# NAS-Bench-1Shot1:
## Benchmarking and Dissecting One-Shot Neural Architecture Search

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

DeToL 07.11.2019

**Julien Siems, Arbër Zela and Frank Hutter**

*Under review as a conference paper at ICLR 2020*

# Motivation

- Recent Neural Architecture Search (NAS) methods use a one-shot model to perform the search.

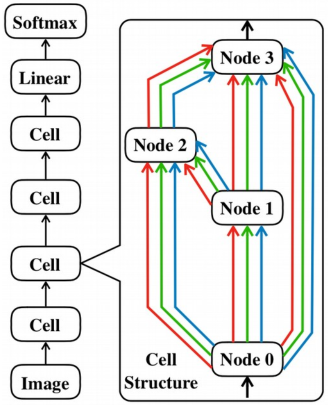



Figure adapted from: Dong, Xuanyi, and Yi Yang. "One-Shot Neural Architecture Search via Self-Evaluated Template Network." *arXiv preprint arXiv:1910.05733* (2019).

# Motivation

- Recent Neural Architecture Search (NAS) methods use a one-shot model to perform the search.

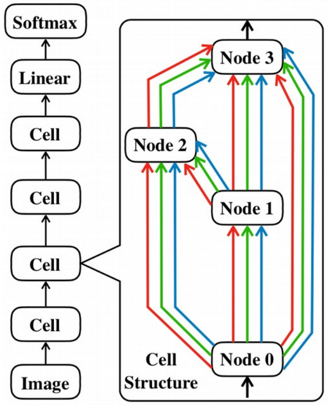



Figure adapted from: Dong, Xuanyi, and Yi Yang. "One-Shot Neural Architecture Search via Self-Evaluated Template Network." *arXiv preprint arXiv:1910.05733* (2019).

- **Reproducibility crisis**
  - Need proper benchmarks [Lindauer and Hutter 2019]
  - NAS-Bench-101 [Ying et al. 2019]

# Motivation

- Recent Neural Architecture Search (NAS) methods use a one-shot model to perform the search.

- Optimize architecture w.r.t. the one-shot validation loss.
  - Goal: Find an architecture which performs well when trained on its own.
  - *Question*: How correlated are the two objectives?

- *Question*: How sensitive are the search methods towards their hyperparameters?

- *Problem*: Independent training of discrete architectures is very expensive.
  - How could we increase the evaluation speed?

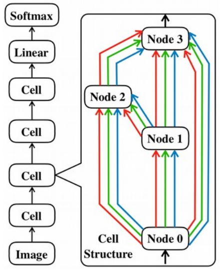



Figure adapted from: Dong, Xuanyi, and Yi Yang. "One-Shot Neural Architecture Search via Self-Evaluated Template Network." *arXiv preprint arXiv:1910.05733* (2019).

# Outline

- **Idea**
- One-Shot NAS Optimizers
- Results
- Conclusion

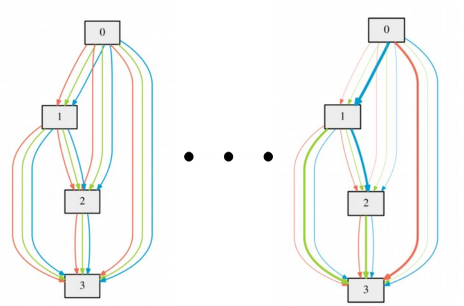
# Idea

## DARTS Search Phases

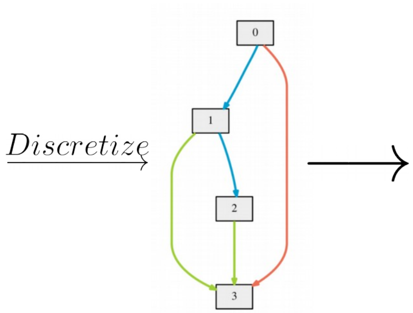
### Architecture Search

$$\min_{\alpha} \; L_{val}(w^*(\alpha), \alpha)$$
$$s.t. \; w^*(\alpha) = \mathrm{argmin}_w \; L_{train}(w, \alpha)$$

Epoch 0 Epoch 50

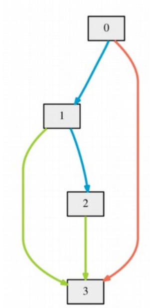



Liu et al. 2018




### Architecture Evaluation

- Train discrete arch. from scratch
- Higher fidelity model:
  - More channels
  - More cells
- Different training hyperparameters



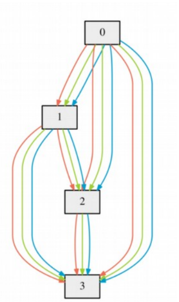
# Idea

## DARTS Search Phases

### Architecture Search

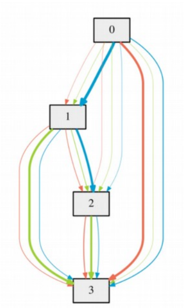
$$\min_{\alpha} \; L_{val}(w^*(\alpha), \alpha)$$
$$s.t. \; w^*(\alpha) = \text{argmin}_w \; L_{train}(w, \alpha)$$

Epoch 0

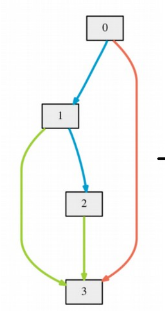


. . .

Epoch 50

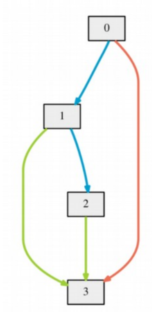


*Discretize*



### Architecture Evaluation

- Train discrete arch. from scratch
- Higher fidelity model:
  - More channels
  - More cells
- Different training hyperparameters



DARTS (first order): **1.5 days**
DARTS (second order): **4 days**

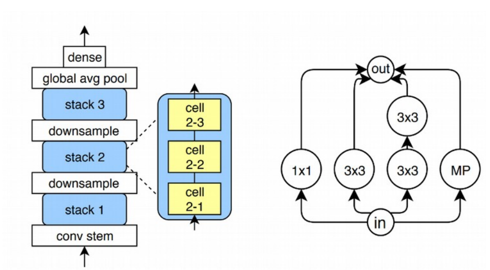
DARTS: **1 day**

*Price to pay to check intermediate architectures*

# Idea

## NASBench-101

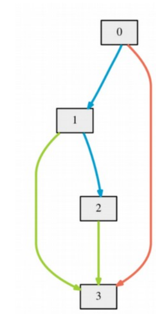
- Exhaustively evaluated search space CIFAR-10 [REF]
  - > 400k unique graphs
- Evaluated on 4 different budgets
- Evaluated 3 times



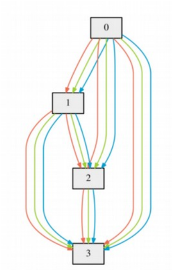

How can we use NASBench for Architecture Evaluation?

## Architecture Evaluation

- Train discrete arch. from scratch
- Higher fidelity model:
  - More channels
  - More cells
- Different training hyperparameters

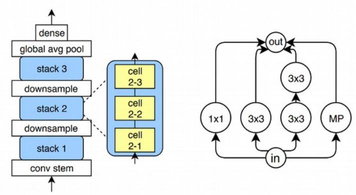


# Idea




DARTS Search Space

- **Representation**: edges are ops, nodes are combinations of tensors
- **Input** of each cell are the **2 previous cells.**
- Intermediate node have **2 incoming edges**
- Output of cell is concatenation of all intermediate node outputs



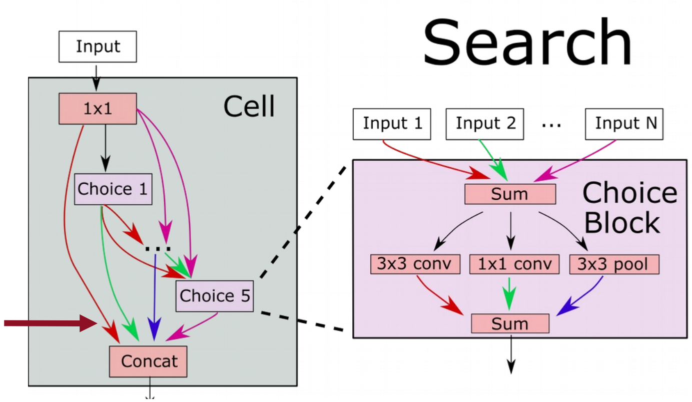

NASBench Search Space

- **Representation**: edges depict tensor flow, nodes are operations
- Limited number of architectures by restricting each cell:
    - **<= 9 edges**
    - <= 5 intermediate nodes
        - Max-Pool, Conv-1x1, Conv-3x3
- Input of each cell is **only previous cell**.

*Architectures in the DARTS Search Space are usually not part of the NASBench Search Space.*

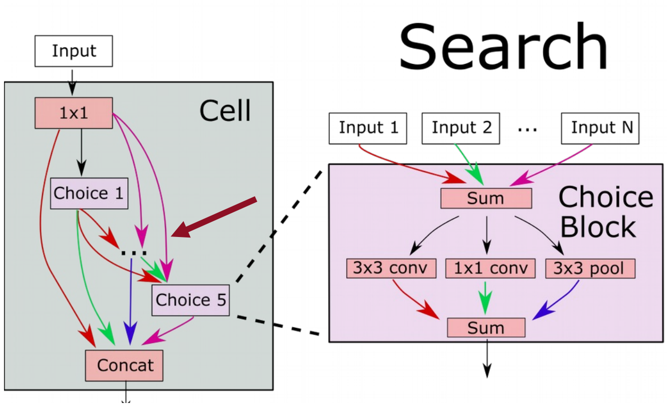
# Idea

- Modified search space by Bender et al. 2018
- Architectural weights:
  - On edges to output
  - On input edges to choice block
  - On the ‘mixed-op’ for each operation



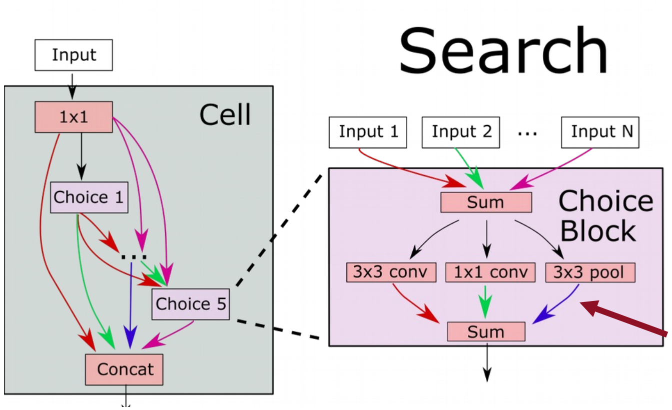
# Idea

- Modified search space by Bender et al. 2018
- Architectural weights:
  - On edges to output
  - On input edges to choice block
  - On the ‘mixed-op’ for each operation



# Idea

- Modified search space by Bender et al. 2018
- Architectural weights:
  - On edges to output
  - On input edges to choice block
  - On the ‘mixed-op’ for each operation

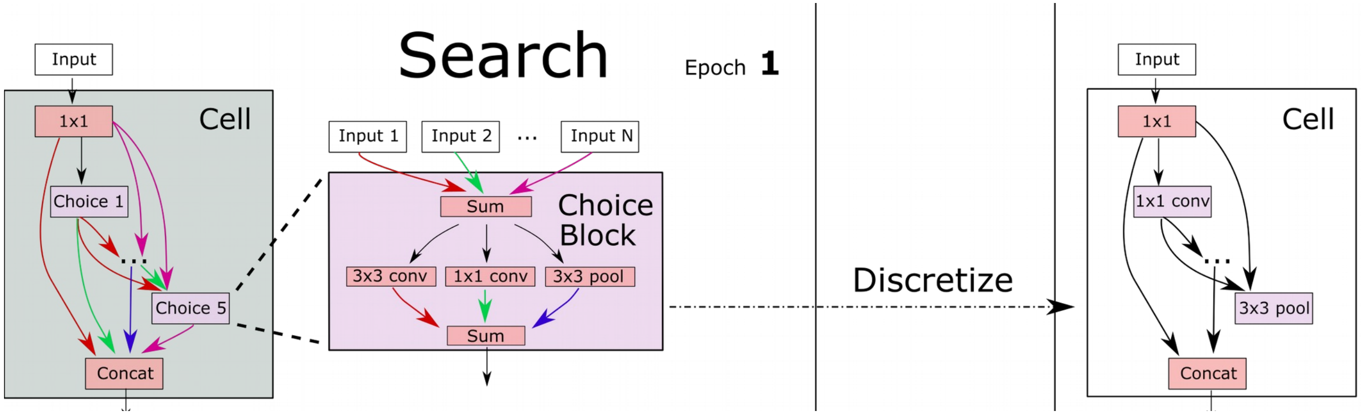


# Idea

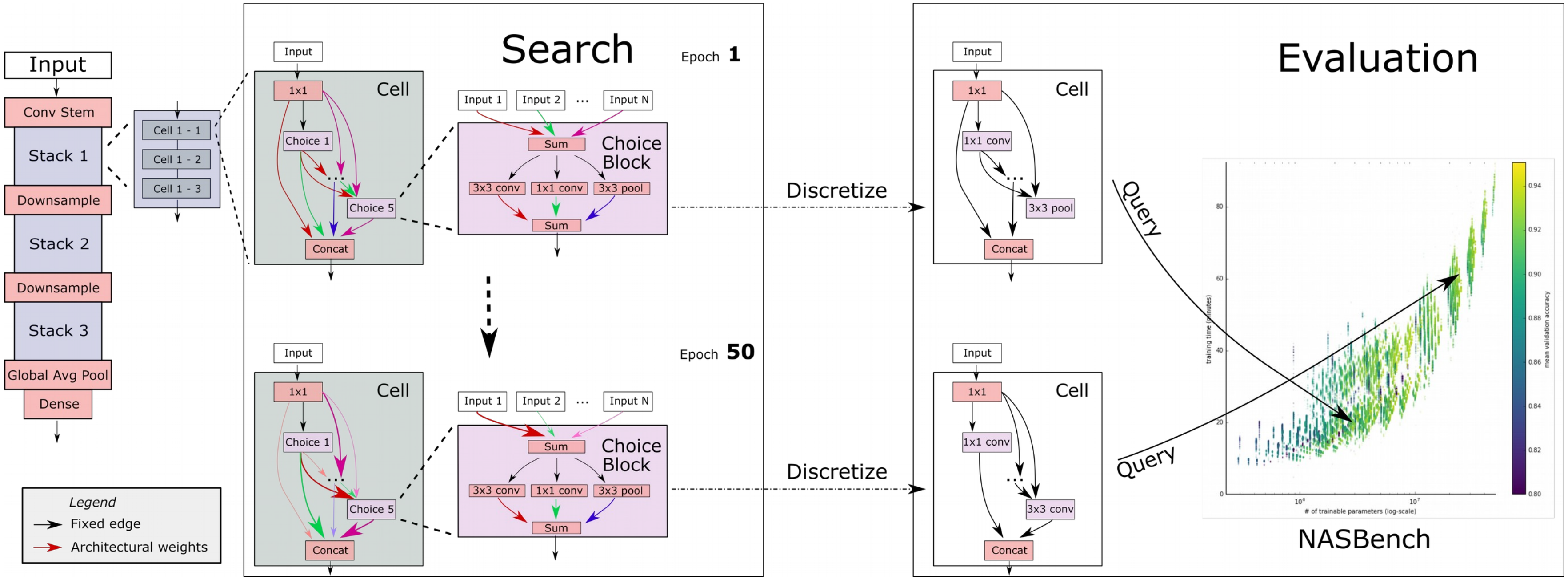
- Define search spaces by number of parents of each node:

Table 1: Characteristic information of the search spaces.

| | | Search space | | |
|---|---|---|---|---|
| | | 1 | 2 | 3 |
| No. parents | Node 1 | 1 | 1 | 1 |
| | Node 2 | 2 | 1 | 1 |
| | Node 3 | 2 | 2 | 1 |
| | Node 4 | 2 | 2 | 2 |
| | Node 5 | - | - | 2 |
| | Output | 2 | 3 | 2 |
| No. archs. | w/ loose ends | 6240 | 29160 | 363648 |
| | w/o loose ends | 2487 | 3609 | 24066 |



# Idea




This allowed the following **analysis**:

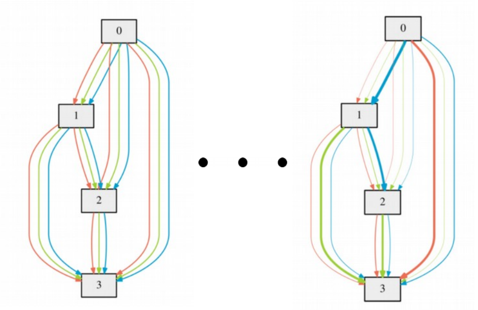
- Follow architecture trajectory of One-Shot NAS
    - **Comparison** of 4 One-shot NAS optimizers
- **Correlation** between One-shot validation error and NASBench validation error
- ***Hyperparameter Optimization*** of search methods.

# Outline

- ✓ Idea
- **One-Shot NAS Optimizers**
- Results
- Conclusion

# One-Shot NAS Optimizers

*DARTS* *[Liu et al. 18]*




*PC- DARTS* *[Xu et al. 19]*

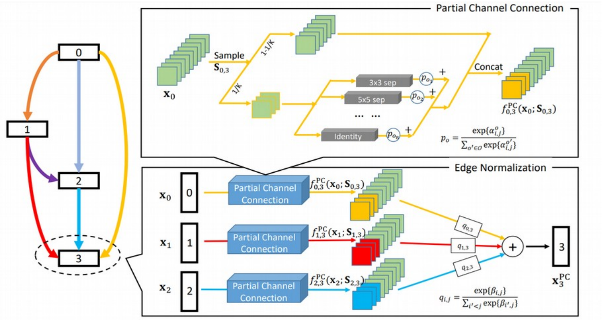



Figure from *Xu, Yuhui, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. "PC-DARTS: Partial Channel Connections for Memory-Efficient Differentiable Architecture Search." (2019).*

**Discrete optimizers:**
- BOHB
- Hyperband
- Random Search
- Regularized Evolution
- SMAC
- TPE
- Reinforce

**More optimizers to be done …**

*GDAS* *[Dong et al. 19]*

- Differentiably sample paths through each cell.
  - Only operations on path need to be evaluated
    - Very fast search
- Avoids co-adaption

*Random Search with Weight Sharing* *[Li et al. 19]*

- *Training:*
  - Sample architecture from search space for each batch and train one-shot model weights.
- *Evaluation:*
  - Sample many archs., rank according to one-shot validation error of 10 batches
  - Fully evaluate top-10 archs.

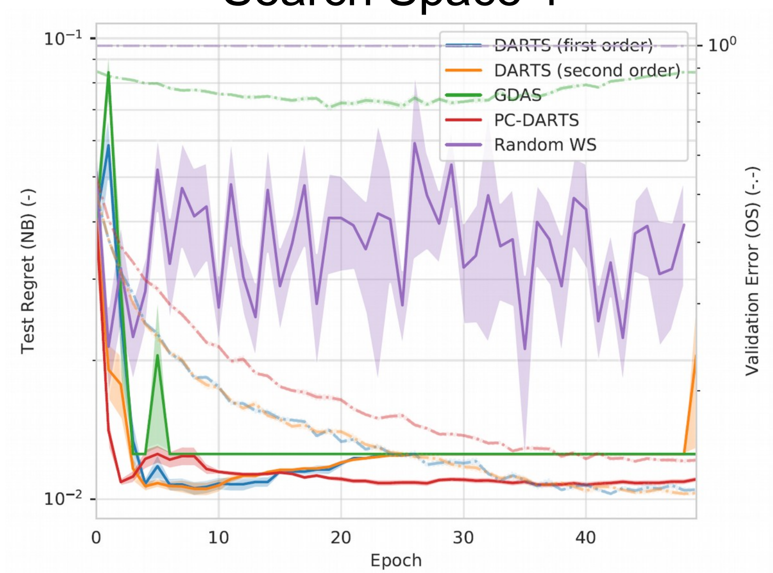
# Outline

- ✓ Idea
- ✓ One-Shot NAS Optimizers
- **Results**
  - NASBench 1-Shot-1 Analysis
  - NASBench 1-Shot-1 HPO
- Conclusion

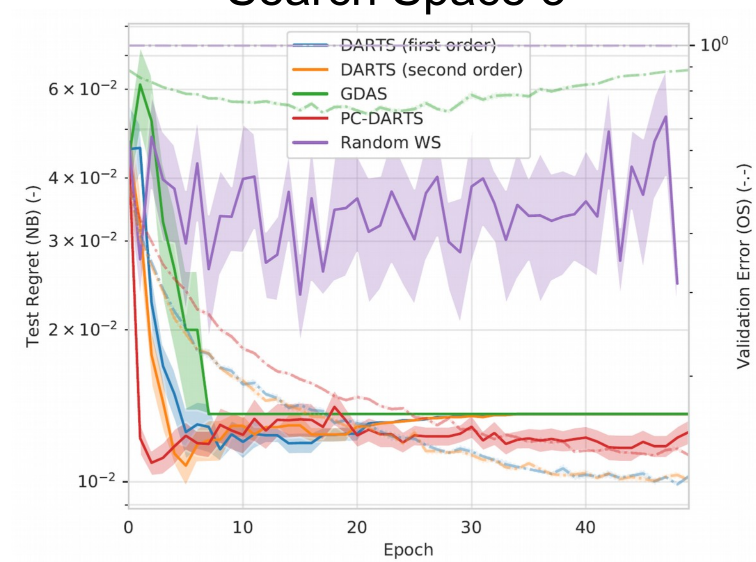
# NAS-Bench-1Shot1 as Analysis Framework

## Optimizer Comparison

Search Space 1



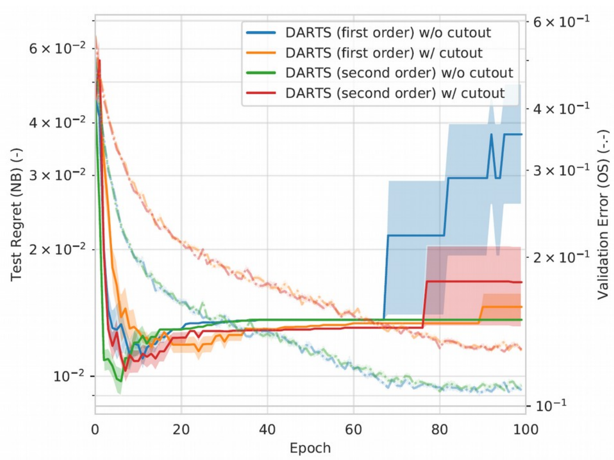

Search Space 3

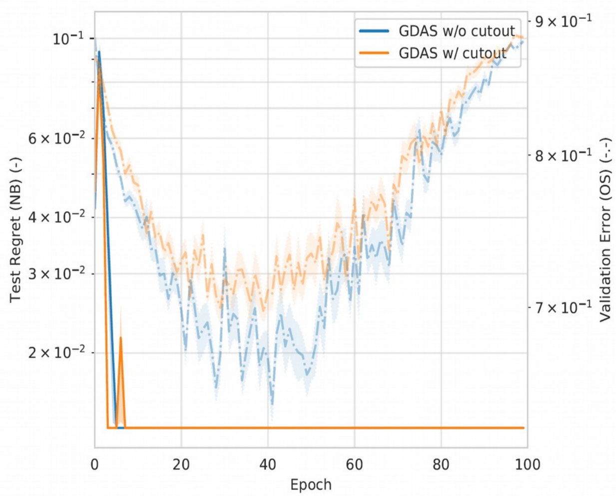


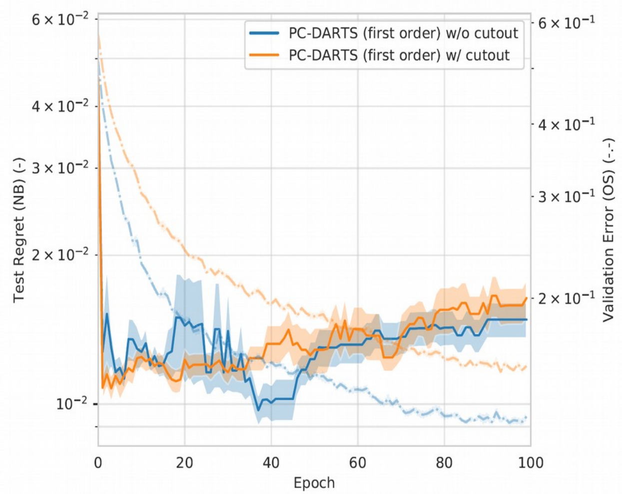

- ***DARTS and GDAS:***
  - stuck in local optimum
- ***PC-DARTS:***
  - stable search and relatively good performance for the given number of epochs
- ***Random Search with WS:***
  - explores mainly poor architectures

# NAS-Bench-1Shot1 as Analysis Framework

## Regularized Search (Cutout) – Search Space 3




### DARTS

- Longer search -> architectural overfitting
- Cutout largely stabilized the search

### GDAS

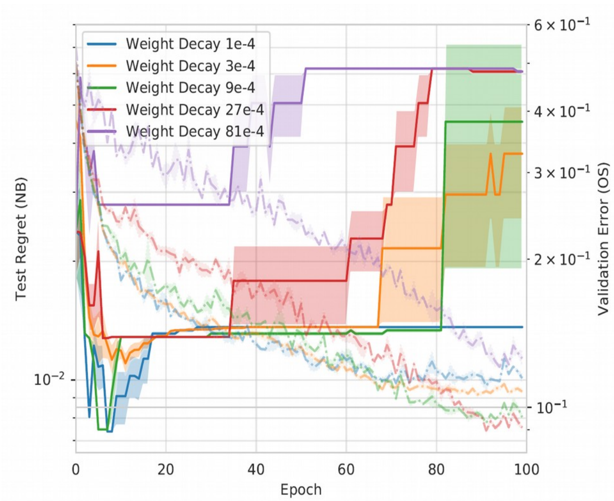
- Little impact of cutout on found architectures.

### PC-DARTS

- Additional regularization has no positive impact

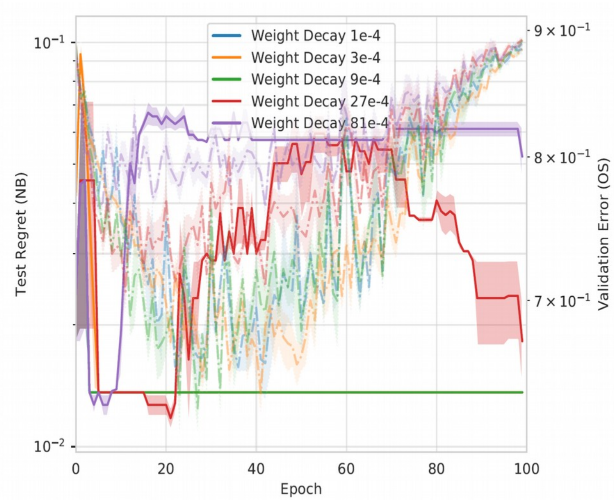
# NAS-Bench-1Shot1 as Analysis Framework

## Regularized Search (Weight Decay) – Search Space 3



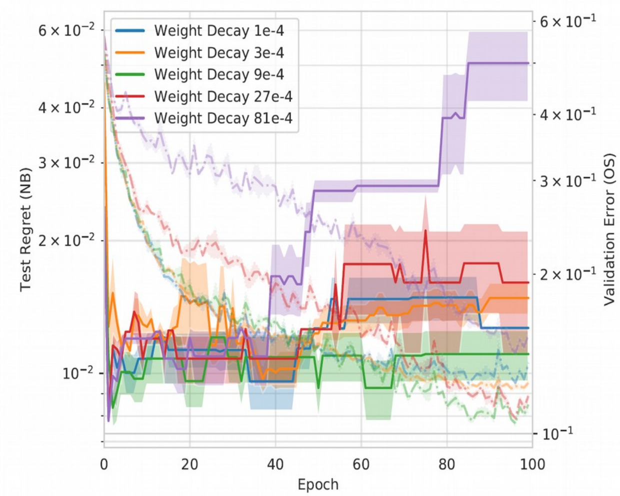

DARTS

Higher regularization -> less stable search




GDAS

Higher regularization -> less stable search

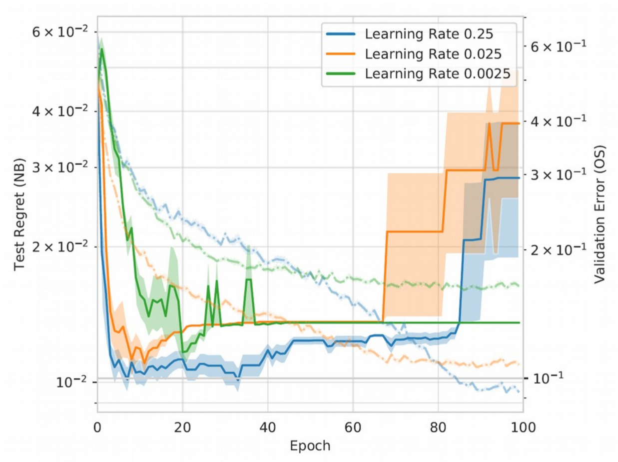



PC-DARTS

High regularization -> less stable search

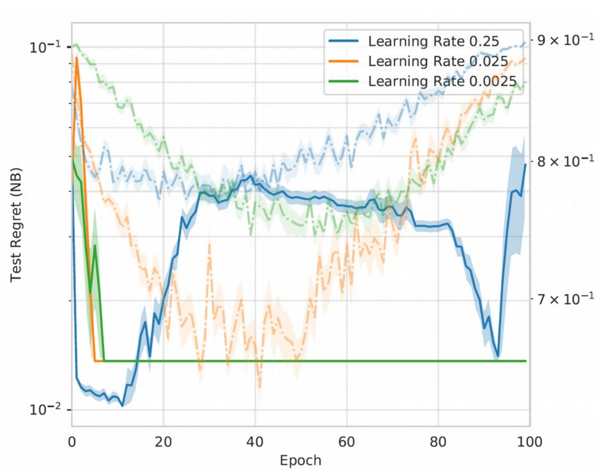
# NAS-Bench-1Shot1 as Analysis Framework

## Effect of one-shot learning rate – Search Space 3



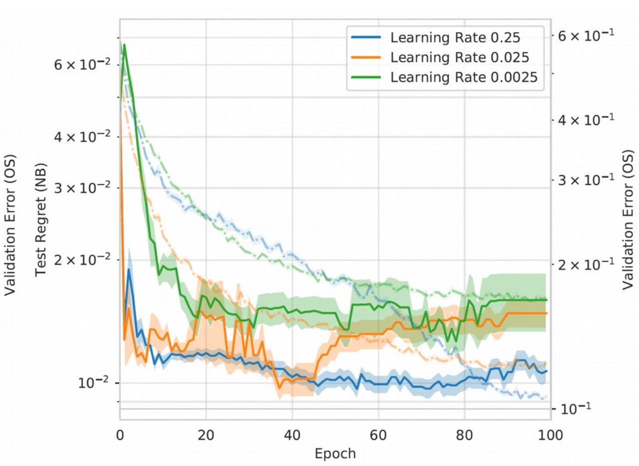

DARTS

High learning-rate -> less stable search




GDAS

High learning-rate -> less stable search

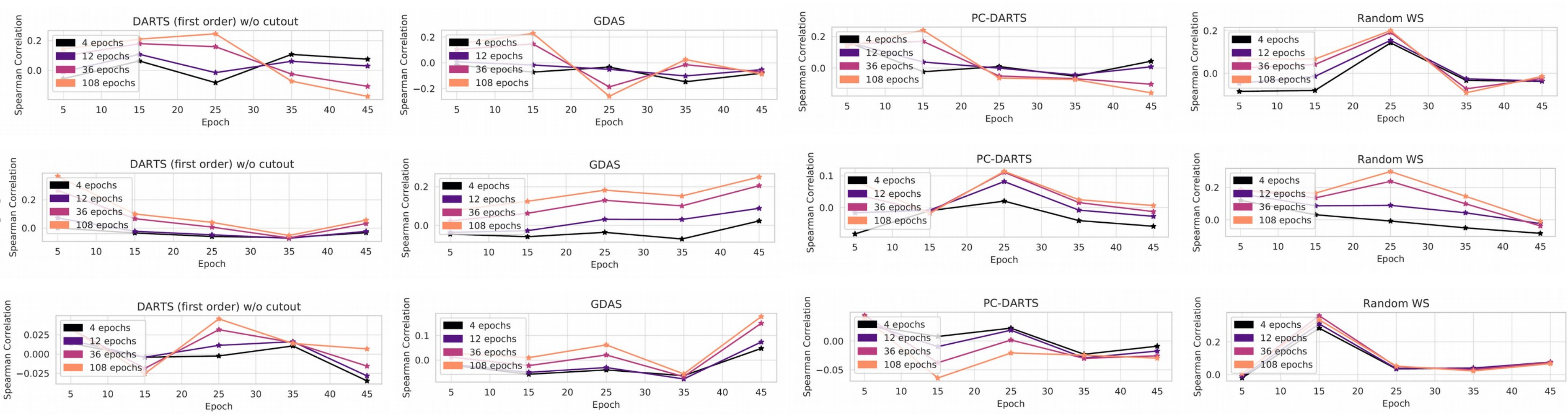



PC-DARTS

High learning-rate -> **better** search

## Correlation

1
2
3

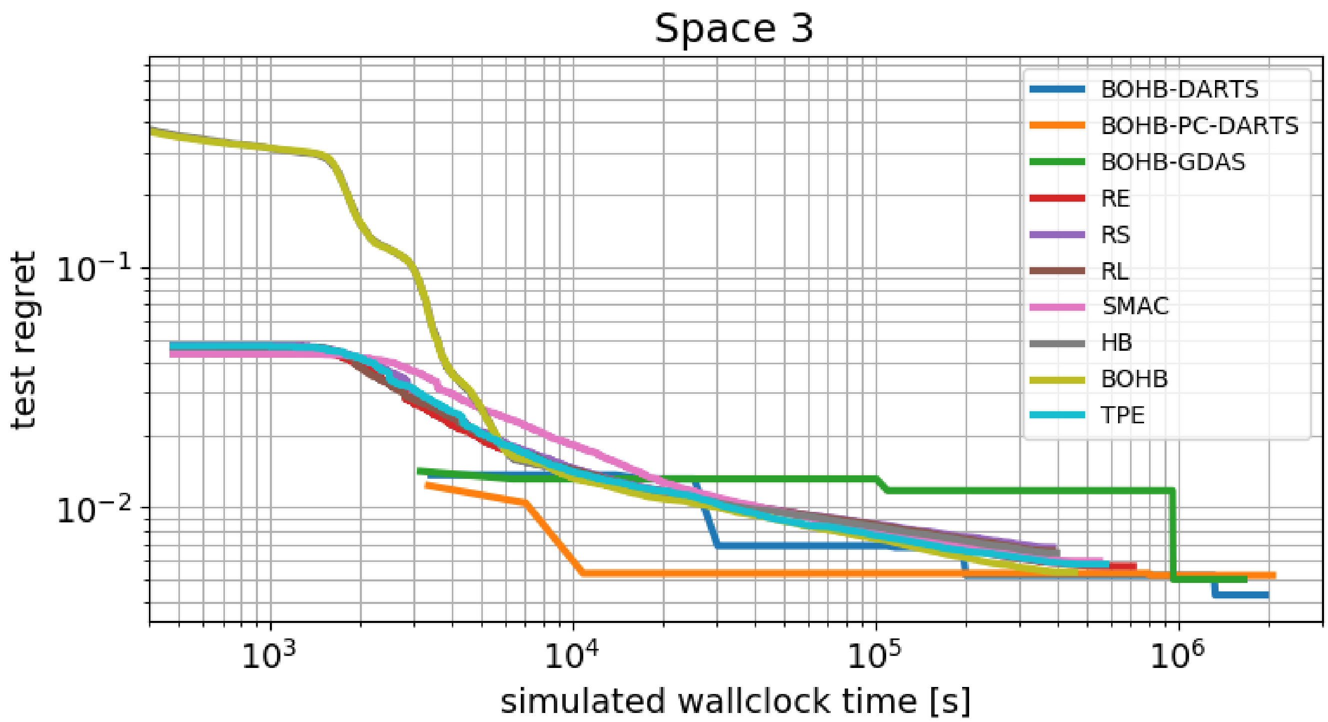



DARTS GDAS PC-DARTS Random-WS

- **No correlation** between one-shot validation error and NASBench validation error:
  - For all one-shot search methods
  - For all search spaces
- Follows results by Sciuto et al. 19: They only estimated using 32 architectures

# Tunability of NAS optimizers

Optimize the hyperparameters of one-shot NAS optimizers using BOHB [Falkner et al. 2018]




- Outperform the default configuration by a factor of 7-10
- With the same number of function evaluations, they are able to outperform black-box NAS optimizers

# Conclusion and Future Directions

- We presented NAS-Bench-1Shot1, a framework containing 3 benchmarks that enable to evaluate the **anytime performance** of one-shot NAS algorithms
- NAS-Bench-1Shot1 as **analysis framework**
- One-shot NAS optimizers can outperform black-box optimizers if **tuned properly**

**Future work:**

- Add other methods such as ENAS [Pham et al. 2018], ProxylessNAS [Cai et al. 2019], etc.
- Automate the generation of plots, analysis results, or benchmark tables.
- Towards NAS-Bench-201