Speedy Performance Estimation for Neural Architecture Search

- 作者：Binxin Ru，Clare Lyle， Lisa Schut et al.
- 机构：OATML，University of Oxford， Imperial College London
- 会议：NIPS2021
- 地址：https://arxiv.org/pdf/2006.04492.pdf
- 代码：https://github.com/rubinxin/TSE

论文主要内容

摘要

提出training-speed的指标来估计模型泛化能力，并且容易集成到其他NAS方法中提升速度、质量。

贡献

1. 基于已有的training-speed与泛化能力关系的理论研究来提出TSE
2. 能够结合query-based、 weight-sharing NAS来显著提升NAS速度和质量

研究内容

Motivation

- training speed 和泛化能力之间的关系已有大量工作在研究，但忽略了training trajectory
- 《ABayesianperspectiveon training speed and model selection》中证明
  - 在linear model、无穷宽的deep model执行贝叶斯更新的setting下，marginal likelihood（理论上合理的model selection工具）可以用training speed（sum of negative log predictive likelihoods）来bound
  - 《Pac-bayesian theory meets bayesian inference》最大边缘似然等价于最小化模型泛化误差的PAC-Bayesian bound

By substituting Equation 5 into Equation 2, we obtain a PAC-Bayesian bound, which depends on a sum of negative log likelihoods, described in Section 2:

$$\mathbb{E}_{\theta\sim\rho}\mathbb{E}_{x,y\sim\mathcal{D}}[\ell(f_\theta(x),y)] \leq a + c\left[1 - e^a(Z_{\mathbf{X},\mathbf{y}}\delta)^{\frac{1}{n}}\right]$$
$$\approx a + c\left[1 - e^a\left(e^{-\sum_{k=1}^{n}\ell(f_\theta(x_k),y_k)}\delta\right)^{\frac{1}{n}}\right]. \tag{6}$$

方法

Training Speed Estimation

model若在较少的训练step中快速取得较低的loss，则会比训练慢的model获得更小的loss曲线下面积：能够在training中早、晚期都能区分model表现。

$$\mathrm{TSE} = \sum_{t=1}^{T}\left[\frac{1}{B}\sum_{i=1}^{B}\ell\left(f_{\theta_{t,i}}\left(\mathbf{X}_i\right),\mathbf{y}_i\right)\right]$$

最近有工作表明，最早期的epoch非常不稳定、对网络最后收敛没有信息。因此，将网络训练晚期的epoch赋予更高的权重。

$$\mathrm{TSE-E} = \sum_{t=T-E+1}^{T}\left[\frac{1}{B}\sum_{i=1}^{B}\ell\left(f_{\theta_{t,i}}\left(\mathbf{X}_i\right),\mathbf{y}_i\right)\right]$$
$$\text{TSE-EMA} = \sum_{t=1}^{T}\gamma^{T-t}\left[\frac{1}{B}\sum_{i=1}^{B}\ell\left(f_{\theta_{t,i}}\left(\mathbf{X}_i\right),\mathbf{y}_i\right)\right]$$

- note：使用training loss而不是validation loss(**SoVL**)
- 否则估计的不再是training speed

实验

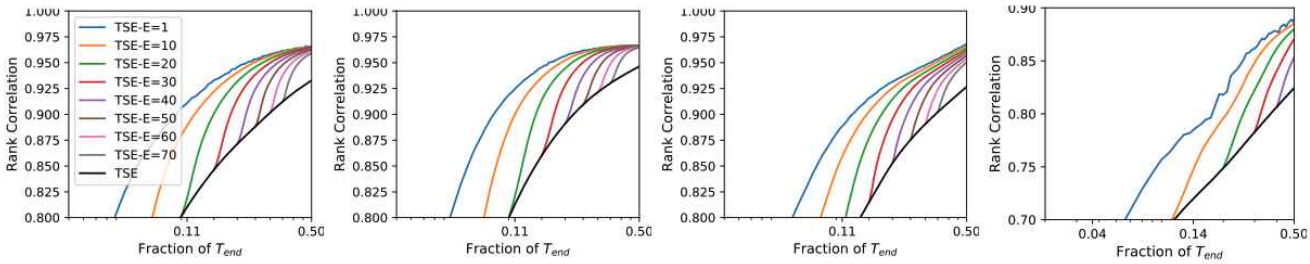
Setting

Table 1: NAS search spaces used. The true test accuracy of architectures from each search space is obtained after training with SGD on the corresponding image datasets for T_{end} epochs. N_{total} denotes the total possible architectures exist in the search space and $N_{samples}$ denotes the number of architectures we sample/generate for our experiments.

Search space	T_{end}	$N_{samples}$	N_{total}	Image datasets
NASBench-201 (NB201) [11]	200	6466	15625	CIFAR10, CIFAR100, ImageNet-16-120
DARTS [28, 42]	100	5000	$\mathcal{O}(2^{42})$	CIFAR10
ResNet/ResNeXt [37]	100	50000	$\mathcal{O}(2^{26})$	CIFAR10
RandWiredNN (RWNN) [46, 40]	250	69×8	$\mathcal{O}(2^{378})$	Flower102

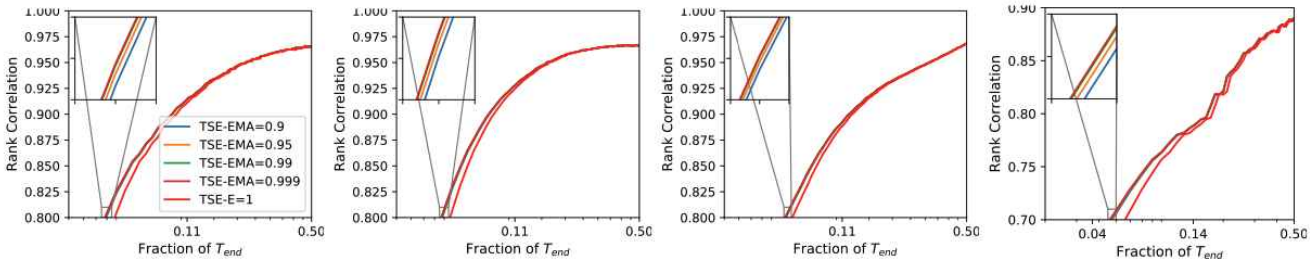
超参

- $E \in [1, 10, 20, \ldots, 70]$ 、 $E \in [0.1, 0.3, 0.5, 0.7]$
  - 结果E=1最好
    - 最靠后的epoch，提供的价值越大
    - 完整一个epoch能够提供利用完整训练集的信息
  - E>=0.3结果和E=1相近
    - 可以使用少量mini-batches来估计TSE（one-shot、gradient-based NAS）
- $\gamma \in [0.9, 0.95, 0.99, 0.999]$ 结果鲁棒



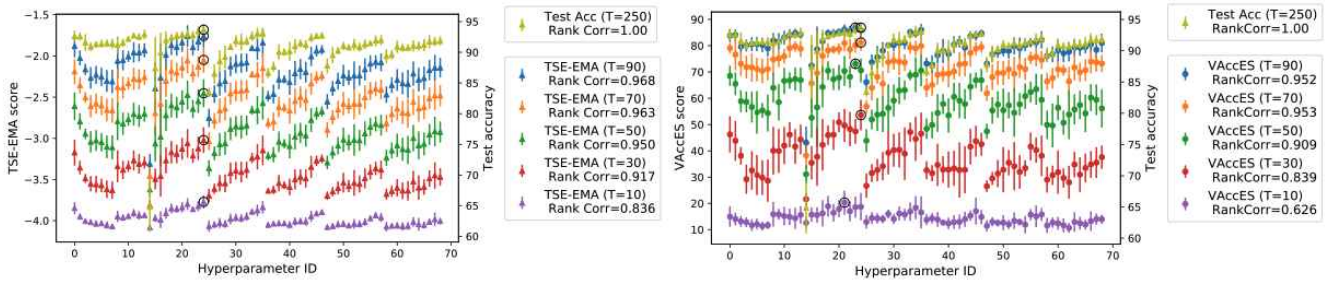
(a) NB201-CIFAR10 (b) NB201-CIFAR100 (c) NB201-IMAGENET (d) DARTS

Figure 10: Rank correlation performance of TSE-E computed over E most recent epochs. Different E values are investigated for architectures in NASBench-201 (NB201) on three image datasets and 5000 architectures from NASBench-301 (DARTS) on CIFAR10. In all cases, smaller E consistently achieves better rank correlation performance with $E = 1$ being the best choice.



(a) NB201-CIFAR10 (b) NB201-CIFAR100 (c) NB201-IMAGENET (d) DARTS

Figure 11: Rank correlation performance of TSE-EMA with γ on architectures in NASBench-201 (NB201) on three image datasets and 5000 architectures from NASBench-301 (DARTS) on CIFAR10. In all cases, TSE-EMA is very robust to different γ values; the difference among TSE-EMA with different γ is indistinguishable compared to that with TSE-E.



(a) Training speed estimator: TSE-EMA (b) Early-stopped validation accuracy: VAccES

Figure 12: Model selection among 69 random graph generator hyperparamters on RandWiredNN dataset using our TSE-EMA (a) and VAccES (b). We use each hyperparameter value to generate 8 architectures and evaluate their true test accuracies after complete training. The mean and standard error of the test performance across 8 architectures for each hyperparameter value are presented as Test Acc (yellow) and treated as ground truth (Right y-axis). We then compute our TSE-EMA estimator for all the architectures by training them for $T < T_{end} = 250$ epochs. The mean and standard error of TSE-EMA scores for $T = 10, \ldots, 90$ are presented in different colours (Left y-axis of (a)). The rank correlation between the mean Test Acc and that of TSE-EMA for various T is shown in the corresponding legends in (a). The same experiment is conducted by using early-stopped validation accuracy (VAccES) for performance estimation (b). With only 10 epochs of training, our TSE-EMA estimator can already capture the trend of the true test performance of different hyperparameters relatively well (Rank correlation$= 0.851$) and can successfully identify 24-th hyperparamter setting as the optimal choice. The prediction of best hyperparameter by VAccES is less consistent and the rank correlation scores of VAccES at all epochs are lower than those of TSE-EMA.

选 $\gamma = 0.999, E = 1$ 作为所有实验默认设置

Results

Baseline对比

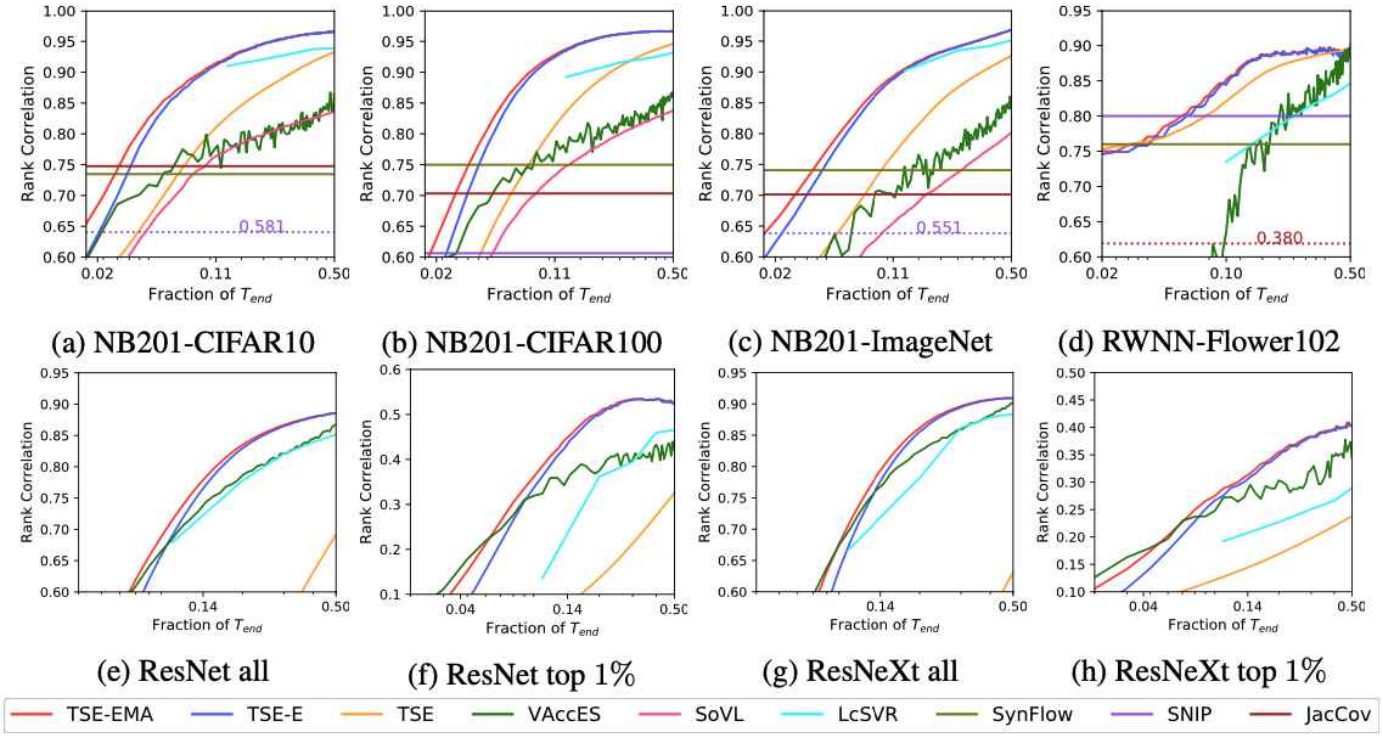


Figure 1: Rank correlation performance of various baselines for architectures from a variety of search spaces: (a) to (c) NB201 architectures on three image datasets, (d) RWNNs on Flowers102 and (e) to (h) ResNet and ResNeXt architectures on CIFAR10. In all cases, our TSE-EMA and TSE-E achieve superior rank correlation with the true test performance in much fewer epochs than other baselines. In (f) and (g), we evaluate estimators on the top 1% of the ResNet/ResNeXt architectures and show that our TSE-EMA and TSE-E can remain competitive on ranking among top architectures, which are particularly desirable for NAS. In (a) and (c), we mark SNIP in a violet dotted line labelled with its rank correlation value as it falls out of the plotted range.

DARTS空间不同training-setting

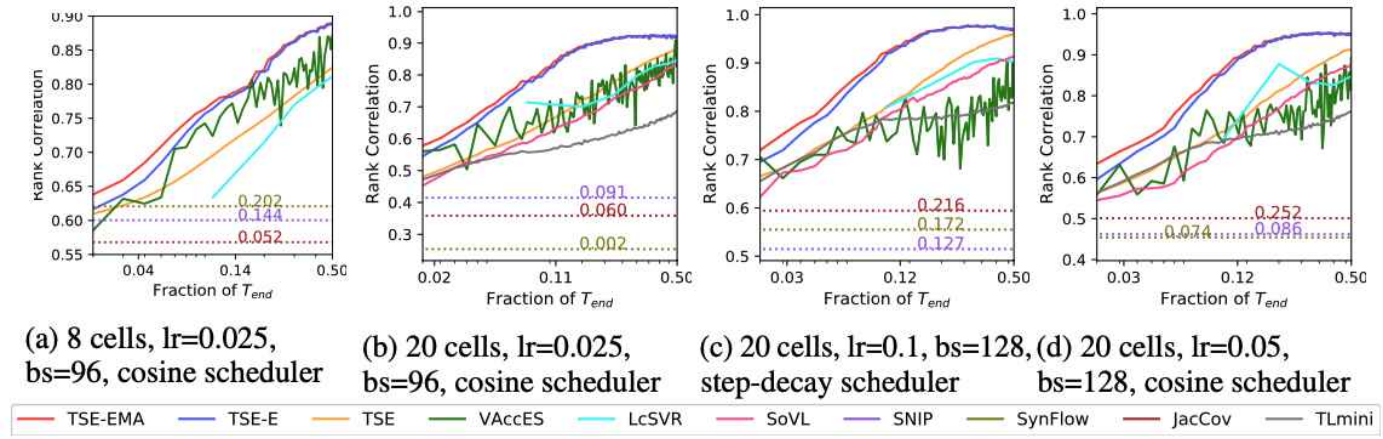
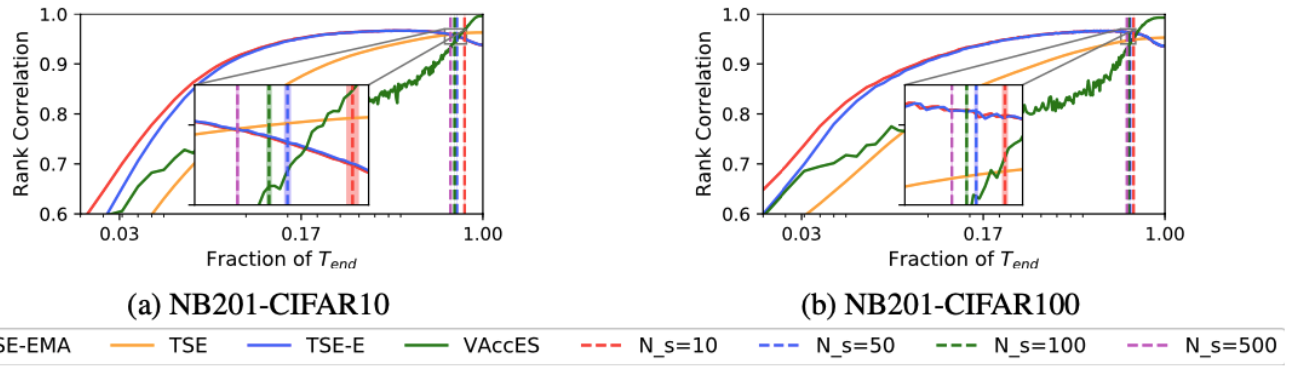


Figure 2: Rank correlation performance of various baselines for 5000 small 8-cell architectures (a) and 150 large 20-cell architectures (b) to (d) from DARTS search space on CIFAR10. We use NAS-Bench-301 dataset(NAS301) for computing (a) and for large architectures, we test three training hyperparameter set-ups with different initial learning rates, learning rate schedulers and batch sizes as denoted in the subcaptions. On all four settings, our TSE-E again consistently achieves superior rank correlation in fewer epochs than other baselines. Note all three zero-cost estimators perform poorly (below the plotted range) on DARTS search space across all settings. We denote them in dotted lines with their rank correlation value labelled.

- TLmini表现比TSE差：对training loss求和才是对training speed的好的估计，而非简单使用一个mini batch loss
- zero-cost在darts空间表现下降很多（甚至比TLmini在T=1的时候还差）
- zero-cost没有考虑不同training-setting带来的影响，所以在不同设置下差异很大

不同traning budget的影响



(a) NB201-CIFAR10　　(b) NB201-CIFAR100

Figure 3: Rank correlation performance up to $T = T_{end}$. If the users want to apply our estimators for large training budget, they can estimate the effective range of our estimators based on the minimum epoch T_o when overfitting happens among the N_s observed architectures. They can then stop our estimators early at $0.9T_o$(marked by vertical lines) or switch back to validation accuracy beyond that.

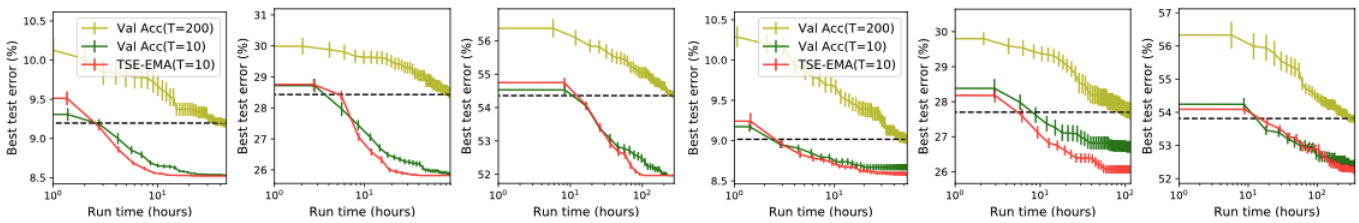
- 在高budget下可以使用在0.9 T_o 处的TSE 或者 换用validation的结果（但需要确定overfit的位置）

Algorithm 1 Find Effective Training Budget for TSE Estimators

Input: A subset of N_s fully trained architectures whose training losses are $\{\{\ell_{i,t}\}_{t=1}^{T_{end}}\}_{i=1}^{N_s}$, Overfitting criterion $is\_overfit()$
Output: The effective training budget $T_{effective}$ to use TSE-EMA or TSE-E
$\mathcal{S}_{T_o} = \emptyset$
for $i = 1, \ldots, N_s$ **do**
  for $t = 1, \ldots, T_{end}$ **do**
    if $is\_overfit(\ell_{i,t}) ==$ True **then**
      $T_{i,o} = T$
      break
    else
      $T_{i,o} = T_{end}$
    end if
  end for
  $\mathcal{S}_{T_o} = \mathcal{S}_{T_o} \cup T_{i,o}$
end for
$T_o = \min \mathcal{S}_{T_o}$
$T_{effective} = 0.9T_o$

加速Query-based NAS

- RE、BO、RS
- nasbench201



(a) Regularised Evolution (RE) (b) Bayesian Optimisation (BO)

Figure 4: NAS performance of RE and BO in combined with final validation accuracy Val Acc (T=200), early-stopping validation accuracy Val Acc (T=10) and our estimator TSE-EMA(T=10) on NB201. For each subplot, we experiment on the three image datasets: on CIFAR10 (left), CIFAR100 (middle) and ImageNet (right). TSE-EMA leads to the fastest convergence to the top performing architectures in all cases. The black dashed line is to facilitate the comparison of runtime taken to reach a certain test error among different variants.

提升one-shot NAS

- 原来每个子网在validation上测acc
- 现在每个子网额外训练100个minibatch算TSE
- DARTS空间、CIFAR-10数据一个架构平均计算val acc需要6.6s。100个TSE minibatch需要7.5s
- NB201空间、CIFAR-10数据一个架构平均计算val acc需要6.4s。100个TSE minibatch需要4.4s

Table 2: Results of performance estimators in one-shot NAS setting over 3 supernetwork training initialisations. For each supernetwork, we randomly sample 500 random subnetworks for DARTS and 200 for NB201, and compute their TSE, Val Acc after inheriting the supernetwork weights and training for B additional mini-batches. Rank correlation measures the estimators' correlation with the rankings of the true test accuracies of subnetworks when *trained from scratch independently*, and we compute the average test accuracy of the top 10 architectures identified by different estimators from all the randomly sampled subnetworks.

B	Estimator	Rank Correlation				Average Accuracy of Top 10 Architectures			
		NB201-CIFAR10			DARTS	NB201-CIFAR10			DARTS
		RandNAS	FairNAS	MultiPaths	RandNAS	RandNAS	FairNAS	MultiPaths	RandNAS
100	TSE	**0.70 (0.02)**	**0.84 (0.01)**	**0.83 (0.01)**	**0.30(0.04)**	**92.67 (0.12**	**92.7 (0.1)**	**92.63 (0.12)**	**93.64(0.04)**
	Val Acc	0.44 (0.15)	0.56 (0.17)	0.67 (0.05)	0.11(0.04)	91.47 (0.31)	91.73 (0.21)	91.77 (0.78)	93.20(0.04)
200	TSE	**0.70 (0.03)**	**0.850 (0.01)**	**0.83 (0.01)**	**0.32(0.04)**	**92.70 (0.00)**	**92.77 (0.06)**	**92.73 (0.06)**	**93.55(0.04)**
	Val Acc	0.41 (0.10)	0.56 (0.17)	0.53 (0.11)	0.09(0.02)	91.53 (0.55)	92.40 (0.10)	92.23 (0.23)	93.34(0.02)
300	TSE	**0.71 (0.03)**	**0.851 (0.00)**	**0.82 (0.01)**	**0.34(0.04)**	**92.70 (0.00)**	**92.77 (0.06)**	**92.70 (0.00)**	**93.65(0.04)**
	Val Acc	0.44 (0.04)	0.62 (0.08)	0.59 (0.71)	0.06(0.02)	91.20 (0.35)	92.10 (0.50)	91.43 (0.72)	93.31(0.02)

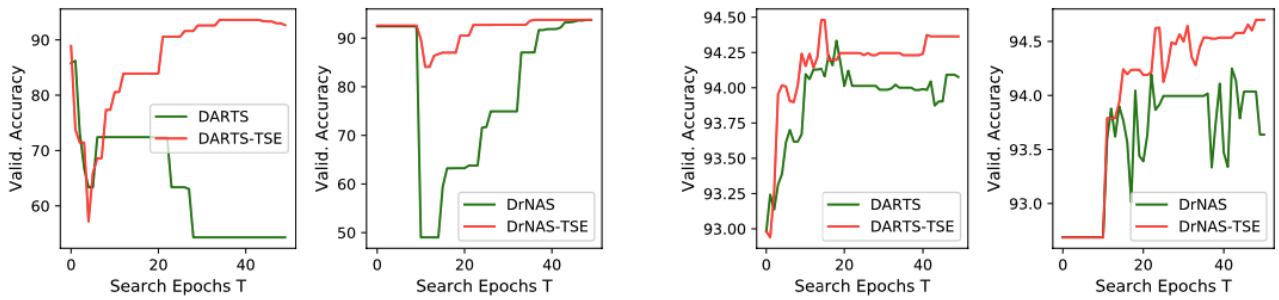
Table 4: Results of performance estimators in one-shot NAS setting over 3 supernetwork training initialisations. For each supernetwork, we randomly sample 500 random subnetworks for DARTS and 200 for NB201, and compute their TSE, Val Acc, SoVL, Tlmini after inheriting the supernetwork weights and training for B additional minibatches. Rank correlation measures the estimators' correlation with the rankings of the true test accuracies of subnetworks when *trained from scratch independently*, and we compute the average test accuracy of the top 10 architectures identified by different estimators from all the randomly sampled subnetworks.

B	Estimator	Rank Correlation				Average Accuracy of Top 10 Architectures			
		NB201-CIFAR10			DARTS	NB201-CIFAR10			DARTS
		RandNAS	FairNAS	MultiPaths	RandNAS	RandNAS	FairNAS	MultiPaths	RandNAS
100	TSE	**0.70 (0.02)**	**0.84 (0.01)**	**0.83 (0.01)**	**0.30(0.04)**	**92.67 (0.12**	**92.70 (0.10)**	92.63 (0.12)	**93.64(0.04)**
	Val Acc	0.44 (0.15)	0.56 (0.17)	0.67 (0.05)	0.11(0.04)	91.47 (0.31)	91.73 (0.21)	91.77 (0.78)	93.20(0.04)
	SoVL	0.54 (0.13)	**0.84 (0.06)**	**0.83 (0.01)**	0.10(0.04)	92.57 (0.15)	92.67 (0.06)	**92.73 (0.06)**	93.21(0.04)
	Tlmini	0.62 (0.03)	0.72 (0.09)	0.74 (0.02)	0.05(0.03)	91.80 (0.40)	92.33 (0.40)	92.43 (0.06)	93.38(0.03)
200	TSE	**0.70 (0.03)**	**0.850 (0.01)**	**0.83 (0.01)**	**0.32(0.04)**	**92.70 (0.00)**	**92.77 (0.06)**	**92.73 (0.06)**	**93.55(0.04)**
	Val Acc	0.41 (0.10)	0.56 (0.17)	0.53 (0.11)	0.09(0.02)	91.53 (0.55)	92.40 (0.10)	92.23 (0.23)	93.34(0.02)
	SoVL	0.52 (0.17)	0.84 (0.06)	0.80 (0.02)	0.08(0.02)	90.70 (1.35)	92.53 (0.15)	92.50 (0.10)	93.36(0.02)
	Tlmini	0.46 (0.14)	0.72 (0.10)	0.69 (0.06)	0.02(0.01)	92.00 (0.35)	92.53 (0.25)	92.40 (0.27)	93.15(0.01)
300	TSE	**0.71 (0.03)**	**0.85 (0.00)**	**0.82 (0.01)**	**0.34(0.04)**	**92.70 (0.00)**	**92.77 (0.06)**	**92.70 (0.00)**	**93.65(0.04)**
	Val Acc	0.44 (0.04)	0.62 (0.08)	0.59 (0.71)	0.06(0.02)	91.20 (0.35)	92.10 (0.50)	91.43 (0.72)	93.31(0.02)
	SoVL	0.45 (0.21)	0.81 (0.05)	0.81 (0.03)	0.05(0.02)	91.00 (1.60)	92.53 (0.15)	92.53 (0.06)	93.26(0.02)
	Tlmini	0.47 (0.12)	0.74 (0.02)	0.70 (0.04)	0.09(0.01)	91.60 (0.44)	92.43 (0.21)	92.27 (0.12)	92.95(0.01)

- 使用TSE，秩相关系数提高170% - 300%

提升Differentiable NAS

- DARTS、DrNAS



(a) NB201-CIFAR10 (b) DARTS-NB301

Figure 5: Test accuracy of the subnetwork recommended by differentiable NAS methods over the search epochs. Original DARTS, DrNAS (in green) use the gradient of validation loss to update the architecture parameters but their variants (DARTS-TSE and DrNAS-TSE) (in red) uses that of our estimator computed over 100 mini-batches. TSE help mitigate the overfitting of DARTS on NB201.

- TSE减轻DARTS在NB201上的过拟合

Algorithm 2 DARTS

Create a mixed operation $\bar{o}^{i,j}$ parametrised by $\alpha^{i,j}$ for each edge (i,j)
for $t = 1, \ldots, BT$ **do**
    Update architecture parameter α by descending $\nabla_{\alpha} \ell_{val}(w, \alpha)$
    Update weights w by descending $\nabla_{w} \ell_{train}(w, \alpha)$
end for
Derive the final architecture based on the learned α

Algorithm 3 DARTS-TSE

Create a mixed operation $\bar{o}^{i,j}$ parametrised by $\alpha^{i,j}$ for each edge (i,j)
for $t = 1, \ldots, \lfloor BT/K \rfloor$ **do**
    Update architecture parameter α by descending $\nabla_{\alpha} \ell_{TSE}(w, \alpha)$
    $\nabla_{\alpha} \ell_{TSE}(w, \alpha) = 0$
    for $k = 1, \ldots, K$ **do**
        Update weights w by descending $\nabla_{w} \ell_{train}(w, \alpha)$
        $\nabla_{\alpha} \ell_{TSE}(w, \alpha) = \nabla_{\alpha} \ell_{TSE}(w, \alpha) + \nabla_{w} \ell_{train}(w, \alpha)$
    end for
end for
Derive the final architecture based on the learned α

用training-speed来作为关于 α 的梯度