Stronger NAS with Weaker Predictors

Junru Wu, Xiyang Dai, Dongdong Chen, Yinpeng Chen, Mengchen Liu, Ye Yu, Zhangyang Wang, Zicheng Liu, Mei Chen, Lu Yuan

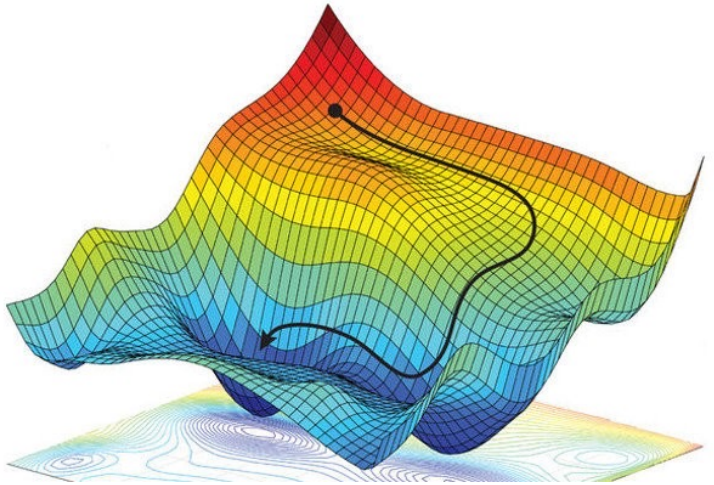
TEXAS
The University of Texas at Austin

Neural Architecture Search (NAS)

- **Objective of Neural Architecture Search**

Find the neural architecture x^* with the highest performance $f(x)$ given the search space X

$$x^* = \underset{x \in X}{\operatorname{argmax}} f(x)$$



$f: X \rightarrow P$

Previous NAS Approaches

- **A naïve solution**
  - Estimate the performance mapping $f(x)$ through the full search space
    - *prohibitively expensive*
- **Predictor-based NAS [1]**
  - Learns a proxy predictor $\tilde{f}(x)$ to approximate $f(x)$ by sampling some architecture-performance pairs
    - *significantly reduces the training cost.*
  - In general, predictor-based NAS can be re-cast as a bi-level optimization problem:

$$x^* = \underset{x \in X}{\operatorname{argmax}} \tilde{f}(x|S),\ s.t.\ \tilde{f} = \underset{S, \tilde{f} \in \tilde{F}}{\operatorname{argmin}} \sum_{s \in S} L(\tilde{f}(s), f(s))$$

[1] Wei Wen, Hanxiao Liu, Hai Li, Yiran Chen, Gabriel Bender, Pieter-Jan Kindermans "Neural Predictor for Neural Architecture Search." *In European Conference on Computer Vision (ECCV), 2020.*

Previous: Predictor-based NAS

Sampling stage:

$$S \subset X$$

Learning stage:

$$x^* = \arg\max_{x \in X} \tilde{f}(x|S)$$

$$\text{s.t. } \tilde{f} = \arg\min_{S, \tilde{f} \in \tilde{\mathcal{F}}} \sum_{s \in S} \mathcal{L}(\tilde{f}(s), f(s))$$

Proposed: WeakNAS

Sampling stage:

$$\tilde{P}^k = \{\tilde{f}_k(s) | s \in X \setminus S^k\}$$

$$S^{k+1} = \{s_i\}_{i=1}^{M} \cup S^k,$$

$$\text{where } \tilde{f}_k(s_i) \in \text{Top}_N(\tilde{P}^k),$$

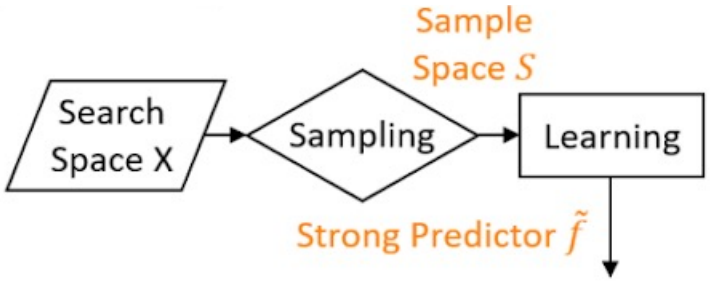
Learning stage:

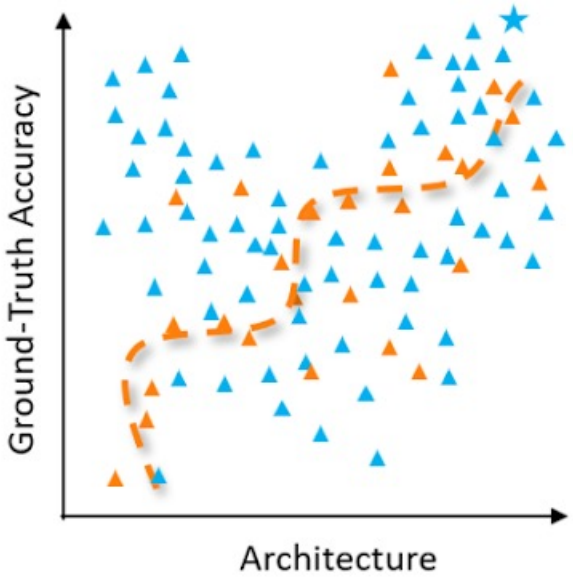
$$x^* = \arg\max_{x \in X} \tilde{f}(x|S^{k+1})$$

$$\text{s.t. } \tilde{f}_{k+1} = \arg\min_{\tilde{f}_k \in \tilde{\mathcal{F}}} \sum_{s \in S^{k+1}} \mathcal{L}(\tilde{f}(s), f(s))$$

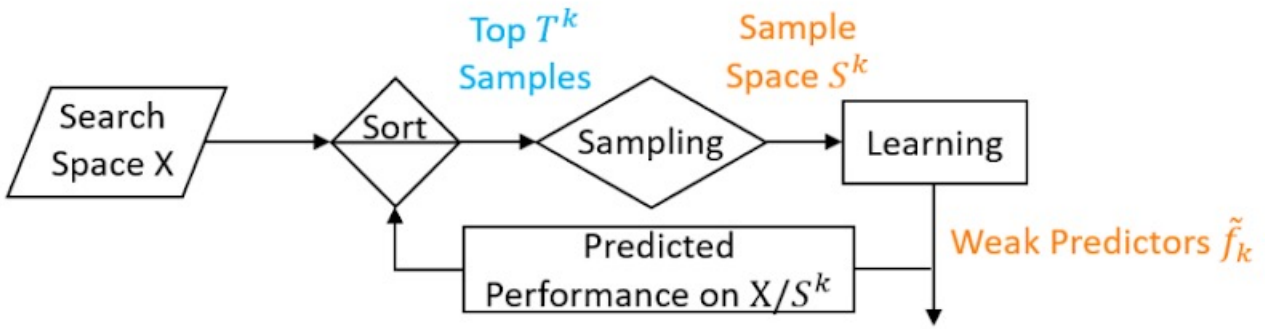
WeakNAS ***jointly evolve*** the Sampling stage & Learning stage

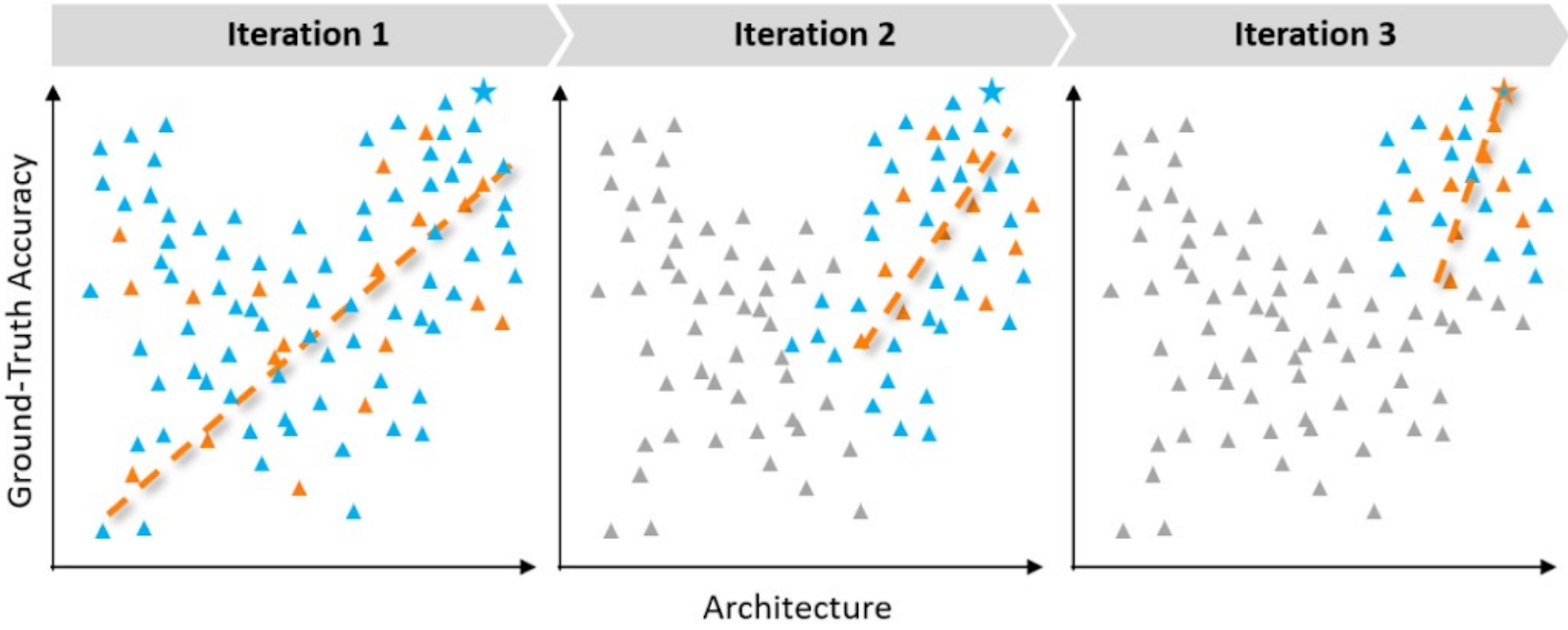
Previous: Predictor-based NAS





Proposed: WeakNAS



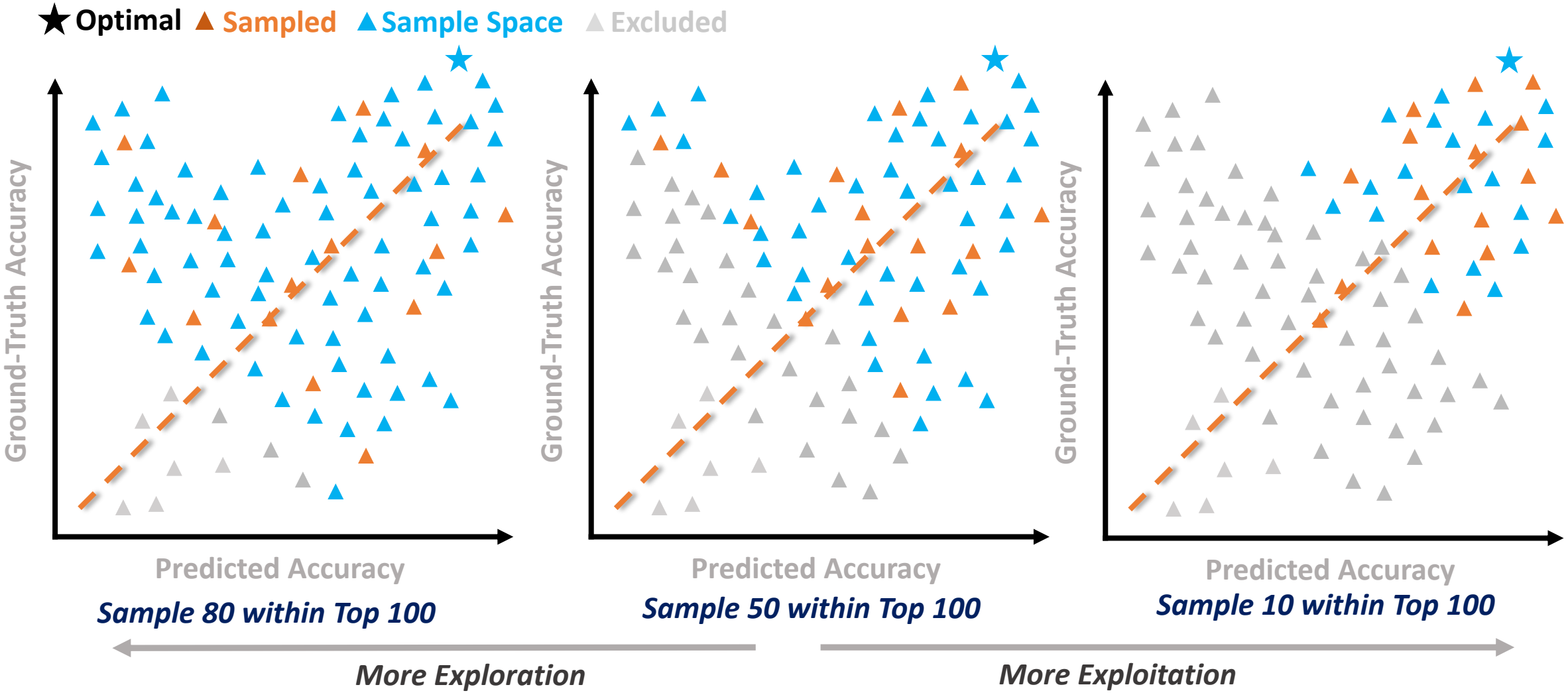


★ Optimal ▲ New Sampled ▲ Sample Space ▲ Excluded

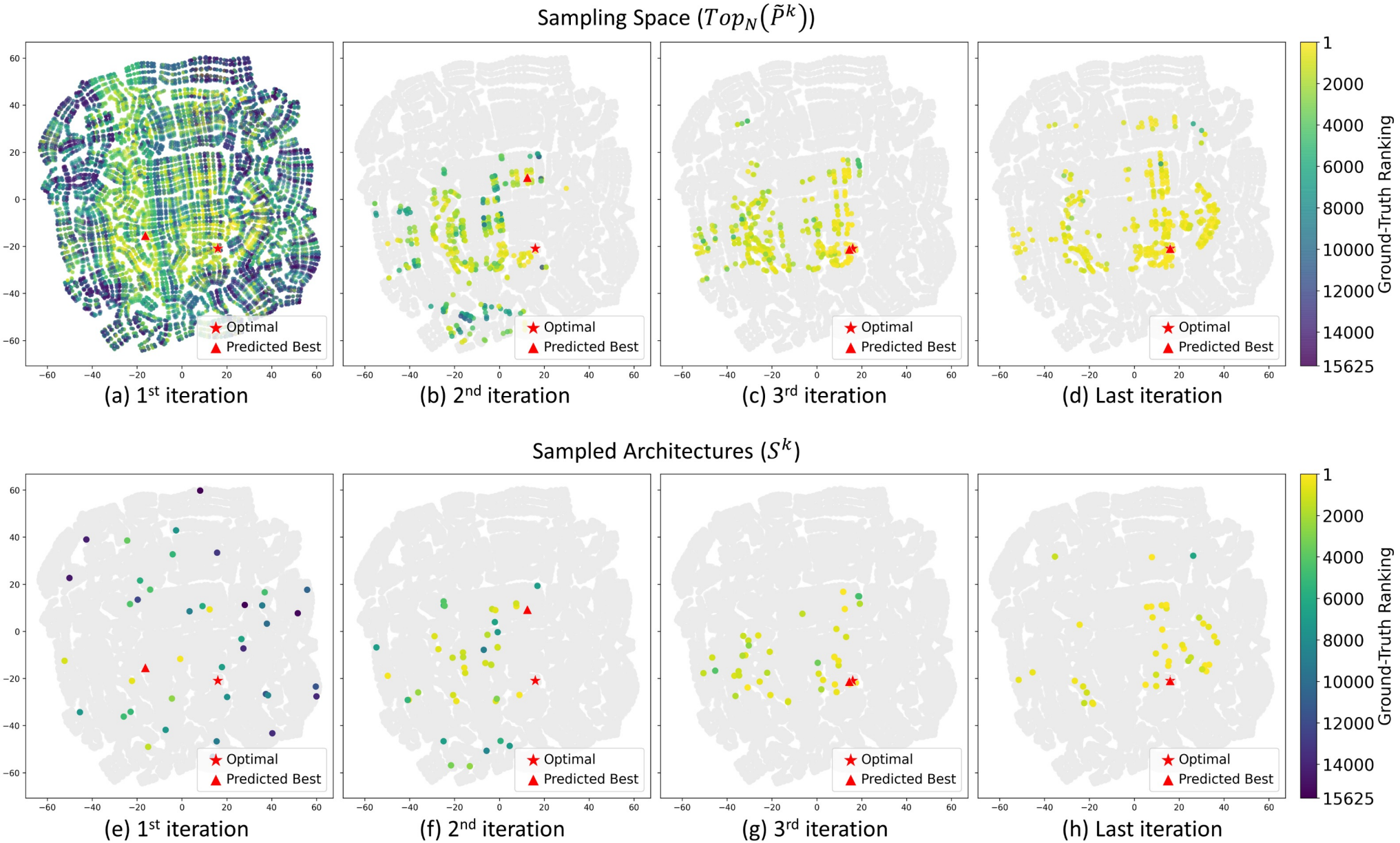
Exploitation-exploration trade-off

Uniformly sample M samples within Top N predictions by $\tilde{f}$

$\boldsymbol{\varepsilon = M/N}$ control the Exploitation-exploration trade-off

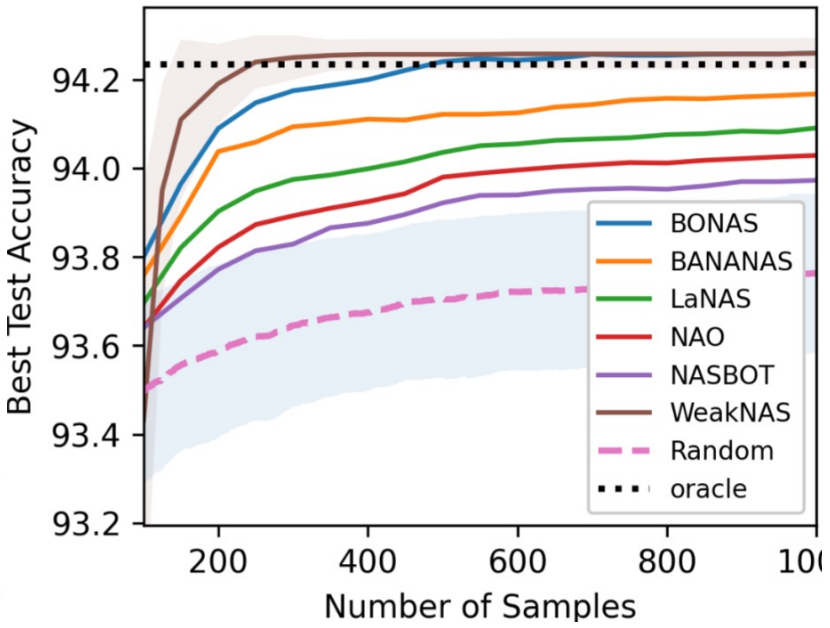


Search Dynamics (t-SNE Visualization)



Comparsion to SoTA on NAS-Bench-101

Method	#Queries	Test Acc.(%)	SD(%)	Test Regret(%)	Avg. Rank
Random Search	2000	93.64	0.25	0.68	1750.0
NAO [2]	2000	93.90	0.03	0.42	168.1
Reg Evolution [14]	2000	93.96	0.05	0.36	85.0
Semi-NAS [20]	2000	94.02	0.05	0.30	42.1
Neural Predictor [7]	2000	94.04	0.05	0.28	33.5
WeakNAS	**2000**	**94.26**	**0.04**	**0.06**	**1.6**
Semi-Assessor [42]	1000	94.01	-	0.31	47.1
LaNAS [21]	1000	94.10	-	0.22	14.1
BONAS [19]	1000	94.22	-	0.10	3.0
WeakNAS	**1000**	**94.25**	**0.04**	**0.07**	**1.7**
arch2vec [41]	400	94.10	-	0.22	14.1
WeakNAS	**400**	**94.24**	**0.04**	**0.08**	**1.9**
LaNAS [21]	200	93.90	-	0.42	168.1
BONAS [19]	200	94.09	-	0.23	18.0
WeakNAS	**200**	**94.18**	**0.14**	**0.14**	**5.6**
Optimal	-	94.32	-	0.00	1.0



Comparsion to SoTA on ImageNet (MobileNet Search Space)

Model	Queries(#)	Top-1 Acc.(%)	Top-5 Acc.(%)	FLOPs(M)
Proxyless NAS[47]	-	75.1	92.9	-
Semi-NAS[20]	300	76.5	93.2	599
BigNAS[42]	-	76.5	-	586
FBNetv3[43]	20000	80.5	95.1	557
OFA[36]	16000	80.0	-	595
LaNAS[21]	800	80.8	-	598
WeakNAS	**1000**	**81.3**	**95.1**	**560**
	800	**81.2**	**95.2**	**593**

Thanks!

Code: https://github.com/VITA-Group/WeakNAS