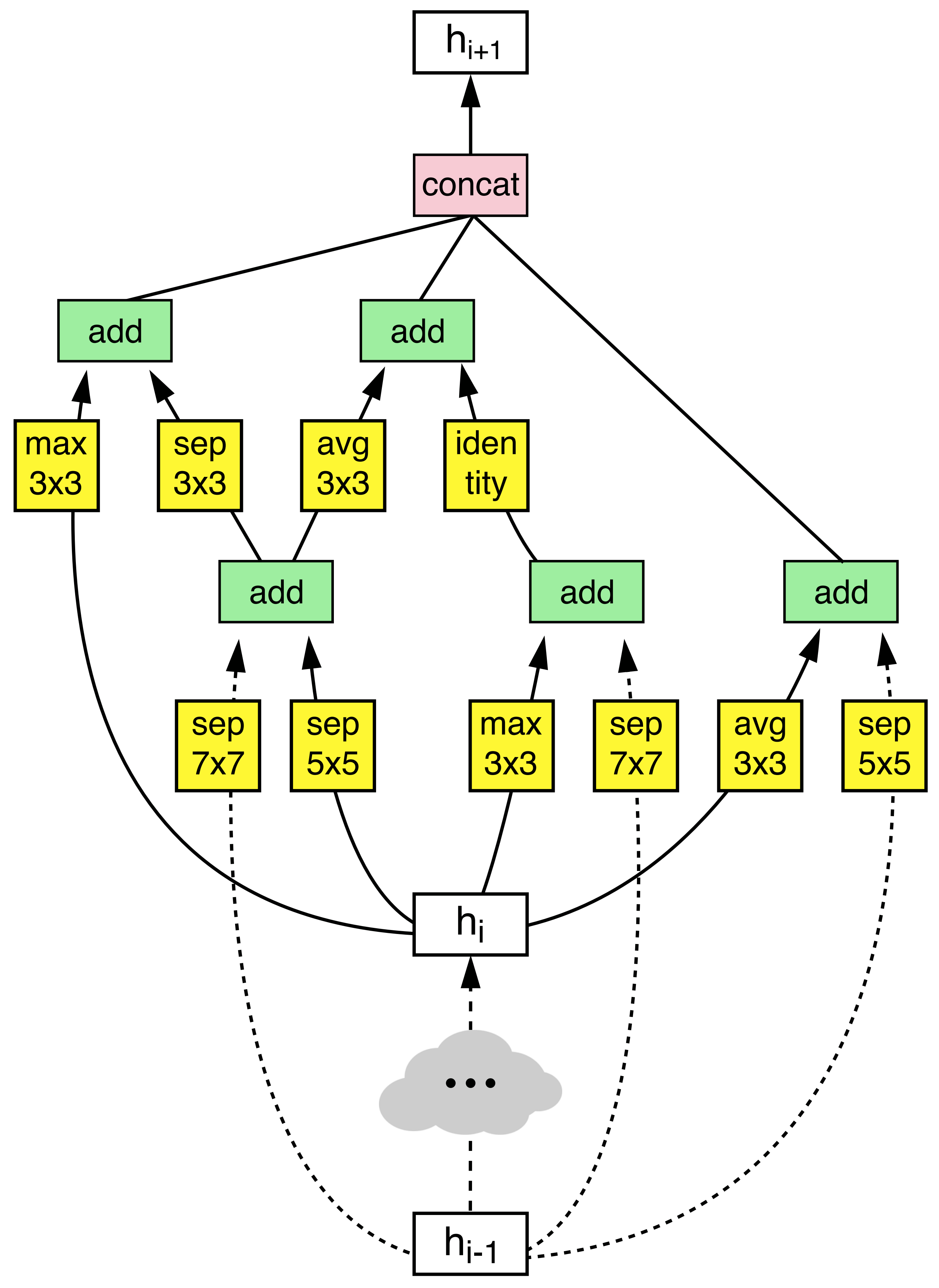


Normal Cell

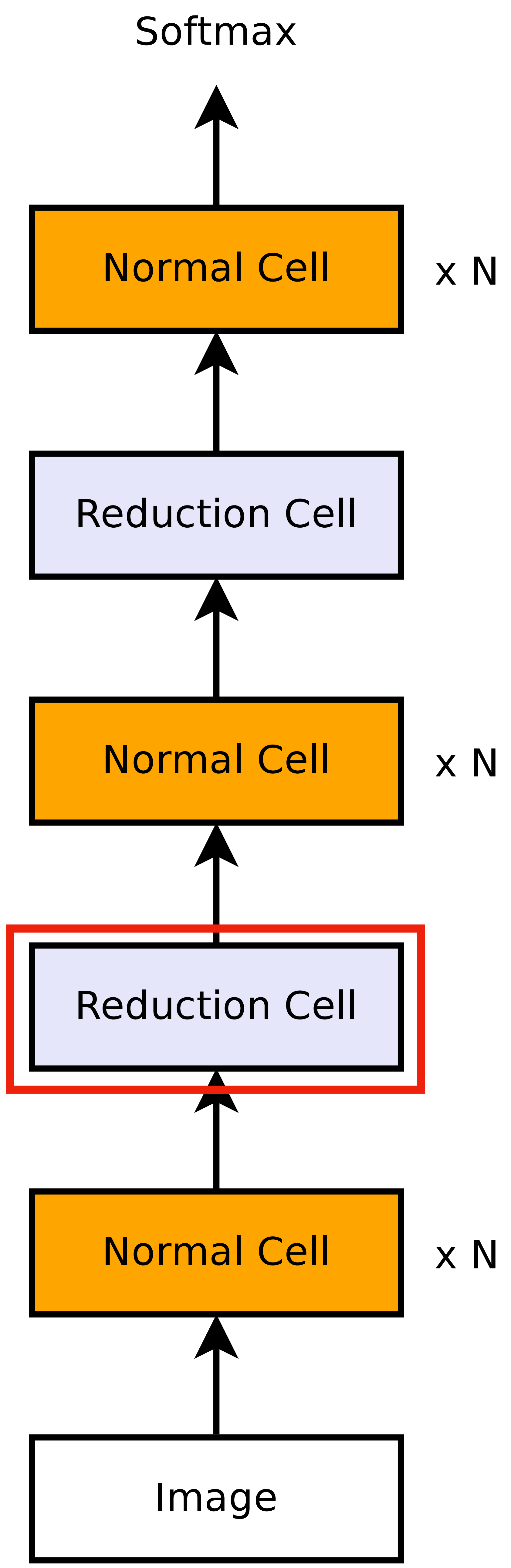


Search via Reinforcement Learning

NASNet



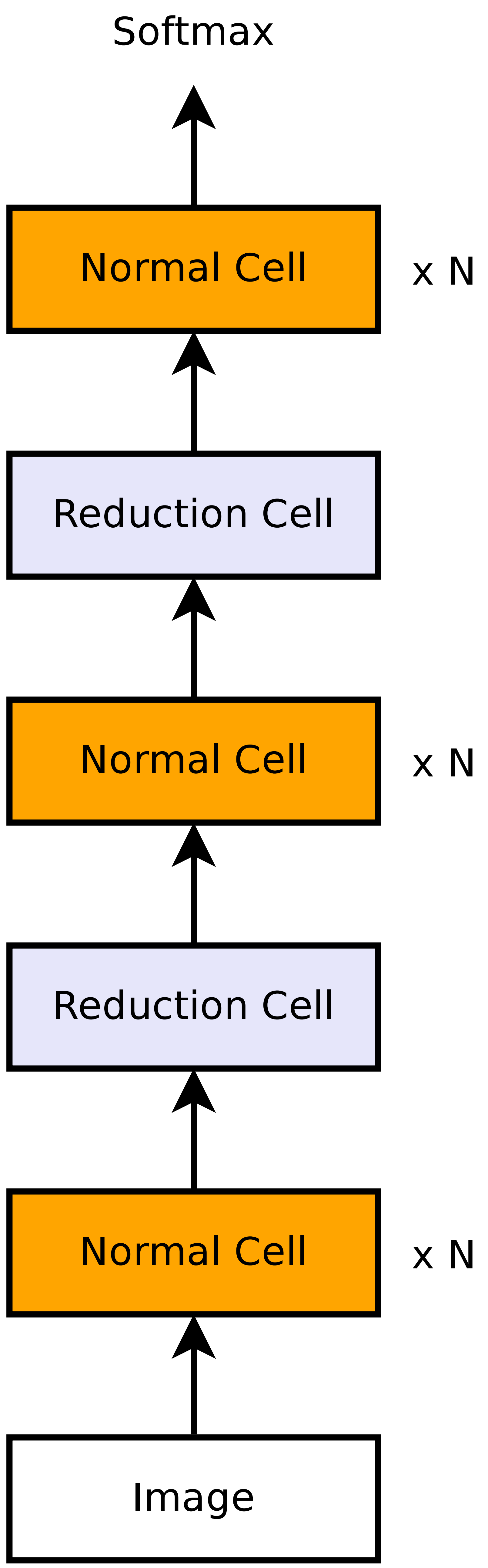
Reduction Cell



Search via Reinforcement Learning

NASNet

- Performance is on-par with other CNNs at the time but with less parameters/compute



Application

Efficient Neural Networks (MnasNet)

Application

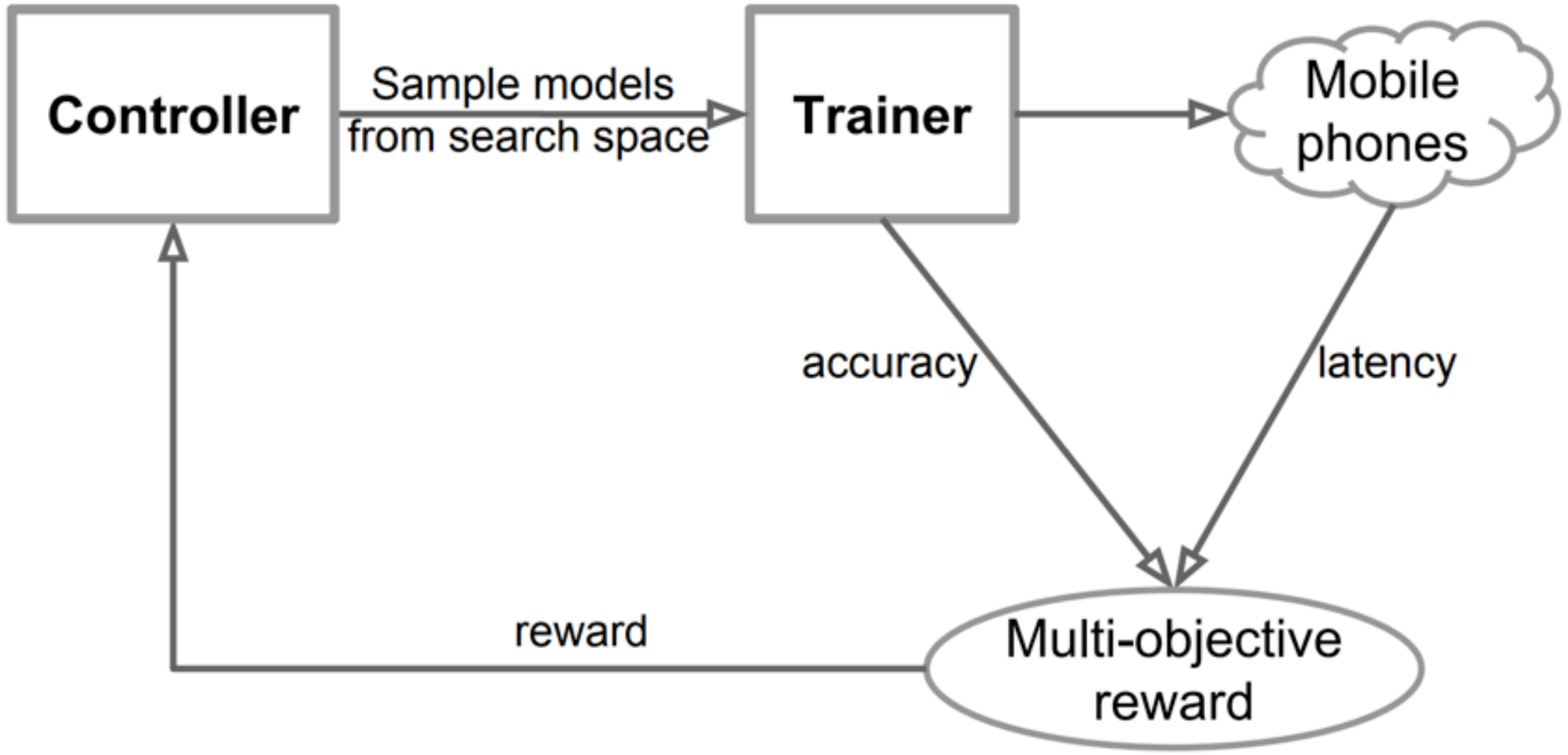
Efficient Neural Networks (MnasNet)

- One benefit of search via RL is that validation performance need not be the *only* metric

Application

Efficient Neural Networks (MnasNet)

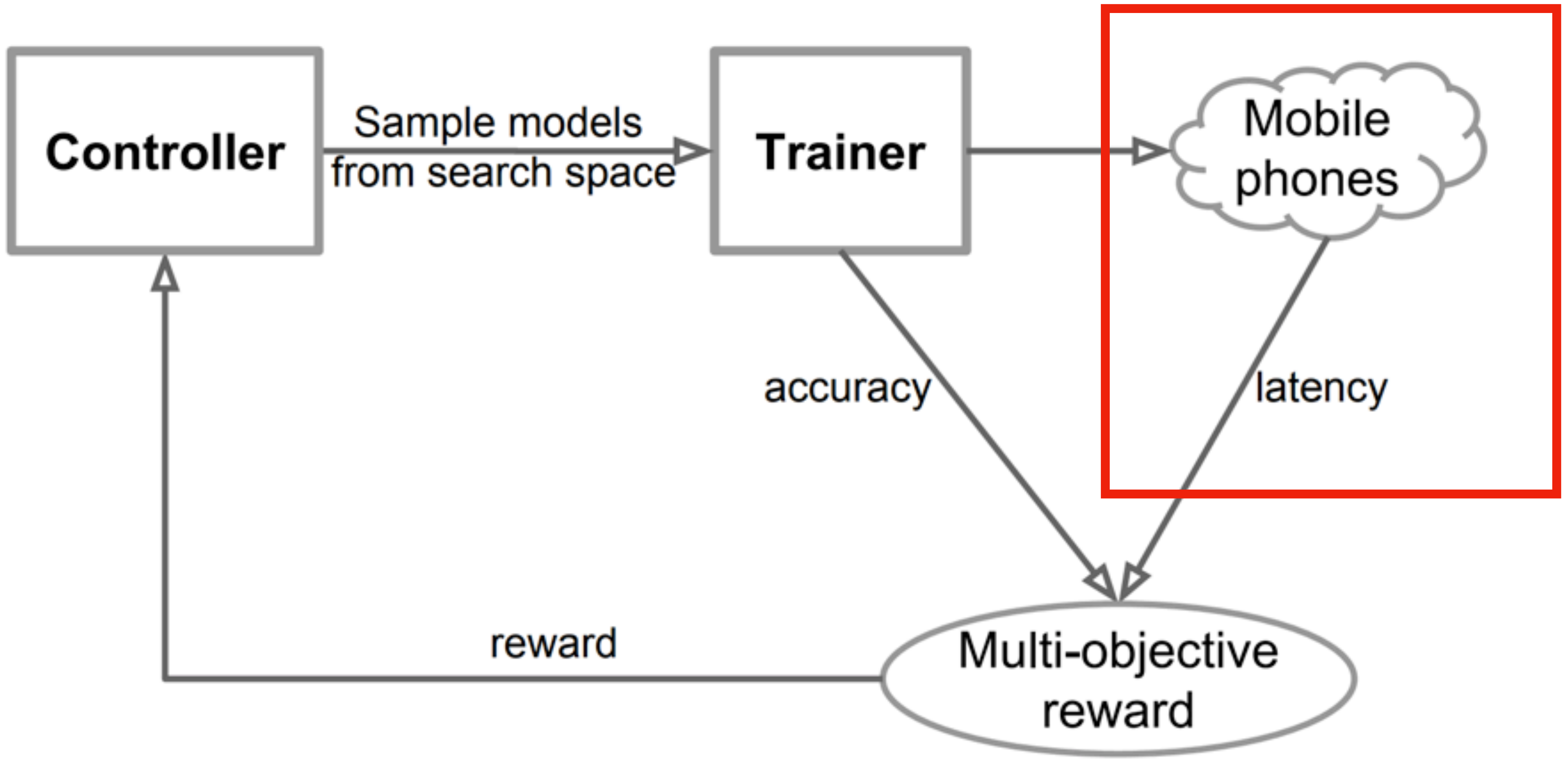
- One benefit of search via RL is that validation performance need not be the *only* metric



Application

Efficient Neural Networks (MnasNet)

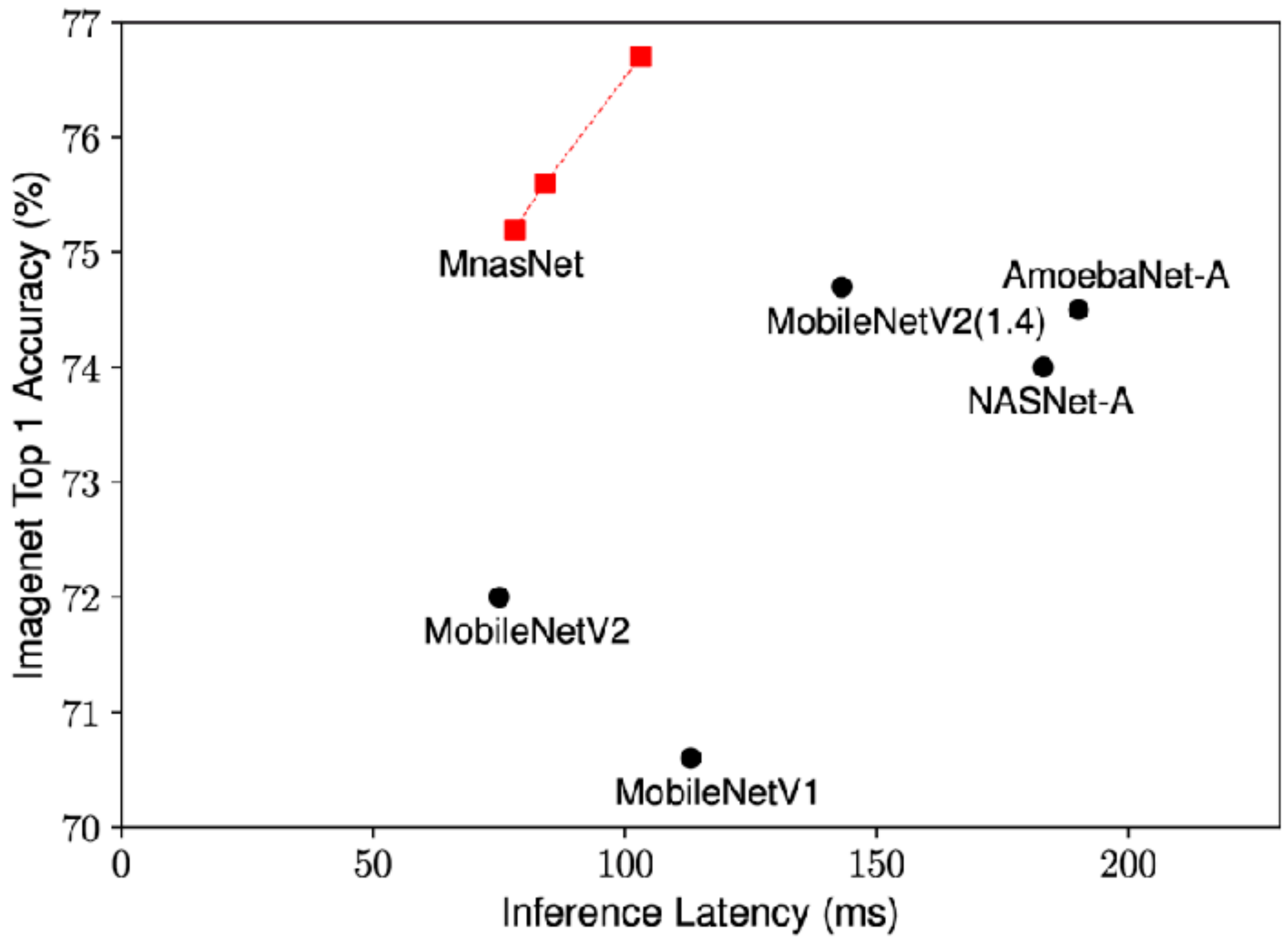
- One benefit of search via RL is that validation performance need not be the *only* metric



Application

Efficient Neural Networks (MnasNet)

- One benefit of search via RL is that validation performance need not be the *only* metric

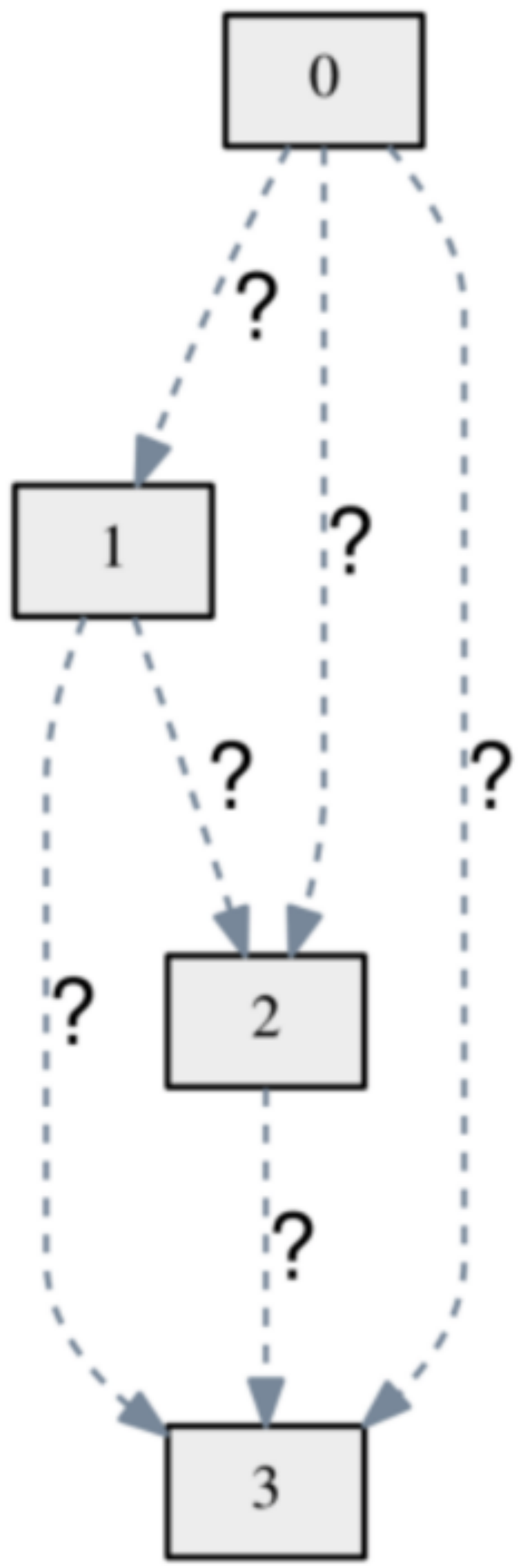


Search via Gradient Optimization

Differentiable Architecture Search (DARTS)

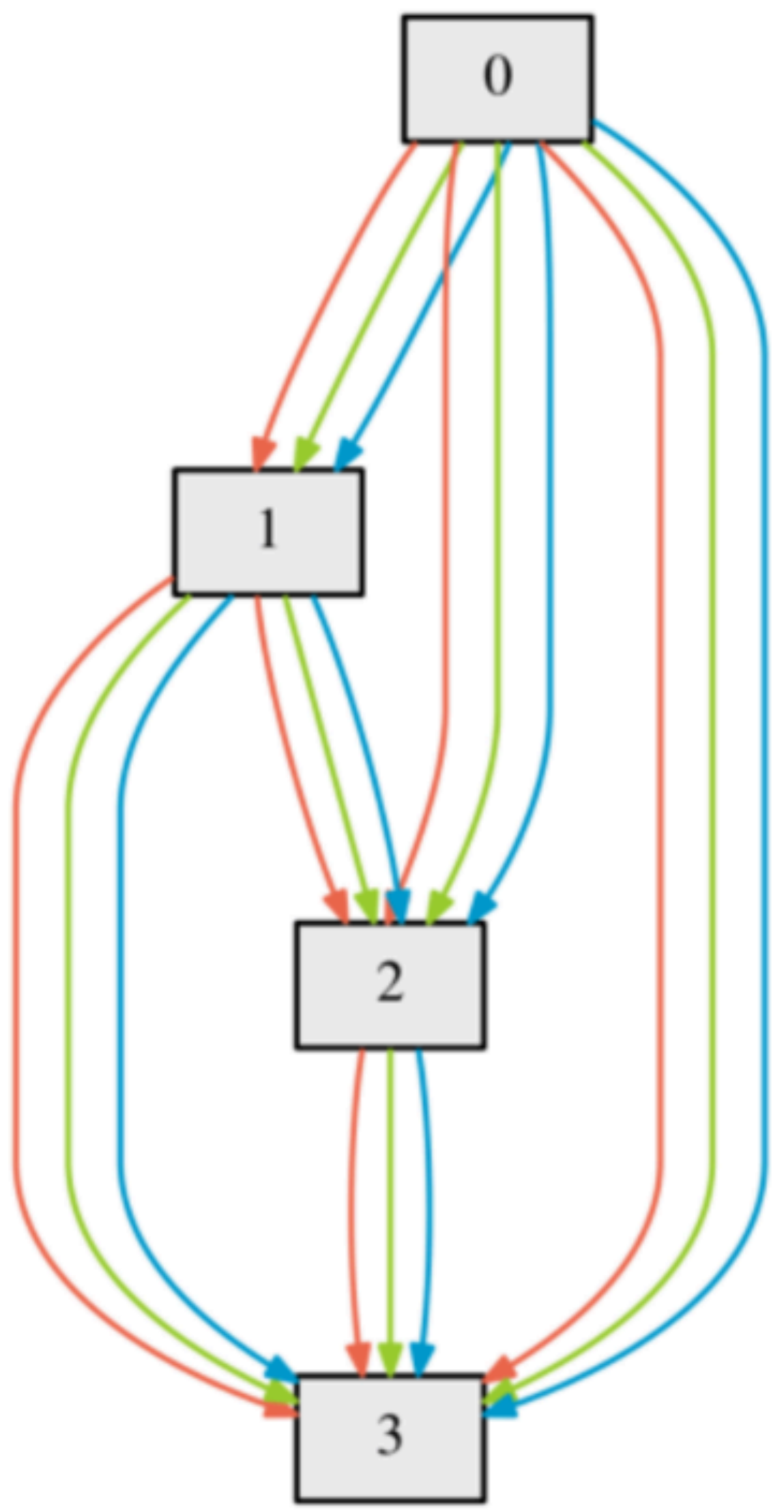
Search via Gradient Optimization

Differentiable Architecture Search (DARTS)



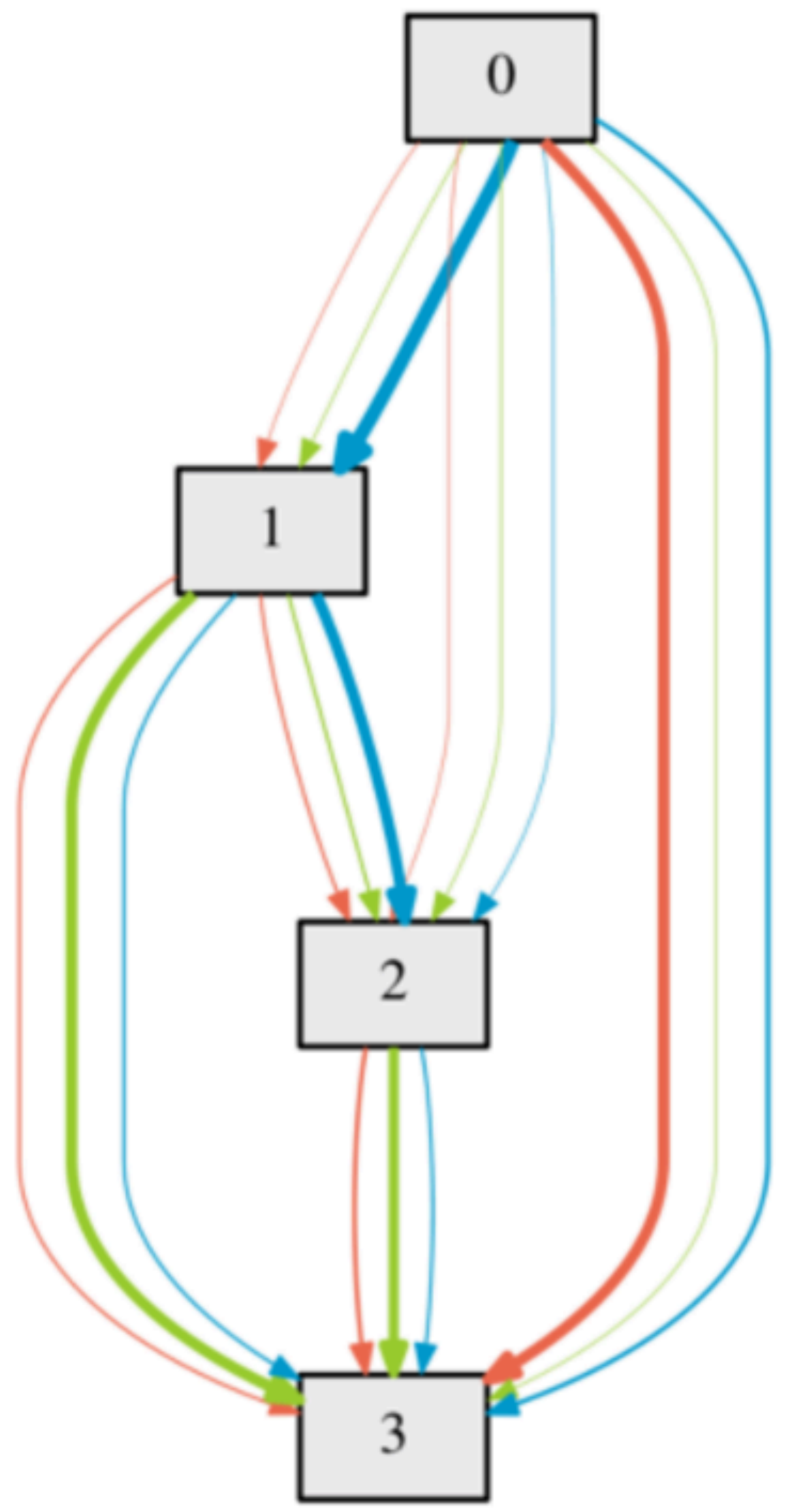
Search via Gradient Optimization

Differentiable Architecture Search (DARTS)



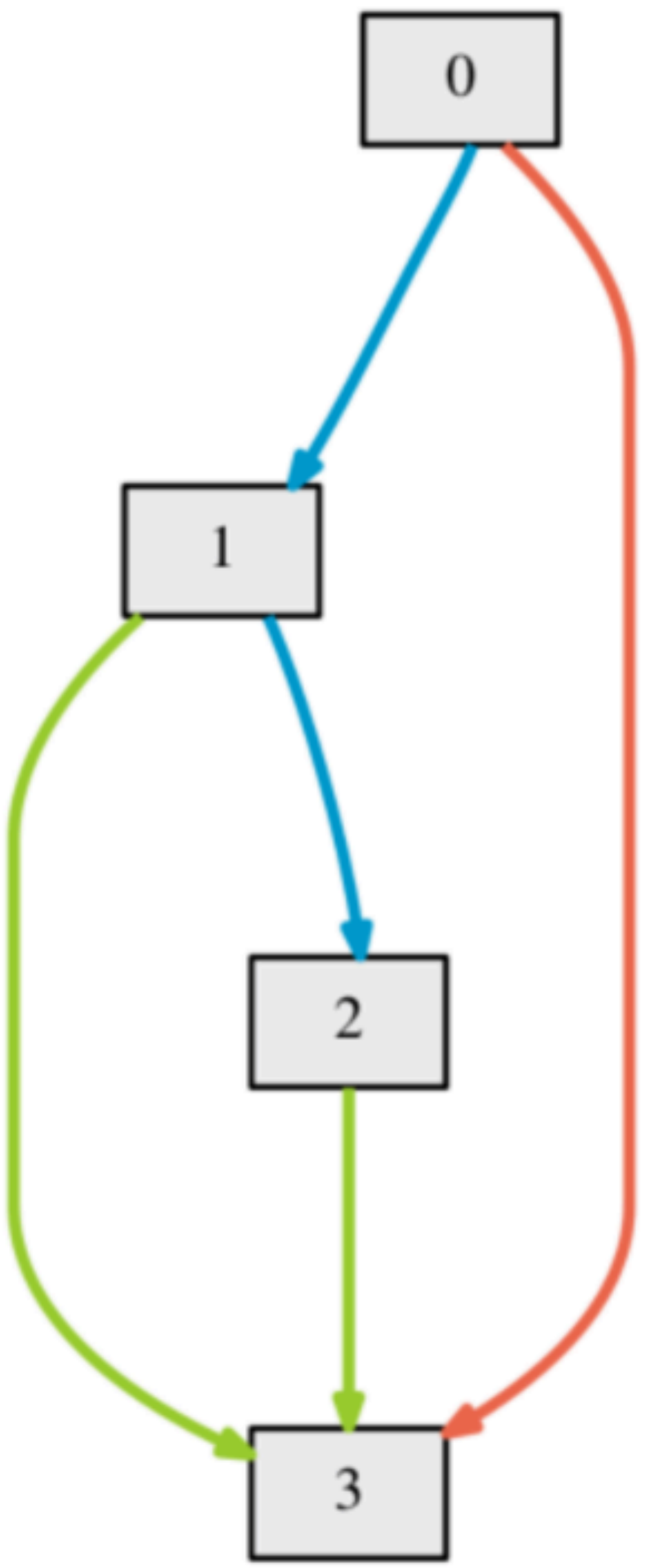
Search via Gradient Optimization

Differentiable Architecture Search (DARTS)



Search via Gradient Optimization

Differentiable Architecture Search (DARTS)



Search via Gradient Optimization

Differentiable Architecture Search (DARTS)

$$\min_{\alpha} \mathcal{L}_{val}(\theta^*(\alpha), \alpha)$$

$$\text{s.t.} \quad \theta^*(\alpha) = \min_{\theta(\alpha)} \mathcal{L}_{train}(\theta, \alpha)$$

Search via Gradient Optimization

Differentiable Architecture Search (DARTS)

$$\min_{\alpha} \mathcal{L}_{val}(\theta^*(\alpha), \alpha)$$

$$\text{s.t.} \quad \boxed{\theta^*(\alpha) = \min_{\theta(\alpha)} \mathcal{L}_{train}(\theta, \alpha)}$$

Search via Gradient Optimization

Differentiable Architecture Search (DARTS)

$$\boxed{\min_{\alpha} \mathcal{L}_{val}(\theta^*(\alpha), \alpha)}$$

$$\text{s.t.} \quad \theta^*(\alpha) = \min_{\theta(\alpha)} \mathcal{L}_{train}(\theta, \alpha)$$

Search via Gradient Optimization

Differentiable Architecture Search (DARTS)

$$\nabla_{\alpha}\mathcal{L}_{val}(\theta^*(\alpha), \alpha)$$

Search via Gradient Optimization

Differentiable Architecture Search (DARTS)

$$\nabla_\alpha \mathcal{L}_{val}(\theta^*(\alpha), \alpha)$$
$$\approx \nabla_\alpha \mathcal{L}_{val}(\theta - \nabla_\theta \mathcal{L}_{train}(\theta, \alpha), \alpha)$$

Search via Gradient Optimization

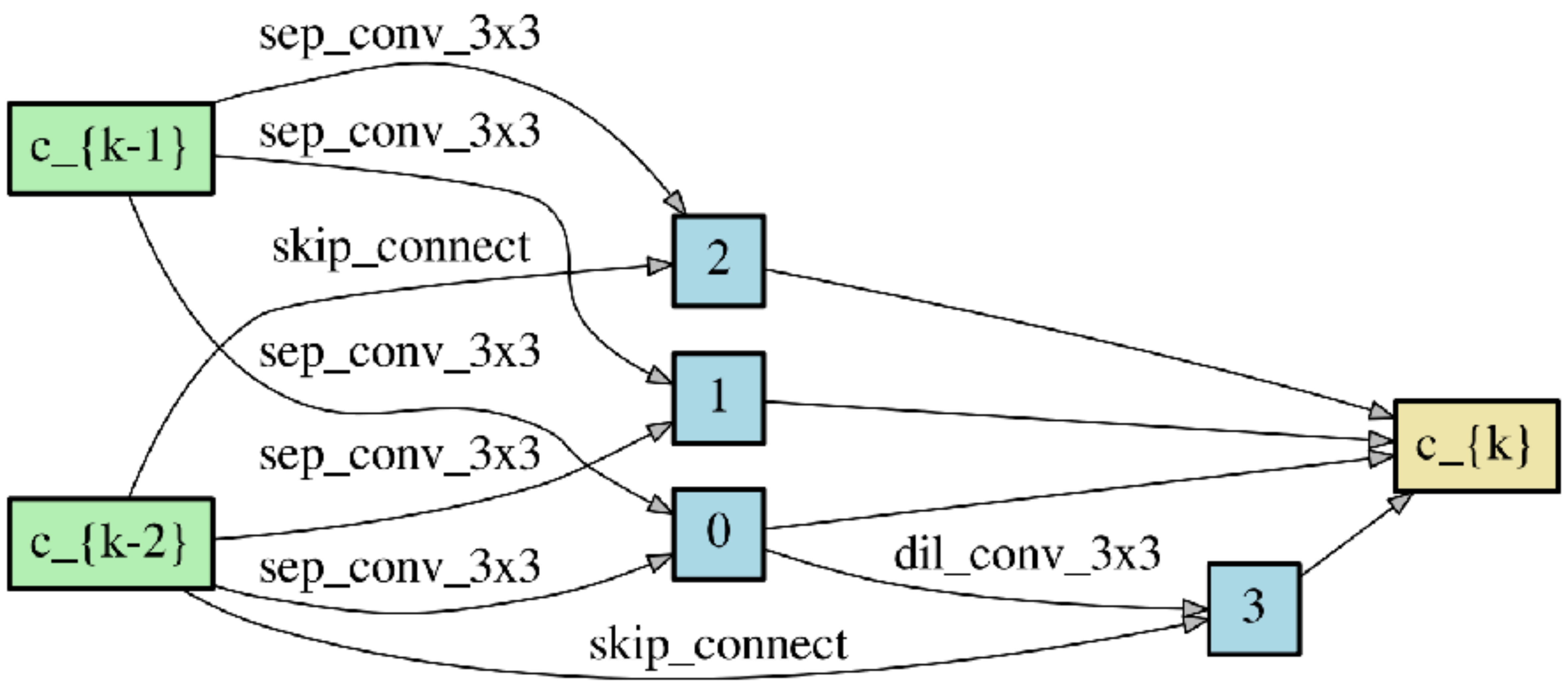
Differentiable Architecture Search (DARTS)

$$
\begin{aligned}
&\nabla_\alpha \mathcal{L}_{val}(\theta^*(\alpha), \alpha) \\
&\approx \nabla_\alpha \mathcal{L}_{val}(\theta - \nabla_\theta \mathcal{L}_{train}(\theta, \alpha), \alpha) \\
&\approx \nabla_\alpha \mathcal{L}_{val}(\theta, \alpha)
\end{aligned}
$$

Search via Gradient Optimization

Differentiable Architecture Search (DARTS)

- Finds networks with very little computation cost (~1 GPU day) that perform better or on-par with existing NAS methods



Search via Scoring

Search via Scoring (without training)

Neural Architecture Search without Training

Search via Scoring (without training)

Neural Architecture Search without Training

- How well a given architecture will do when fully trained can be approximated by how “flexible” the network is.

Search via Scoring (without training)

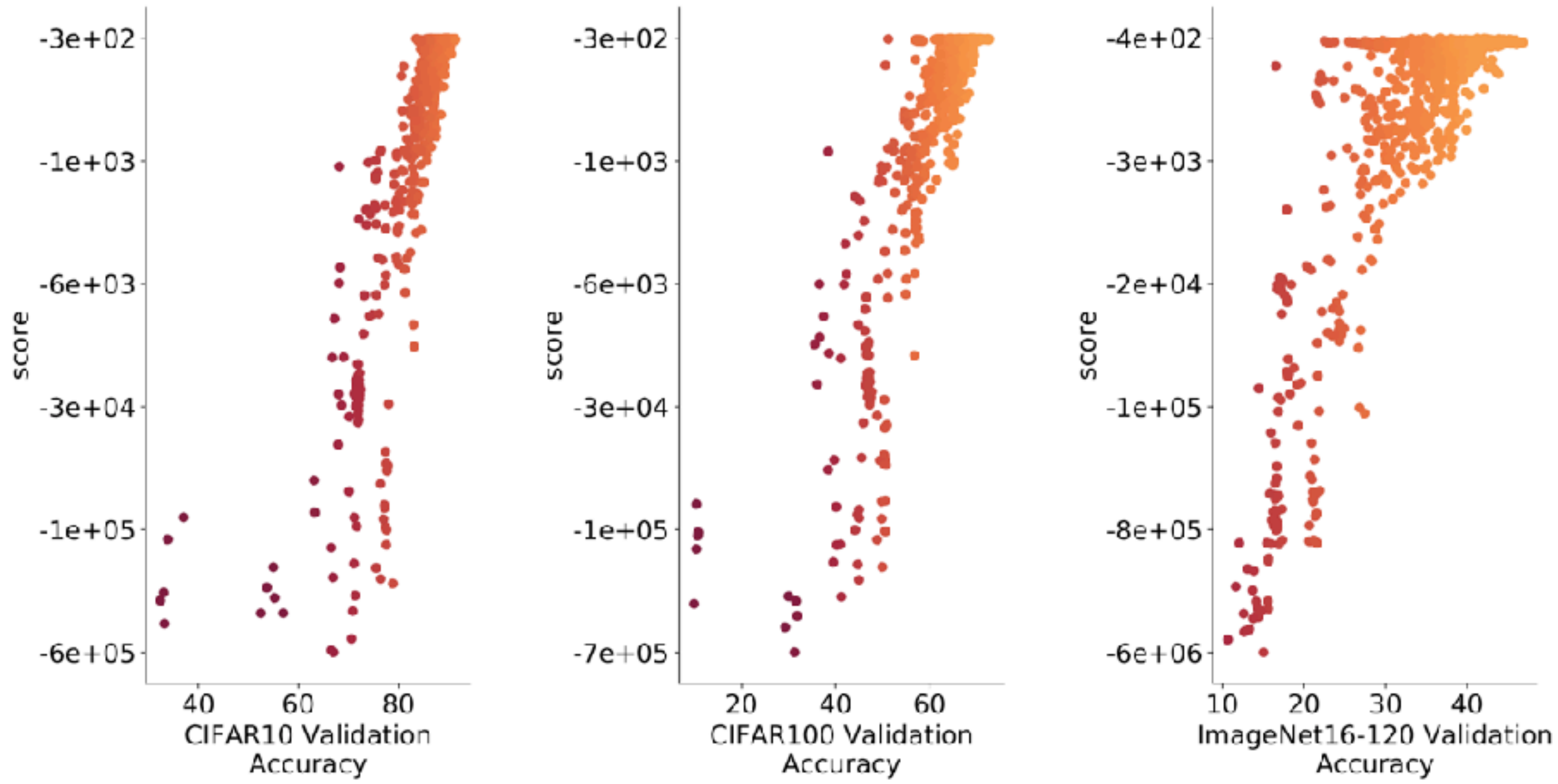
Neural Architecture Search without Training

- How well a given architecture will do when fully trained can be approximated by how “flexible” the network is. This “flexibility” can be determined without training.

Search via Scoring (without training)

Neural Architecture Search without Training

- How well a given architecture will do when fully trained can be approximated by how “flexible” the network is. This “flexibility” can be determined without training.



Summary

- Neural Architecture Search (NAS) focuses on automatically finding highly performant network architectures

Summary

- Neural Architecture Search (NAS) focuses on automatically finding highly performant network architectures

Google's AutoML

Summary

- Neural Architecture Search (NAS) focuses on automatically finding highly performant network architectures
- Search is commonly done with either RL or gradient methods (e.g. DARTS)

Summary

- Neural Architecture Search (NAS) focuses on automatically finding highly performant network architectures
- Search is commonly done with either RL or gradient methods (e.g. DARTS)
- One fruitful use has been searching for compute efficient networks