JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

PA&DA: Jointly Sampling PAth and DAta for Consistent NAS

Shun Lu[1,2], Yu Hu[1,2]*, Longxing Yang[1,2], Zihao Sun[1,2], Jilin Mei[1], Jianchao Tan[3], Chengru Song[3]
[1] Research Center for Intelligent Computing Systems,
Institute of Computing Technology, Chinese Academy of Sciences
[2] School of Computer Science and Technology, University of Chinese Academy of Sciences
[3] Kuaishou Technology

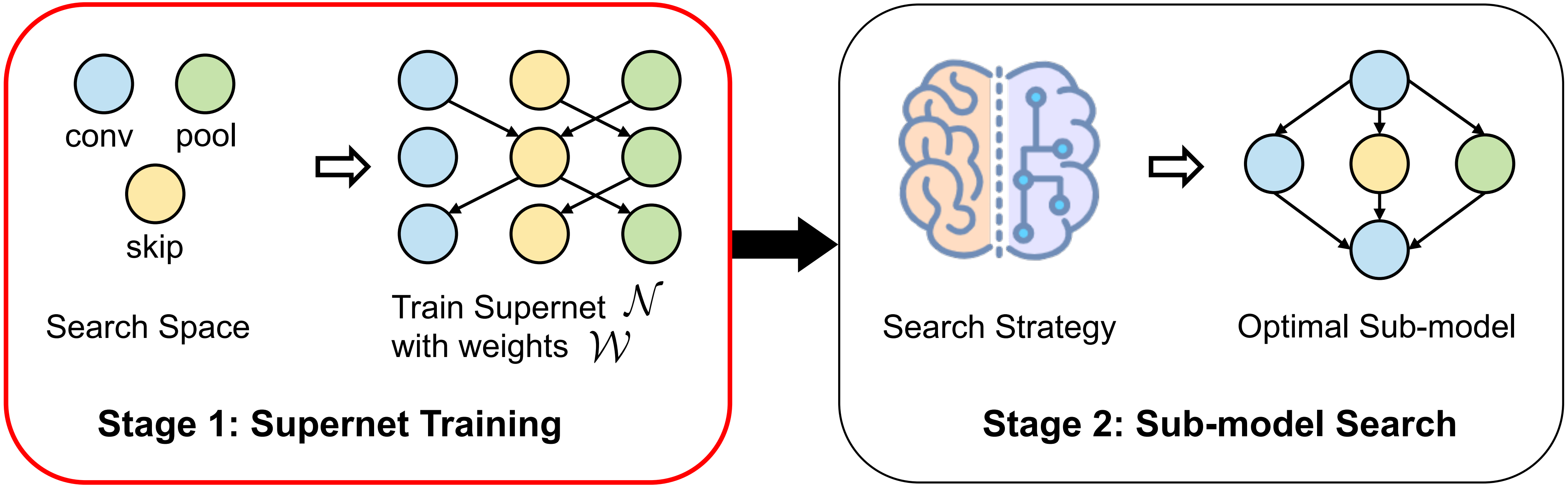
Paper: https://arxiv.org/abs/2302.14772

Code: https://github.com/ShunLu91/PA-DA

Tag: WED-AM-354

1 Background & Motivation

1.1 One-shot Neural Architecture Search (NAS)

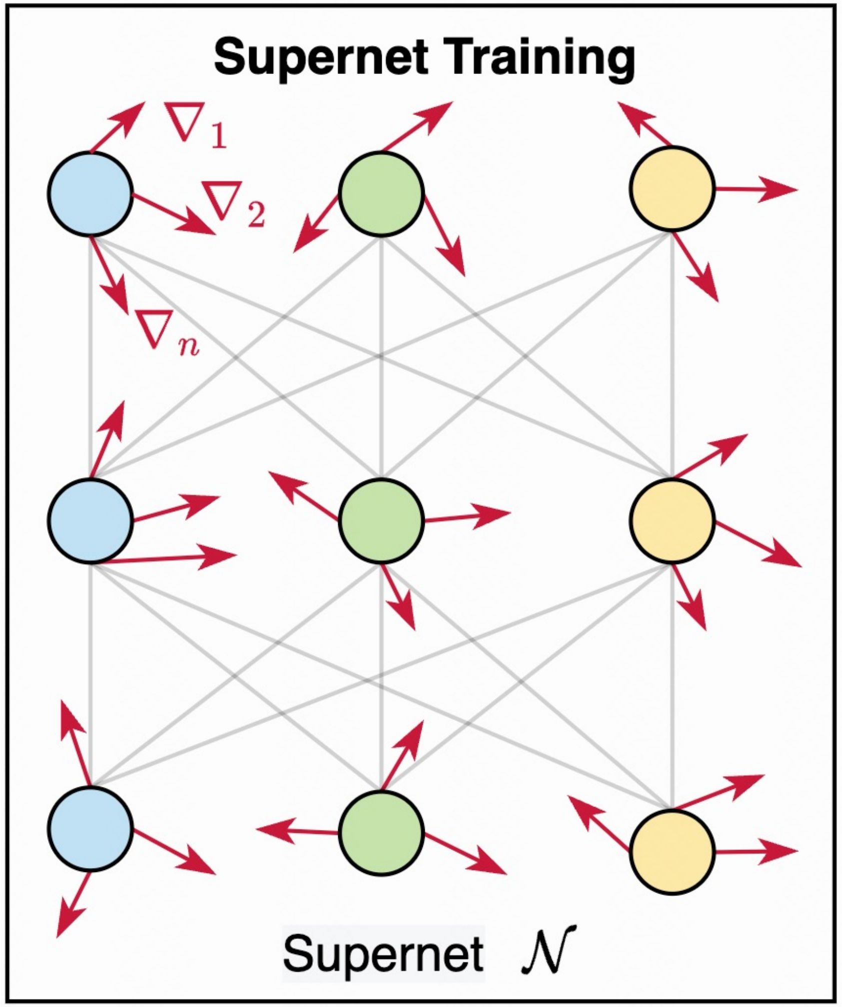


1 Background & Motivation

智能计算机研究中心
Research Center for Intelligent Computing Systems

1.2 Problem in One-Shot NAS

- **Prombel** : shared weights suffer from different gradient descent directions
- **Available solutions** :
  - Elaborate a better path sampling strategy
    - FairNAS [ICCV'21], Magic-AT [ICML'22]
  - Maintain multi-copies of supernet weights
    - Few-Shot-NAS [ICML'21], GM [ICLR'22], CLOSE [ECCV'22]
  - Introduce additional loss regularizations
    - NSAS [CVPR'20] , SUMNAS [ICLR'22], Magic-AT [ICML'22]
  - Drawbacks : require multiple computation burdens and obtain unsatisfying results
- Motivate us to explore a better solution.

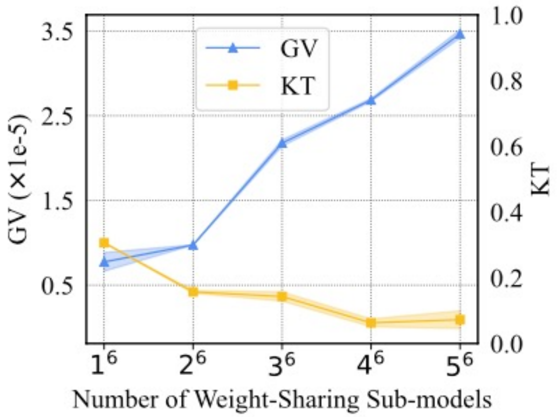


ANS@ICT

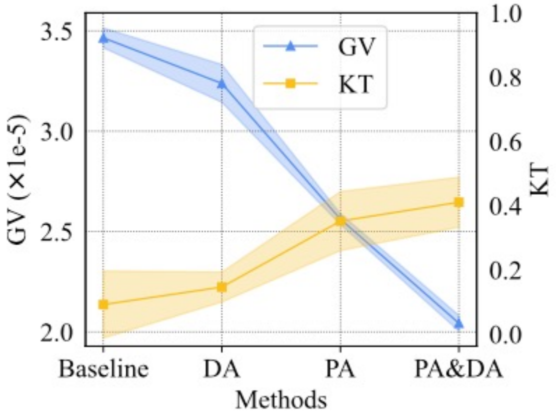
2 Method-PA&DA

2.1 Observations

- **Kendall's Tau (KT)**: we calculate the correlation between predicted scores and ground-truth scores of sub-models, to indicate the ranking consistency of sub-models.
- **Gradient Variance (GV)**: we record the average GV of all candidate operation weights during training



(a) Different weight-sharing extent



(b) Different methods

  - With more sub-models sharing weights, GV increases and KT becomes worse
  - When using different methods, GV decreases and KT becomes better
- **Prompts**: reduce GV to improve the ranking consistency KT.

2 Method-PA&DA

2.1 Observations

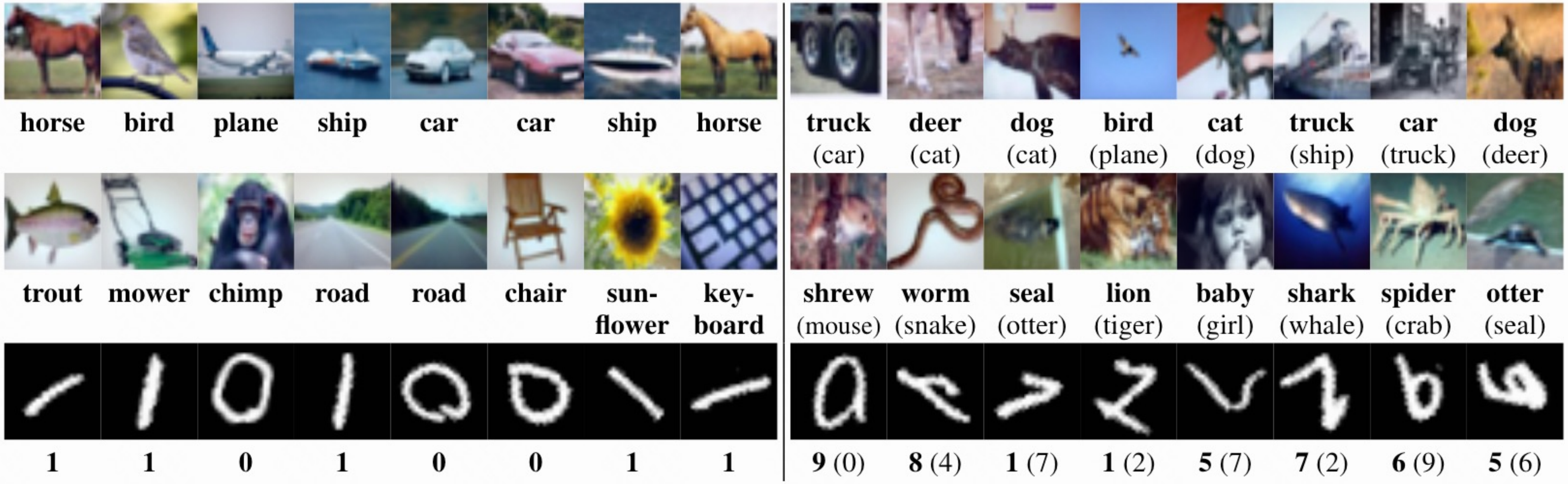
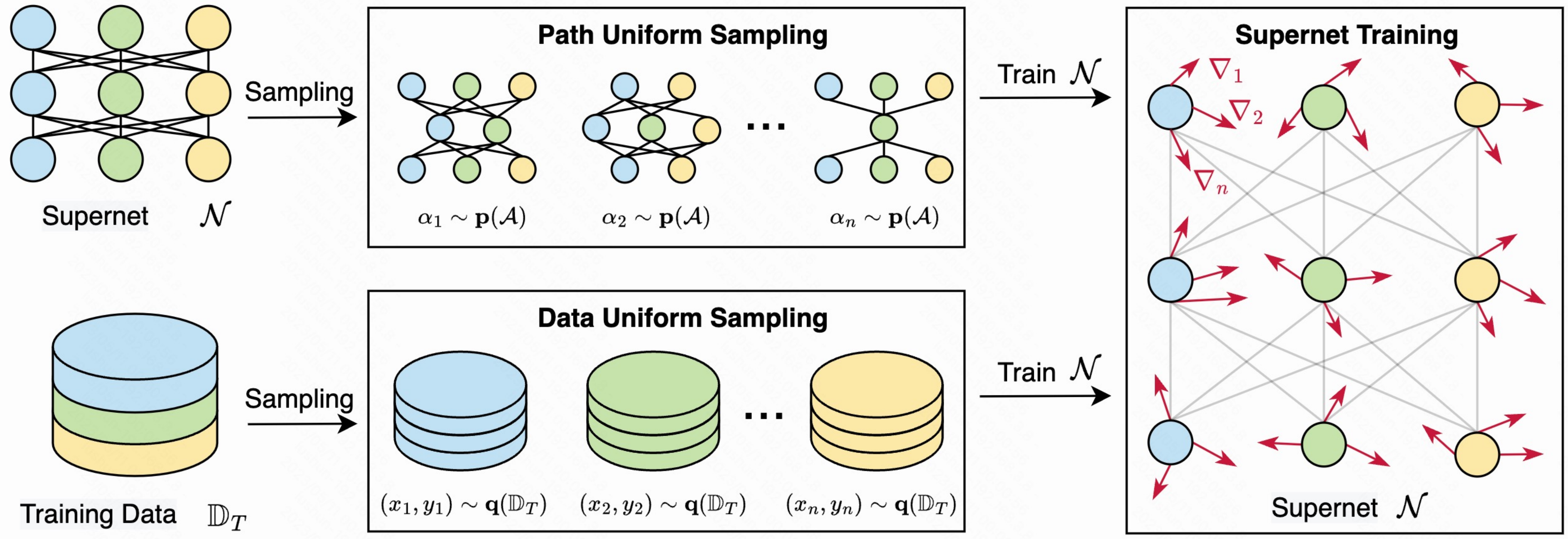


Figure 1: Nonpriority and priority training examples for image classification. *Left:* Examples that RAIS samples infrequently during training. *Right:* Examples that RAIS prioritizes. Bold denotes the image's label. Parentheses denote a different class that the model considers likely during training. Datasets are CIFAR-10 (top), CIFAR-100 (middle), and rotated MNIST (bottom).

- Inspiration from RAIS [NeurIPS'18]: better data sampling strategy can reduce the gradient variance of model training, thereby improving the generalization of the model.

智能计算机研究中心
Research Center for Intelligent Computing Systems

2.2 Traditional Sampling-based One-Shot NAS



- Stage 1: Supernet Training

$$\mathcal{W}^\star = \underset{\mathcal{W}}{\operatorname{argmin}} \, \mathbb{E}_{\substack{\alpha \sim \mathbf{p}(\mathcal{A}) \\ (x,y) \sim \mathbf{q}(\mathbb{D}_T)}} \left[\mathcal{L}(\mathcal{N}(x, \alpha; \mathcal{W}_\alpha), y) \right] \quad (1)$$

- Stage 2: Sub-model Search

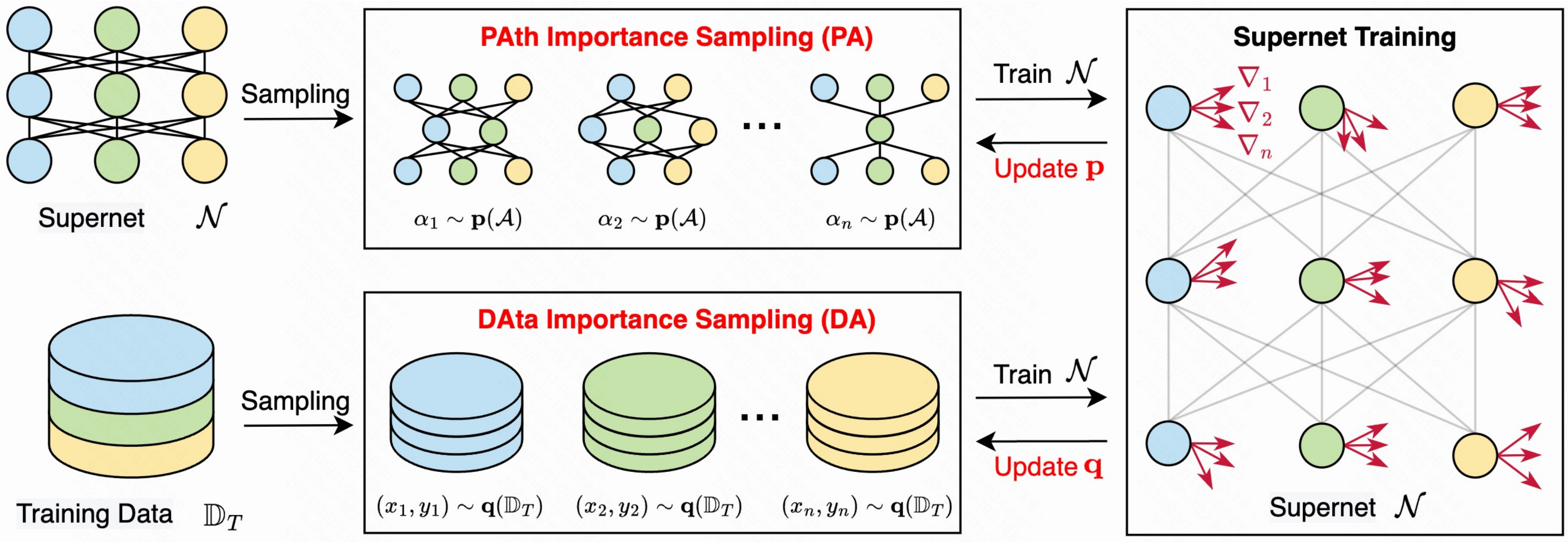
$$\alpha^\star = \underset{\alpha \in \mathcal{A}}{\operatorname{argmax}} \, \mathbb{E}_{(x,y) \sim \mathbf{q}(\mathbb{D}_V)} \left[\mathcal{P}(\mathcal{N}(x, \alpha; \mathcal{W}^\star_\alpha), y) \right] \quad (2)$$

ANS@ICT

2 Method-PA&DA

智能计算机研究中心
Research Center for Intelligent Computing Systems

2.3 Importance Sampling One-Shot NAS



- Formulation of our objective: jointly optimize path and data sampling distribution during supernet training.

$$
\mathcal{W}^\star = \underset{\mathcal{W}}{\operatorname{argmin}}\, \mathbb{E}[\mathcal{L}(\mathcal{N}(x, \alpha; \mathcal{W}_\alpha), y)]
$$
$$
\text{s.t.} \begin{cases} \alpha \sim \mathbf{p}^\star(\mathcal{A}),\ (x, y) \sim \mathbf{q}^\star(\mathbb{D}_T), \\ \mathbf{p}^\star = \underset{\mathbf{p}}{\operatorname{argmin}} \mathbb{V}[d(\mathbf{p})], \\ \mathbf{q}^\star = \underset{\mathbf{q}}{\operatorname{argmin}} \mathbb{V}[d(\mathbf{q})] \end{cases} \tag{3}
$$

ANS@ICT

2 Method-PA&DA

2.3 Importance Sampling One-Shot NAS

- Supernet training in sampling-based one-shot NAS :

$$\mathcal{W}^{\star} = \underset{\mathcal{W}}{\operatorname{argmin}} \, \mathbb{E}_{\substack{\alpha \sim \mathbf{p}(\mathcal{A}) \\ (x,y) \sim \mathbf{q}(\mathbb{D}_T)}} [\mathcal{L}(\mathcal{N}(x, \alpha; \mathcal{W}_{\alpha}), y)] \qquad (1)$$

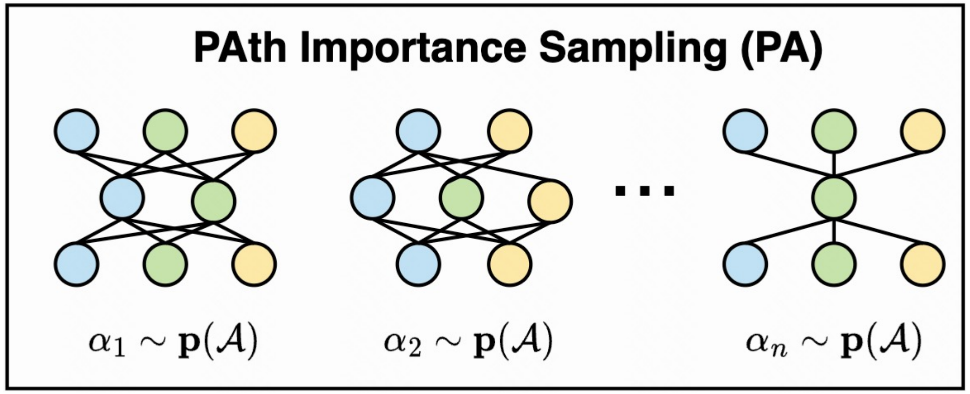
- PA&DA-Jointly optimize path and data sampling distribution during training:

$$\mathcal{W}^{\star} = \underset{\mathcal{W}}{\operatorname{argmin}} \, \mathbb{E}[\mathcal{L}(\mathcal{N}(x, \alpha; \mathcal{W}_{\alpha}), y)]$$

$$\text{s.t.} \begin{cases} \alpha \sim \mathbf{p}^{\star}(\mathcal{A}), \ (x, y) \sim \mathbf{q}^{\star}(\mathbb{D}_T), \\ \mathbf{p}^{\star} = \underset{\mathbf{p}}{\operatorname{argmin}} \mathbb{V}[d(\mathbf{p})], \\ \mathbf{q}^{\star} = \underset{\mathbf{q}}{\operatorname{argmin}} \mathbb{V}[d(\mathbf{q})] \end{cases} \qquad (3)$$

2 Method-PA&DA

2.4 PAth Importance Sampling (PA)



- At i-th training step, the stochastic gradient is:

$$d_i(p_i) = \frac{1}{Np_i} \nabla_{\mathcal{W}} \mathcal{L}(\mathcal{N}(x_i, \alpha_i; \mathcal{W}_{\alpha_i}), y_i) \qquad (5)$$

- Introduce the gradient to our objective:

$$\min_{\mathbf{p}} \mathbb{V}[d(\mathbf{p})] = \mathbb{E}\left[d^\top d\right] - \mathbb{E}\left[d\right]^\top \mathbb{E}\left[d\right] \qquad (6)$$

- Reformulate the problem in Eq.6 as a constrained optimization problem:

$$\min_{\mathbf{p}} \sum_{i=1}^{N} \frac{1}{N^2} \frac{1}{p_i} \|\nabla_{\mathcal{W}} \mathcal{L}(\mathcal{N}(x_i, \alpha_i; \mathcal{W}_{\alpha_i}), y_i)\|^2$$
$$\text{s.t.} \sum_{i=1}^{N} p_i = 1 \quad \text{and} \quad p_i \geq 0 \quad \forall i = 1, 2, \ldots N$$

- Use the Lagrange multiplier method to solve the optimal path sampling distribution:

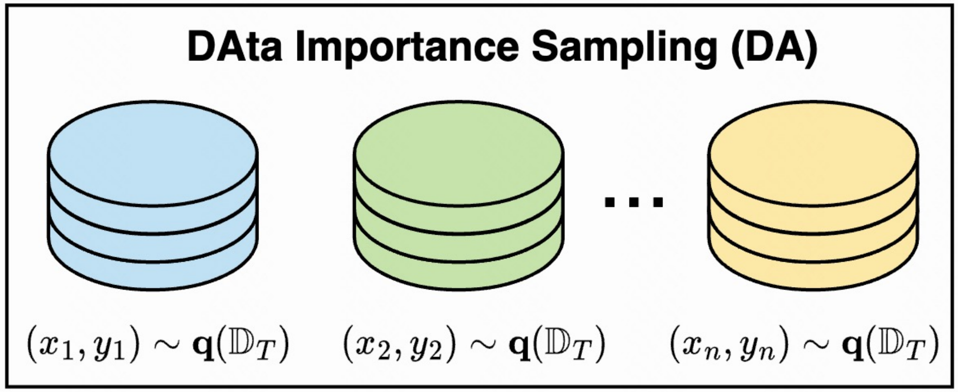
$$p_i^\star = \frac{\|\nabla_{\mathcal{W}} \mathcal{L}(\mathcal{N}(x_i, \alpha_i; \mathcal{W}_{\alpha_i}), y_i)\|}{\sum_{i=1}^{N} \|\nabla_{\mathcal{W}} \mathcal{L}(\mathcal{N}(x_i, \alpha_i; \mathcal{W}_{\alpha_i}), y_i)\|} \qquad (10)$$

- Conclusion: the optimal p_i^* is proportional to the normalized gradient norm of the sub-model.

2 Method-PA&DA

智能计算机研究中心
Research Center for Intelligent Computing Systems

2.5 DAta Importance Sampling (DA)



- According to previous works, the optimal data sampling distribution q_i^* is given by:

$$q_i^\star \propto \|\nabla_{\mathcal{W}} \mathcal{L}(\mathcal{N}(x_i, \alpha_i; \mathcal{W}_{\alpha_i}), y_i)\| \qquad (11)$$

- In mini-batch training, it is time-consuming and laborious to compute per-sample gradient norm. Thereby we use the Upper-bound [ICML'18] method to approximate:

$$\sup\{\|\nabla_{\mathcal{W}} \mathcal{L}(\mathcal{N}(x_i, \alpha_i; \mathcal{W}_{\alpha_i}), y_i)\|\} \leq \|\nabla_L\| \qquad (12)$$

- For image classification with a cross-entropy loss, the approximated upper bound is:

$$\nabla_L = \mathrm{softmax}(y_L) - \mathbb{1}(y_i) \qquad (13)$$

- In this way, we can efficiently approximate the gradient norm of each sample via a batch-wise mannar.

ANS@ICT

2 Method-PA&DA

2.6 Supernet training in practice

Algorithm 1 Supernet training algorithm of PA&DA

Input: Input training data $\mathbb{D}_T$, supernet $\mathcal{N}$ with weights $\mathcal{W}$, training epochs n_{epochs}, training steps n_{steps} per epoch.

Output: Optimized supernet weights $\mathcal{W}^{\star}$.

```
for j = 1 to n_epochs do
    for k = 1 to n_steps do
        Sample a path based on the distribution p(A);
        Sample a mini-batch training data based on the distribution q(D_T);
        Train supernet weights W by gradient descent;
        Record gradient norm of the sampled path after back-propagation;
        Approximate and record gradient norm of the sampled data using Eq.13.
    end for
    Linearly increase smoothing parameters δ and τ;
    Update the path sampling distribution p(A) according to Eq.10 and add it to uniform distribution;
    Update the data sampling distribution q(D_T) according to Eq.11 and add it to uniform distribution;
end for
```

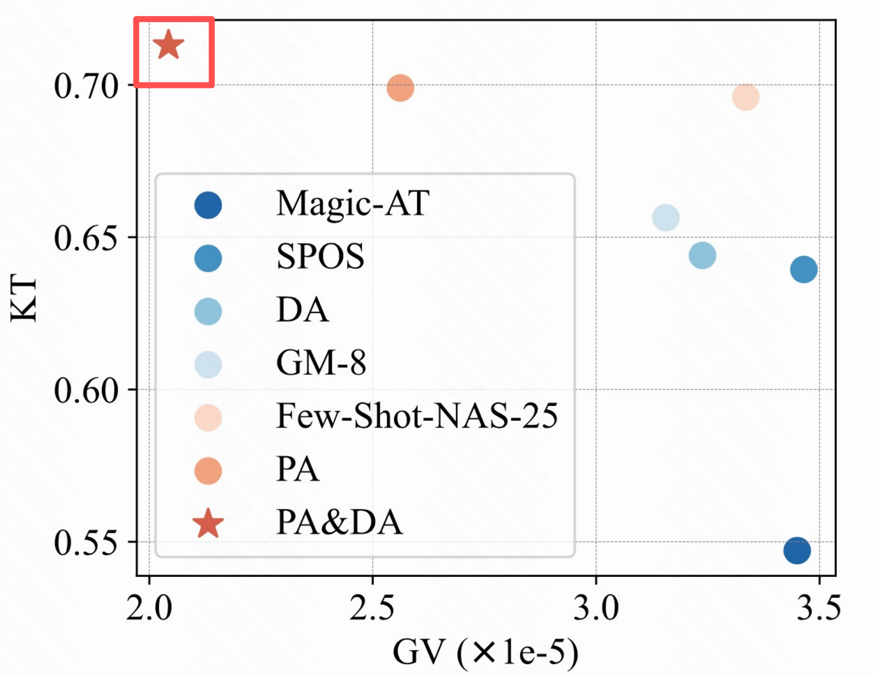
- **Path importance sampling**:
  - Update the path sampling distribution after each epoch.
  - To handle those parameter-free operations, employ a smoothing parameter δ to add path importance sampling distribution and the uniform sampling distribution together.
- **Data importance sampling**:
  - Update the data sampling distribution after each epoch.
  - To handle those data not sampled in the current epoch, employ a smoothing parameter τ to add data importance sampling distribution and the uniform sampling distribution.

3 Experiments

智能计算机研究中心
Research Center for Intelligent Computing Systems

3.1 Ranking Consistency in NAS-Bench-201 using CIFAR-10

Method	Cost	KT	P@Top5%
SPOS [16]	**1.6**	0.639 ± 0.030	0.211 ± 0.168
FairNAS† [7]	5.4	0.541 ± 0.023	0.160 ± 0.034
Magic-AT† [46]	4.4	0.547 ± 0.059	0.019 ± 0.011
NSAS [48]	14.6	0.653 ± 0.051	0.064 ± 0.028
SUMNAS† [17]	22.6	0.505 ± 0.039	0.145 ± 0.061
Few-Shot-25 [51]	18.6	0.696	-
GM†-8 [18]	18.0	0.656 ± 0.011	0.153 ± 0.006
CLOSE [52]	2.5	0.643 ± 0.050	0.031 ± 0.021
PA&DA	1.8	$\mathbf{0.713 \pm 0.002}$	$\mathbf{0.301 \pm 0.018}$



- PA&DA only consumes 1.8 GPU hours and reaches the highest KT and P@Top5%.
- Supernet trained by PA&DA has the lowest GV and the highest KT.

ANS@ICT

3 Experiments

智能计算机研究中心
Research Center for Intelligent Computing Systems

3.2 Search performance in DARTS using CIFAR-10

Method	Test Accuracy		Parameters	Search Cost	Search
	Best(%)	Average(%)	(M)	(GPU Days)	Method
NASNet-A [59]	97.35	-	3.3	1,800	RL
ENAS [34]	97.11	-	4.6	0.5	RL
DARTS [30]	-	97.00 ± 0.14	3.3	0.4	Gradient
GDAS [14]	97.07	-	3.4	**0.3**	Gradient
RandomNAS [28]	-	97.15 ± 0.08	4.3	2.7	Random
DARTS-PT [46]	97.52	97.39 ± 0.08	**3.0**	0.8	Gradient
BaLeNAS [54]	-	97.50 ± 0.07	3.8	0.6	Gradient
AGNAS [42]	97.54	97.47 ± 0.003	3.6	0.4	Gradient
ZARTS [47]	-	97.46 ± 0.07	3.7	1.0	Gradient
GDAS-NSAS [53]	97.27	-	3.5	0.4	Gradient
RandomNAS-NSAS [53]	97.36	-	3.1	0.7	Random
Few-Shot-NAS† [56]	97.42	97.37 ± 0.06	3.8	2.8	Gradient
GM [20]	97.60	97.51 ± 0.08	3.7	2.7	Gradient
CLOSE [57]	-	97.28 ± 0.04	4.1	0.6	Gradient
PA&DA	**97.66**	**97.52** ± **0.07**	3.9	0.4	Random

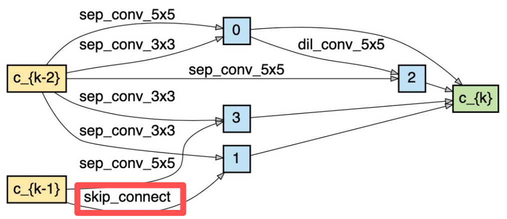
Table 2. Comparison with other state-of-the-art methods on the CIFAR-10 dataset using DARTS search space. We report the best and average test accuracy of repeated experiments. †: reported by GM [20].

- PA&DA only consumes 0.4 GPU days and achieves the best performance.

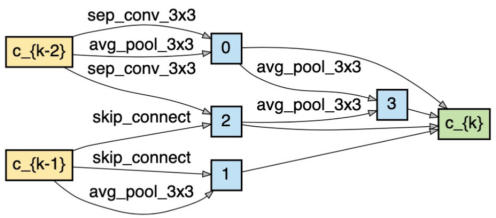
ANS@ICT

3 Experiments

3.2 Searched architectures in DARTS using CIFAR-10

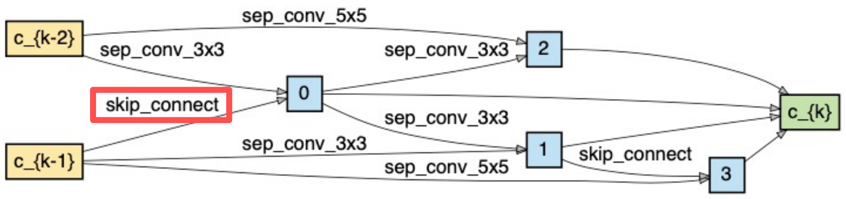


(a) Normal Cell

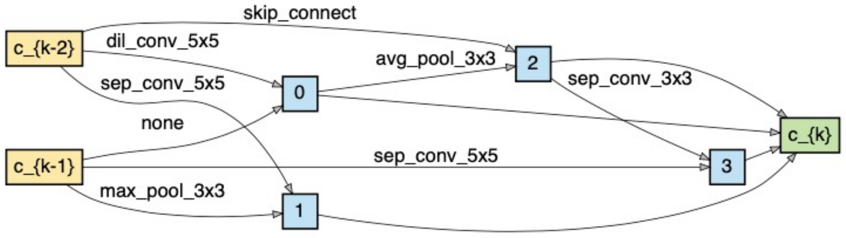


(b) Reduction Cell

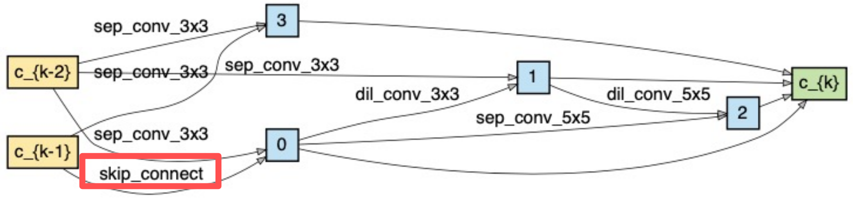
- As pointed out in Cell-based-NAS-Analysis [ICLR'22], such a ResNet-style residual link is helpful for achieving the SOTA performance.



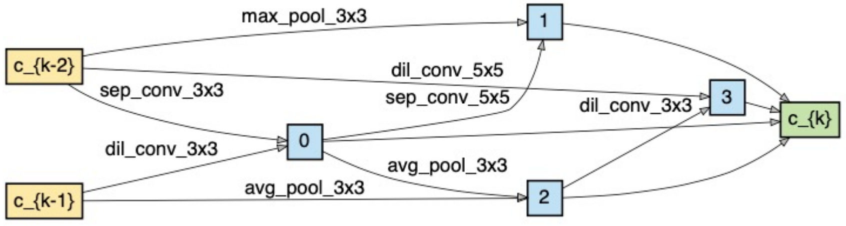
(a) PA&DA-1 (normal cell)



(b) PA&DA-1 (reduction cell)



(c) PA&DA-2 (normal cell)



(d) PA&DA-2 (reduction cell)

3.3 Search performance in ProxylessNAS on ImageNet

Method	Params. (M)	FLOPs (M)	Top-1 (%)	Top-5 (%)
AmoebaNet-A [35]	5.1	555	74.5	92.0
MnasNet-A1 [43]	**3.9**	**312**	75.2	92.5
PNAS [29]	5.1	588	74.2	91.9
TNASP-C [32]	5.3	497	75.8	92.7
DA-NAS [12]	-	389	74.6	-
SPOS [18]	5.4	472	74.8	-
FBNet-C [48]	5.5	375	74.9	-
ProxylessNAS [4]	7.1	465	75.1	92.3
FairNAS-A [9]	4.6	388	75.3	-
MAGIC-AT [50]	6.0	598	76.8	93.3
Few-Shot NAS [56]	4.9	521	75.9	-
GM [20]	4.9	530	76.6	93.0
PA&DA	5.3	399	**77.3**	**93.5**

Table 3. Comparison with other state-of-the-art methods on the ImageNet dataset using the ProxylessNAS search space.

- PA&DA obtains the SOTA performance while using similar FLOPs.

4 Conclusion & Future Work

智能计算机研究中心
Research Center for Intelligent Computing Systems

- **Conclusion**
  - In this paper, we observe that large gradient variance during supernet training harms the ranking consistency.
  - Then we derive the relationship between the gradient variance and the sampling distributions.
  - Finally, we reduce the gradient variance for the supernet training by jointly optimizing the path and data sampling distributions to improve the supernet ranking consistency.

- **Future Work**
  - Explore more effective metrics for data importance.
  - Concentrate more on sub-models located in the Pareto-front, rather than exhaustively evaluate all sub-models.

ANS@ICT

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

智能计算机研究中心
Research Center for Intelligent Computing Systems

Thank You !