Supplementary Material for Less is More: Proxy Datasets in NAS approaches

Brian Moser[1,2], Federico Raue[1], Jörn Hees[1] and Andreas Dengel[1]
[1]German Research Center for Artificial Intelligence (DFKI), Germany
[2]TU Kaiserslautern, Germany
first.second@dfki.de

Table 1. Hyper-parameters for the different NAS-Approaches. ENAS, DARTS, GDAS are listed for Cell Search. Cell Evaluation is using the same hyper-parameters for all approaches. The hyper-parameter values are default, given by NAS-Bench-201.

Parameter	ENAS	DARTS	GDAS	Eval
Optimizer	SGD	SGD	SGD	SGD
LR	0.05	0.025	0.025	0.1
Momentum	0.9	0.9	0.9	0.9
Nesterov	✓	✓	✓	✓
LR Scheduler	Cos	Cos	Cos	Cos
Min. LR	0.0005	0.001	0.001	0
L2-Reg.	0.00025	0.0005	0.0005	0.0005
Epochs	250	50	250	200
Batch Size	128	64	64	256

1. Hyper-Parameters and Architecture Details

Table 1 lists all hyper-parameters during the Cell Search and the Cell Evaluation. The code of NAS-Bench-201 mostly suggests the hyper-parameter values. The Cell Evaluation uses the same hyper-parameters for all Architectures derived by the NAS approaches. One exception is given for ENAS because it has a Controller trained differently from its Child Models. It uses an Adam (learning rate=0.001, β=[0, 0.999], denominator=0.001, default parameters) optimizer for the Controller during the Cell Search [4]. For Cell Evaluation, the last point is irrelevant since the Controller is only necessary for deriving a cell design.

1.1. Sampling via Transfer Learning

ResNet-18 [2] was used for Transfer Learning. Remember that the goal was not to train or fine-tune the best Transfer Learned performance but to have a loss-value-based metric sampling. Therefore, the following modifications might not be the best ones w.r.t. performance.

Two linear layers replaced the last Fully Connected layer with a mapping of $512 \rightarrow 256$ and $256 \rightarrow 100$. In addition, Dropout of 50% and Batch Normalization is used in between [3, 7]. SGD trains the weights with a learning rate of 0.001, a momentum of 0.9, and a batch size of 128 for 50 epochs. In addition, L2 regularization of 10^{-4} is applied [6]. As for other hyper-parameters, these are default values.

1.2. Sampling via Autoencoder

This work uses a simple four layers deep Encoder and four layers deep Decoder. The Encoder is encoding the input into a 64-dimensional latent representation. The architecture is visualized in Figure 1. Like for Sampling via Transfer Learning, the goal was not to find the best performing reconstructing Autoencoder. Instead, it is implemented without any deeper analysis.

1.3. K-Means Outlier Removal

This work uses the K-Means implementation of the Scikit-learn[1] package for Python [5]. Four cases for the number of clusters, K equal 50, 100, 150, and 200 are investigated. Reassigning of clusters happens max. three hundred times or until convergence. Since K-Means is highly dependent on the initialization, this procedure was repeated 100 times for each K. The best clustering was selected, where the cumulative distance of the samples to its centroid is minimal.

1.4. Macro skeleton

For the experiments, this work uses the macro skeleton of NAS-Bench-201; Figure 2 gives an illustration. The Normal Cell Block consists of N stacked Cells. In the following, all experiments use the default value $N = 5$. Within each cell, the number of vertices is also set to five, higher than the default value of NAS-Bench-201 to give more space for variety. During the Cell Search, each Normal Cell's operation has a kernel size of 16. It is different for Cell Evaluation, where the Normal Cell's operation uses a kernel size of 16, 32, and 64.

[1]https://scikit-learn.org

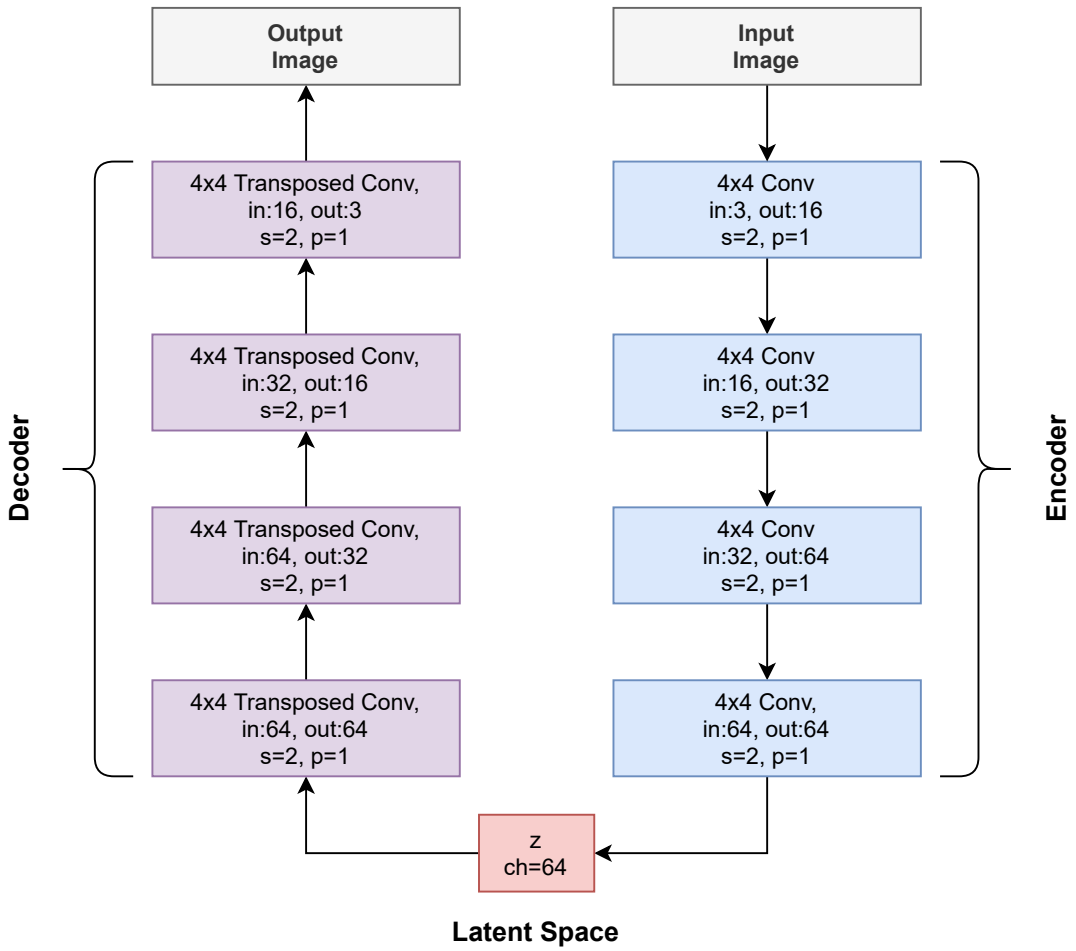

Figure 1. Illustration of the Autoencoder used in this work for CIFAR-100. The input and the output image are RGB images and elements of $\mathbb{R}^{32x32x3}$. The Encoder, as well as the decoder, is a 4-layers deep convolution network. Each layer has a window size of 4. The Encoder uses standard Convolution operations, which reduces the spatial size, whereas the decoder uses Transposed Convolution, which upsamples the spatial size [1]. The latent space representation is a 64-dimensional vector, which is the output of the Encoder and the input for the decoder (denoted as "z" in the illustration). For both components, the layers use a striding of two and a zero-padding of one. Batch Normalization (normalization of the layers' inputs by re-centering and re-scaling) and ReLU (Rectified Linear Unit, a non-linear activation function, $f(x) = max[0, x]$) layers are applied between the convolution layers.

2. Additional Experiments

Like mentioned in the main paper, we also did some additional experiments which did not achieve extraordinary results and are therefore reported here.

2.1. Sampling via Autoencoder

As stated in the main paper, we used the following equation to derive a proxy dataset:

$$\mathcal{D}_r = \underset{\substack{\mathcal{D}' \subset \mathcal{D}, \\ \text{s.t. } |\mathcal{D}'| \simeq r * \mathcal{N}_{\mathcal{D}}}}{\arg\min} \sum_{(\mathcal{X}_i, \cdot) \in \mathcal{D}'} \mathcal{L}\left(D\left(E\left(\mathcal{X}_i\right)\right), \mathcal{X}_i\right), \quad (1)$$

where samples with high loss-values are removed. However, it is possible to do the opposite and derive a proxy dataset, where samples with low loss-values are removed first:

$$\mathcal{D}_r = \underset{\substack{\mathcal{D}' \subset \mathcal{D}, \\ \text{s.t. } |\mathcal{D}'| \simeq r * \mathcal{N}_{\mathcal{D}}}}{\arg\max} \sum_{(\mathcal{X}_i, \cdot) \in \mathcal{D}'} \mathcal{L}\left(D\left(E\left(\mathcal{X}_i\right)\right), \mathcal{X}_i\right), \quad (2)$$

Table 6 shows the results for this second approach.

2.2. Sampling via Transfer Learning

As stated in the main paper, we used the following equation to derive a proxy dataset:

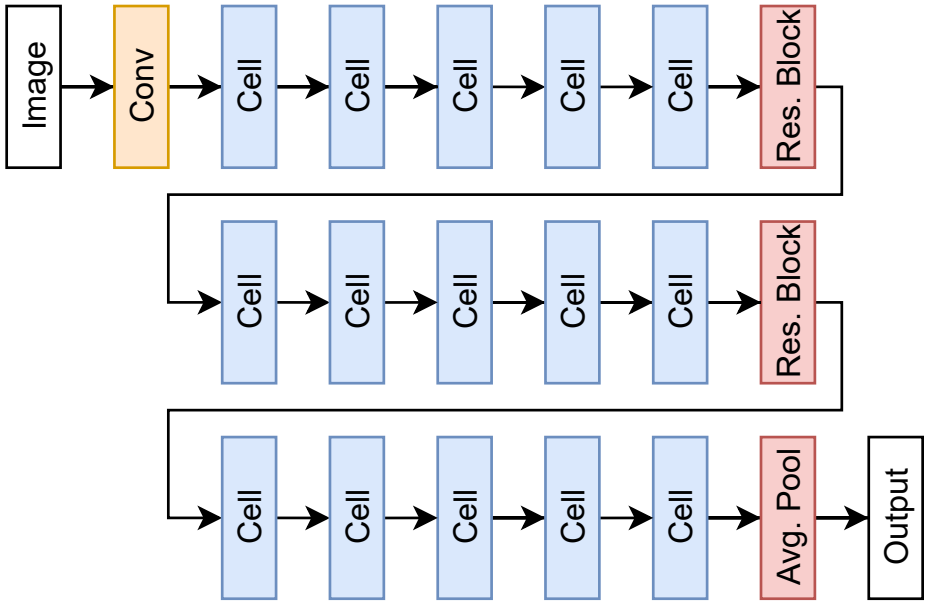

Figure 2. Marco skeleton of NAS-Bench-201 used in this work. It uses $N = 5$ stacked searched cells as Normal Cell Block and Residual Blocks with a stride of two as Reduction Cell Block. The Normal Cells are the product and target of the Search Strategy. Therefore, the Normal Cell changes during the Cell Search, whereas the Normal Cell is fixed during the evaluation.

$$\mathcal{D}_r = \underset{\substack{\mathcal{D}' \subset \mathcal{D}, \\ \text{s.t. } |\mathcal{D}'| \simeq r * \mathcal{N}_{\mathcal{D}}}}{\arg\min} \sum_{(\mathcal{X}_i, \mathcal{Y}_i) \in \mathcal{D}'} \mathcal{L}\left(\varphi\left(\mathcal{X}_i\right), \mathcal{Y}_i\right), \quad (3)$$

where samples with high loss-values are removed. However, it is possible to do the opposite and derive a proxy dataset, where samples with low loss-values are removed first:

$$\mathcal{D}_r = \underset{\substack{\mathcal{D}' \subset \mathcal{D}, \\ \text{s.t. } |\mathcal{D}'| \simeq r * \mathcal{N}_{\mathcal{D}}}}{\arg\max} \sum_{(\mathcal{X}_i, \mathcal{Y}_i) \in \mathcal{D}'} \mathcal{L}\left(\varphi\left(\mathcal{X}_i\right), \mathcal{Y}_i\right), \quad (4)$$

Table 5 shows the results for this second approach.

2.3. K-Means Outlier Removal

Other experiments carried out with different K-Values (50, 150, 200) are shown in Table 2, Table 3, and in Table 4.

References

[1] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016. 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[3] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 1

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 1

[6] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015. 1

[7] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 1

Table 2. Experimental results for K-Means-50.

		Cell Search			Cell Evaluation		
	NAS	Time [s]	Train	Val	CH=16	CH=32	CH=64
Baseline	Darts-V1	66820.5	68.83	61.44	32.60 ± 0.74	40.15 ± 0.37	47.30 ± 0.33
	Darts-V2	194543.4	72.95	61.92	35.70 ± 0.47	45.99 ± 0.28	53.74 ± 0.37
	ENAS	40562.8	12.44	8.92	11.03 ± 0.50	12.51 ± 0.02	13.02 ± 0.02
	GDAS	115855.7	41.82	39.43	65.79 ± 0.33	71.61 ± 0.32	74.31 ± 0.51
$r = 75\%$	Darts-V1	50046.8	65.31	56.16	15.87 ± 0.66	17.71 ± 0.18	18.25 ± 0.06
	Darts-V2	145446.8	66.82	56.50	15.87 ± 0.66	17.71 ± 0.18	18.25 ± 0.06
	ENAS	38980.2	15.04	12.18	11.36 ± 0.38	12.83 ± 0.05	13.07 ± 0.08
	GDAS	87029.2	22.39	32.95	65.85 ± 0.26	71.69 ± 0.46	73.73 ± 0.16
$r = 50\%$	Darts-V1	33618.2	63.80	49.20	49.77 ± 0.33	54.50 ± 0.17	54.91 ± 0.30
	Darts-V2	97712.0	57.56	45.60	28.17 ± 0.29	34.93 ± 0.43	40.52 ± 0.50
	ENAS	35224.4	10.6	9.42	11.62 ± 0.28	12.93 ± 0.03	13.17 ± 0.07
	GDAS	56350.8	28.4	23.00	62.57 ± 0.23	68.15 ± 0.44	72.04 ± 0.43
$r = 25\%$	Darts-V1	16900.9	57.55	37.24	32.93 ± 0.23	41.41 ± 0.53	48.45 ± 0.38
	Darts-V2	48541.7	57.75	36.02	44.99 ± 1.15	51.61 ± 0.80	51.64 ± 0.20
	ENAS	33932.8	7.94	7.34	10.53 ± 0.42	12.21 ± 0.10	12.69 ± 0.08
	GDAS	28608.7	19.77	10.24	59.69 ± 0.52	62.46 ± 0.77	65.86 ± 0.81

Table 3. Experimental results for K-Means-150.

		Cell Search			Cell Evaluation		
	NAS	Time [s]	Train	Val	CH=16	CH=32	CH=64
Baseline	Darts-V1	66820.5	68.83	61.44	32.60 ± 0.74	40.15 ± 0.37	47.30 ± 0.33
	Darts-V2	194543.4	72.95	61.92	35.70 ± 0.47	45.99 ± 0.28	53.74 ± 0.37
	ENAS	40562.8	12.44	8.92	11.03 ± 0.50	12.51 ± 0.02	13.02 ± 0.02
	GDAS	115855.7	41.82	39.43	65.79 ± 0.33	71.61 ± 0.32	74.31 ± 0.51
$r = 75\%$	Darts-V1	50249.9	67.97	57.92	33.39 ± 0.63	42.06 ± 0.23	49.25 ± 0.26
	Darts-V2	146275.5	66.45	56.52	15.87 ± 0.66	17.71 ± 0.18	18.25 ± 0.06
	ENAS	39756.4	11.62	11.58	11.85 ± 0.39	13.18 ± 0.07	13.39 ± 0.03
	GDAS	83889.5	31.73	28.57	66.25 ± 0.35	71.59 ± 0.33	74.23 ± 0.12
$r = 50\%$	Darts-V1	33625.3	61.87	49.12	49.01 ± 0.35	54.52 ± 0.90	54.77 ± 0.64
	Darts-V2	97843.2	63.42	48.32	29.75 ± 0.18	37.40 ± 0.31	42.47 ± 0.12
	ENAS	36354.7	10.80	10.30	10.84 ± 0.54	12.32 ± 0.04	12.89 ± 0.07
	GDAS	56706.0	27.64	20.59	61.69 ± 0.46	68.35 ± 0.14	71.16 ± 0.60
$r = 25\%$	Darts-V1	16885.9	49.36	31.50	32.13 ± 0.56	39.73 ± 0.37	46.38 ± 0.23
	Darts-V2	48906.6	50.65	32.50	42.59 ± 0.50	53.87 ± 0.44	58.94 ± 0.44
	ENAS	33372.3	8.32	7.68	11.43 ± 0.40	12.77 ± 0.06	13.09 ± 0.17
	GDAS	27682.9	21.03	11.42	60.45 ± 0.30	66.09 ± 0.77	69.37 ± 0.28

Table 4. Experimental results for K-Means-200.

		Cell Search			Cell Evaluation		
	NAS	Time [s]	Train	Val	CH=16	CH=32	CH=64
Baseline	Darts-V1	66820.5	68.83	61.44	32.60 ± 0.74	40.15 ± 0.37	47.30 ± 0.33
	Darts-V2	194543.4	72.95	61.92	35.70 ± 0.47	45.99 ± 0.28	53.74 ± 0.37
	ENAS	40562.8	12.44	8.92	11.03 ± 0.50	12.51 ± 0.02	13.02 ± 0.02
	GDAS	115855.7	41.82	39.43	65.79 ± 0.33	71.61 ± 0.32	74.31 ± 0.51
$r = 75\%$	Darts-V1	50046.8	65.31	56.16	15.87 ± 0.66	17.71 ± 0.18	18.25 ± 0.06
	Darts-V2	145446.8	66.82	56.50	15.86 ± 0.66	17.73 ± 0.17	18.27 ± 0.06
	ENAS	38980.2	15.04	12.18	11.87 ± 0.39	13.18 ± 0.07	13.39 ± 0.03
	GDAS	87029.2	22.39	32.95	65.16 ± 0.66	70.67 ± 0.33	73.97 ± 0.37
$r = 50\%$	Darts-V1	33618.2	63.80	49.20	34.94 ± 0.42	43.49 ± 0.15	52.04 ± 0.41
	Darts-V2	97712.0	57.56	45.60	47.17 ± 0.37	57.16 ± 0.51	60.20 ± 0.47
	ENAS	35224.4	10.60	9.42	12.23 ± 0.37	13.31 ± 0.08	13.53 ± 0.03
	GDAS	56350.8	28.40	23.00	65.83 ± 0.26	70.61 ± 0.44	74.02 ± 0.05
$r = 25\%$	Darts-V1	16900.9	57.55	37.24	32.89 ± 0.32	41.58 ± 0.34	48.39 ± 0.23
	Darts-V2	48541.7	57.75	36.02	49.35 ± 0.51	55.57 ± 0.57	55.90 ± 0.27
	ENAS	33932.8	7.94	7.34	11.85 ± 0.39	13.19 ± 0.08	13.40 ± 0.03
	GDAS	28608.7	19.77	10.24	59.89 ± 0.31	65.78 ± 0.15	69.83 ± 0.64

Table 5. Experimental results for Transfer Learning Hard.

		Cell Search			Cell Evaluation		
	NAS	Time [s]	Train	Val	CH=16	CH=32	CH=64
Baseline	Darts-V1	66820.5	68.83	61.44	32.60 ± 0.74	40.15 ± 0.37	47.30 ± 0.33
	Darts-V2	194543.4	72.95	61.92	35.70 ± 0.47	45.99 ± 0.28	53.74 ± 0.37
	ENAS	40562.8	12.44	8.92	11.03 ± 0.50	12.51 ± 0.02	13.02 ± 0.02
	GDAS	115855.7	41.82	39.43	65.79 ± 0.33	71.61 ± 0.32	74.31 ± 0.51
$r = 75\%$	Darts-V1	50127.4	59.68	56.86	33.99 ± 0.32	42.76 ± 0.52	49.95 ± 0.54
	Darts-V2	145719.2	60.73	56.08	62.99 ± 0.32	67.70 ± 0.23	72.73 ± 0.15
	ENAS	40028.8	15.04	14.30	15.87 ± 0.66	17.71 ± 0.18	18.25 ± 0.06
	GDAS	86606.2	22.39	25.12	65.83 ± 0.72	70.66 ± 0.20	73.69 ± 0.50
$r = 50\%$	Darts-V1	33580.9	47.59	44.56	33.36 ± 0.33	43.05 ± 0.60	50.92 ± 0.19
	Darts-V2	95804.4	48.39	45.60	10.57 ± 0.27	12.00 ± 0.03	12.62 ± 0.13
	ENAS	37071.3	10.24	10.08	15.87 ± 0.66	17.71 ± 0.18	18.25 ± 0.06
	GDAS	55989.0	13.02	15.64	58.65 ± 0.21	63.87 ± 1.09	68.34 ± 0.35
$r = 25\%$	Darts-V1	16892.3	30.60	28.26	30.43 ± 0.16	39.07 ± 0.77	46.46 ± 0.31
	Darts-V2	47920.9	33.69	29.30	28.81 ± 0.64	36.45 ± 0.38	43.11 ± 0.47
	ENAS	35037.5	4.48	3.90	10.23 ± 0.25	10.80 ± 0.25	11.69 ± 0.05
	GDAS	27507.8	5.75	6.31	56.51 ± 0.18	60.31 ± 0.45	63.29 ± 0.46

Table 6. Experimental results for AE-Hard.

		Cell Search			Cell Evaluation		
	NAS	Time [s]	Train	Val	CH=16	CH=32	CH=64
Baseline	Darts-V1	66820.5	68.83	61.44	32.60 ± 0.74	40.15 ± 0.37	47.30 ± 0.33
	Darts-V2	194543.4	72.95	61.92	35.70 ± 0.47	45.99 ± 0.28	53.74 ± 0.37
	ENAS	40562.8	12.44	8.92	11.03 ± 0.50	12.51 ± 0.02	13.02 ± 0.02
	GDAS	115855.7	41.82	39.43	65.79 ± 0.33	71.61 ± 0.32	74.31 ± 0.51
$r = 75\%$	Darts-V1	50889.9	64.67	56.98	15.87 ± 0.66	17.71 ± 0.18	18.25 ± 0.06
	Darts-V2	144991.5	64.61	56.68	15.87 ± 0.66	17.71 ± 0.18	18.25 ± 0.06
	ENAS	38535.4	16.74	17.74	15.87 ± 0.66	17.71 ± 0.18	18.25 ± 0.06
	GDAS	85033.6	29.13	27.90	63.73 ± 0.43	69.29 ± 0.12	72.49 ± 0.21
$r = 50\%$	Darts-V1	34021.8	61.59	50.54	34.81 ± 0.23	44.61 ± 0.71	52.57 ± 0.28
	Darts-V2	97419.7	60.10	48.36	27.51 ± 0.54	34.02 ± 0.14	39.67 ± 0.05
	ENAS	35536.5	15.94	13.92	15.87 ± 0.66	17.71 ± 0.18	18.25 ± 0.06
	GDAS	56726.8	27.52	23.90	64.33 ± 0.43	69.91 ± 0.47	73.05 ± 0.42
$r = 25\%$	Darts-V1	17090.7	51.77	36.36	32.91 ± 0.65	42.62 ± 0.10	49.33 ± 0.43
	Darts-V2	48898.3	51.30	35.50	15.87 ± 0.66	17.71 ± 0.18	18.25 ± 0.06
	ENAS	32207.6	7.84	7.36	12.70 ± 0.42	13.46 ± 0.11	13.84 ± 0.11
	GDAS	28389.8	19.03	14.64	58.09 ± 0.61	60.53 ± 0.48	64.36 ± 0.52