Accelerating Neural Architecture Search via Proxy Data

Byunggook Na[1] , **Jisoo Mok**[1] , **Hyeokjun Choe**[1] and **Sungroh Yoon**[1,2*]
[1] Department of Electrical and Computer Engineering, Seoul National University, Seoul, South Korea
[2] AIIS, ASRI, INMC, and Interdisciplinary Program in AI, Seoul National University, Seoul, South Korea
{byunggook.na, jmok908, hyeokjun.choe}@gmail.com, sryoon@snu.ac.kr

Abstract

Despite the increasing interest in neural architecture search (NAS), the significant computational cost of NAS is a hindrance to researchers. Hence, we propose to reduce the cost of NAS using proxy data, *i.e.*, a representative subset of the target data, without sacrificing search performance. Even though data selection has been used across various fields, our evaluation of existing selection methods for NAS algorithms offered by NAS-Bench-1shot1 reveals that they are not always appropriate for NAS and a new selection method is necessary. By analyzing proxy data constructed using various selection methods through data entropy, we propose a novel proxy data selection method tailored for NAS. To empirically demonstrate the effectiveness, we conduct thorough experiments across diverse datasets, search spaces, and NAS algorithms. Consequently, NAS algorithms with the proposed selection discover architectures that are competitive with those obtained using the entire dataset. It significantly reduces the search cost: executing DARTS with the proposed selection requires only 40 minutes on CIFAR-10 and 7.5 hours on ImageNet with a single GPU. Additionally, when the architecture searched on ImageNet using the proposed selection is inversely transferred to CIFAR-10, a state-of-the-art test error of 2.4% is yielded. Our code is available at https://github.com/nabk89/NAS-with-Proxy-data.

1 Introduction

Neural architecture search (NAS), one of the most widely-studied fields in automated machine learning, aims to reduce the human cost of designing and testing hundreds of neural architectures. In early studies pertaining to NAS [Zoph and Le, 2017; Zoph *et al.*, 2018; Real *et al.*, 2019], however, enormous computational overhead occurred, thereby requiring a significant amount of computing resources to execute search algorithms. To reduce the search cost, most recently developed NAS algorithms employ weight-sharing on a super-network and/or differentiable approach to optimize the architecture parameters in the super-network [Xie *et al.*, 2020]. These techniques enable a more approximate yet faster evaluation of candidate neural architecture performance, thereby significantly reducing the cost of NAS.

In this study, we further reduce the search cost of NAS by incorporating proxy data, *i.e.*, a representative subset of the target data. Data selection is widely used across various fields in deep learning, such as active learning [Settles, 2009; Coleman *et al.*, 2020] and curriculum learning [Graves *et al.*, 2017; Chang *et al.*, 2017]. For instance, given a trained model, the core-set selection methods used in active learning aim to select the training data from a large unlabeled dataset to label the selected data with minimum labeling costs, resulting in the effective reduction of the computational cost. However, comprehensive studies regarding the problem of data selection for NAS do not exist. Developing an appropriate selection for NAS is important because NAS algorithms could benefit from the significant reduction in the search cost.

We first evaluate the proxy data constructed using five existing data selection methods on NAS-Bench-1shot1 [Zela *et al.*, 2020b]. While the substantial results provide strong empirical support for our hypothesis, they also reveal the necessity for a new, improved selection method, designed specifically for NAS. Subsequently, we analyze the relationship between the search performances and properties of different selection methods based on the entropy [Shannon, 1948] of examples in the proxy data. Based on our analysis, the characteristics of a selection method that renders proxy data effective in preserving the search performance of NAS algorithms are identified. When the selection method chooses primarily low-entropy examples, a competitive architecture is discovered with the resulting proxy data, even when the size of the proxy data is small. To achieve the search performance obtained using the entire data, it is important to include additional high-entropy examples in the proxy data.

Based on these observations, we propose a new selection method that prefers examples in the tail ends of the data entropy distribution. The selection method can be implemented in a deterministic or probabilistic manner. For the former, we adopt the ratio between low- and high-entropy examples, such that the examples from the opposite ends of the entropy distribution are selected. For the probabilistic manner, we suggest three sampling probabilities that satisfy the identified charac-

*Contact Author

teristics of the proxy data effective for NAS. We demonstrate the superiority of the proposed selection to existing selections using NAS-Bench-1shot1 and show that the search performance is preserved even when using only 1.5K of training examples selected from CIFAR-10.

We further demonstrate that the proposed selection method can be applied universally across various differentiable NAS algorithms and four benchmark datasets for image classification. The NAS algorithms with the proposed selection discover competitive neural architectures in a significantly reduced search time. For example, executing DARTS [Liu *et al.*, 2019] using our selection method requires only **40 GPU minutes** on a single GeForce RTX 2080ti GPU. Owing to the reduction in search cost, searching on ImageNet can be completed in **7.5 GPU hours** on a single Tesla V100 GPU when incorporating the proposed selection into DARTS. The searched architecture achieves the top-1 test error of 24.6%, surpassing the performance of the architecture, which is discovered by DARTS on CIFAR-10 and then transferred to ImageNet. In addition, when this architecture is evaluated on CIFAR-10, it achieves a top-1 test error of **2.4%**, demonstrating state-of-the-art performance on CIFAR-10 among recent NAS algorithms. This indicates that the architectures discovered by NAS algorithms with proxy data selected from a large-scale dataset are highly transferable to smaller datasets.

To summarize, our main contributions are as follows:

- We provide substantial experimental results conducted on NAS-Bench-1shot1 to demonstrate that the existing selection is inappropriate for NAS.
- By identifying the characteristics of effective proxy data selection methods, we propose a novel selection method and two approaches for implementing it.
- We demonstrate the efficacy of the proposed selection for NAS and its general applicability to various NAS algorithms and datasets. We expect the field of NAS to benefit significantly from the reduced search cost afforded using the proposed proxy data selection.

2 Related Work

2.1 Neural Architecture Search

Recently, NAS methods have become diversified significantly, as more complex algorithms have been developed to achieve higher performance or lower search cost [Xie *et al.*, 2020]. Herein, we discuss studies that focus on reducing the search cost. To reduce the search cost, most differentiable and one-shot NAS methods have adopted weight-sharing [Pham *et al.*, 2018] or a continuous architecture with mixing weights [Liu *et al.*, 2019] on a cell-based search space [Zoph *et al.*, 2018]. During the search, a super-network, which stacks multiple cells but is smaller than the target network, is trained.

To further reduce the cost, PC-DARTS [Xu *et al.*, 2020] reduced the number of trainable weight parameters in the cells used during the search by partially bypassing channels in a shortcut. EcoNAS [Zhou *et al.*, 2020] suggested four reduction factors, resulting in a much smaller network than the super-network of conventional cell-based NAS algorithms, and proposed a hierarchical evolutionary algorithm-based search strategy to improve the accuracy of architecture performance estimation using the smaller super-networks. Regarding data selection, EcoNAS briefly reported the search result using a subset randomly sampled from CIFAR-10. In this study, we evaluate various selection methods, including random selection, and provide meaningful insights into the selection method tailored to NAS.

NAS algorithms suffer from the lack of reproducibility, and hence, a fair comparison of NAS algorithms is challenging. Benchmarks for NAS [Ying *et al.*, 2019; Zela *et al.*, 2020b; Dong and Yang, 2020; Dong *et al.*, 2020] aim to alleviate this issue in NAS research. Because these benchmarks provide architecture databases and easy-to-implement NAS algorithm platforms, re-training searched architectures for evaluation can be omitted. Therefore, in this study, we utilize two benchmarks, *i.e.*, NAS-Bench-1shot1 [Zela *et al.*, 2020b] and NATS-Bench [Dong *et al.*, 2020], to investigate selection methods for constructing effective proxy data and evaluate our proxy data selection method.

2.2 Data Selection in Deep Learning

Data selection or subsampling is a well-established methodology in deep learning, and it is used across various fields in deep learning. Active learning [Settles, 2009; Sener and Savarese, 2018; Beluch *et al.*, 2018; Coleman *et al.*, 2020] adopts core-set selections to reduce the labeling cost by selecting the smallest possible number of examples from a large unlabeled dataset. As an application, the approach in core-set selections can be applied to reduce the batch size for training generative adversarial networks [Sinha *et al.*, 2020]. In curriculum learning [Graves *et al.*, 2017; Chang *et al.*, 2017], examples are fed into a neural network efficiently to avoid catastrophic forgetting and accelerate training; hence, curriculum learning ends up utilizing the entire dataset, rather than a subset. However, our results reveal that existing selection methods are not always appropriate for NAS; as such, a new selection method specifically for NAS is necessary.

3 Exploration Study

In this study, we extensively evaluate the search performance of NAS algorithms offered by NAS-Bench-1shot1 using different proxy data constructed using existing selection methods. Based on the results obtained, we identify the characteristics of selection methods that yield particularly effective proxy data for NAS.

3.1 Exploration Setting

NAS-Bench-1shot1 is used as the primary testing platform to observe the effect of the proxy data on the search performance of three NAS algorithms on CIFAR-10: DARTS [Liu *et al.*, 2019], ENAS [Pham *et al.*, 2018], and GDAS [Dong and Yang, 2019]. To construct proxy data of size k, k examples among 50K training examples of CIFAR-10 are selected using selection methods. The selected examples are segregated into two parts: one for updating weight parameters and the other for updating architecture parameters. For

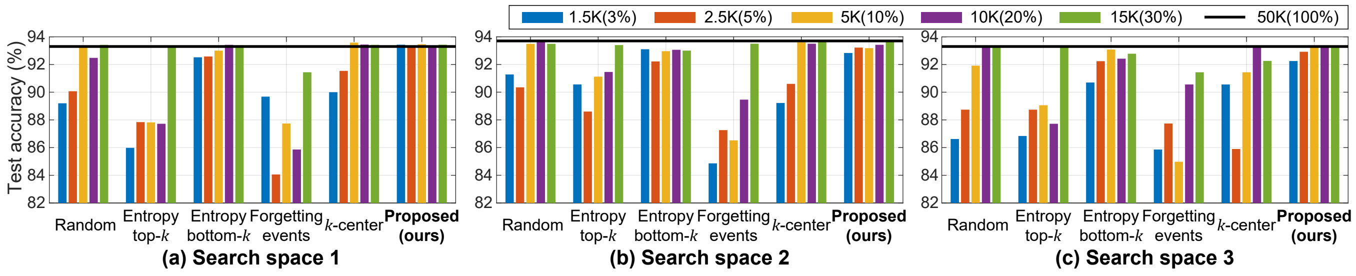

Figure 1: Search performance (CIFAR-10 test accuracy) on NAS-Bench-1shot1 (DARTS) with various proxy data. **Proposed (ours)** indicates the proposed selection method in Section 4, specifically, the probabilistic selection method based on P_1.

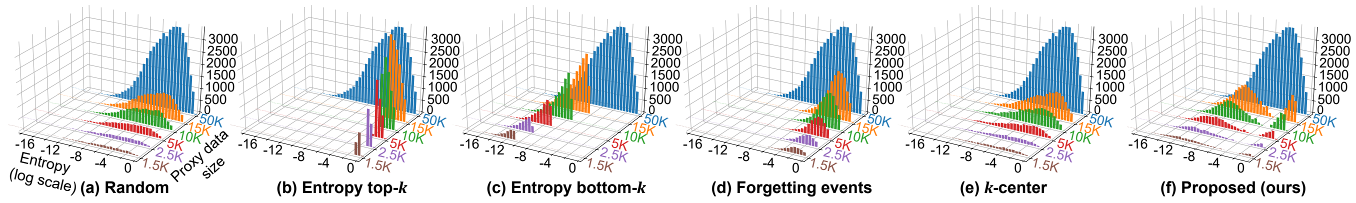

Figure 2: Histograms of data entropy in log scale. Blue histograms indicate the entropy distributions of 50K training examples of CIFAR-10 and the others indicate those of various proxy data constructed using selection methods.

a fair comparison, the same hyperparameter settings as those offered in NAS-Bench-1shot1 are used for all the tested NAS algorithms. To avoid cherry-picking results, we execute the search process five times with different seeds and report the average performance.

The five selection methods utilized in this study are: random, entropy top-k, entropy bottom-k, forgetting events, and k-center. Herein, we provide a brief description of each selection method; more details are included in the Appendix. Random selection, as the name suggests, samples examples uniformly. For entropy-based selection [Settles, 2009], we use the entropy value f_{entropy} of example x calculated as:

$$f_{\text{entropy}}(x; M) = -\sum_{\tilde{\boldsymbol{y}}} P(\tilde{y}|x; M) \log P(\tilde{y}|x; M), \quad (1)$$

where $\tilde{\boldsymbol{y}} = M(x)$ is the predictive distribution of x with respect to the pre-trained baseline model M, *i.e.*, the input of softmax in a classifier. Entropy top-k selection selects examples that have top-k entropy, and entropy bottom-k selection performs the opposite. For the forgetting event selection [Toneva *et al.*, 2019], we train a model from scratch and monitor the number of misclassifications referred to as forgetting events per example. After training the model, examples whose number of forgetting events is in the top-k are selected. In the k-center selection [Sener and Savarese, 2018], given feature vectors extracted from a pre-trained model for all examples, k examples are selected by a greedy k-center algorithm. We set $k \in \{1.5, 2.5, 5, 10, 15\}$K, and pre-train ResNet-20 [He *et al.*, 2016] for the selection methods; the test accuracy of the pre-trained model is 91.7%. Proxy data constructed using the five selections are denoted by D_{random}, D_{top}, D_{bottom}, D_{forget}, and D_{center}, correspondingly.

3.2 Observations and Analysis

The search results of DARTS are shown in Figure 1, and those of ENAS and GDAS are provided in the Appendix. As k changes, two interesting phenomena are observed. First, for $k \leq 2.5$K examples, searching with D_{bottom} consistently yields a search performance closer to that of DARTS with the entire data, namely the original performance. Second, as k increases above 5K, searching with most proxy data results in a rapid increase in the resulting search performance; in the case of D_{bottom}, however, the improvement is less prominent, and the original performance is hardly achieved.

We analyze different proxy data to identify the most significant factor that contributes to the search performance competitive to the original performance using data entropy, f_{entropy}. Data entropy is a typically used metric to quantify the difficulty of an example; furthermore, it is used as the defining property of proxy data in this study. Figure 2 shows the histograms of data entropy of all proxy data in log scale.

As shown in Figure 2(a)-(e), the composition of D_{bottom} differs significantly from those of the other proxy data. When $k \leq 2.5$K, D_{bottom}, which achieves a more competitive search performance than other proxy data, contains a significantly larger number of easy examples. It suggests that to construct proxy data with a number of easy examples is effective for minimizing the size of proxy data and obtaining the original search performance. Meanwhile, D_{random} (or D_{center}) gradually includes easy, middle, and difficult examples, and most additional examples in D_{forget} (or D_{top}) are difficult. It appears that NAS with the proxy data, which appropriately includes middle and difficult examples, can achieve the original search performance when k is sufficiently large; the appropriate value of k differs for each selection. Comprehensively, based on the observed correlations in the search performance and the compositions in the proxy data, we deduce that selection methods for NAS satisfy the following characteristics:

- When a small number of examples are selected, easy examples are more likely to discover a relatively competitive architecture than difficult examples.
- When easy examples are already selected, adding middle and difficult examples enables the original search performance to be achieved.

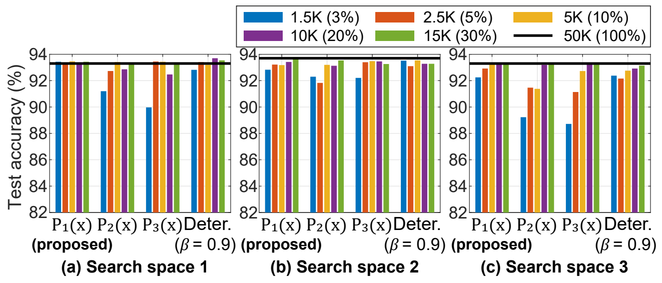

Figure 3: Search performance on NAS-Bench-1shot1 (DARTS) using the proposed methods. $P_{\{1,2,3\}}(x)$ are sampling probabilities used in the proposed probabilistic selection, and Deter. indicates the proposed deterministic selection with $\beta = 0.9$.

4 Proposed Selection Method

Figure 1 shows that the size of the smallest effective proxy data obtained using the existing selection methods is 5K. Although random selection may be considered a strong baseline selection method, its performance deteriorates significantly when $k \leq 2.5$K. In addition, it is noticeable that D_{bottom} with $k \leq 2.5$K achieves the better search performance than the other proxy data. Therefore, to further minimize the size of the proxy data, we propose a new selection method that weighs on examples belonging to both-sided tailed distribution in the data entropy. It is intuitive that the increase in difficult examples provides more information to NAS with easy examples than additional middle examples. As shown in Figure 1, this is supported by results of D_{bottom} with $k = 15$K, which includes a large number of easy examples and a small number of middle examples.

Herein, we suggest two methods for implementing the proposed selection method: deterministic and probabilistic methods. For deterministic selection, we adopt the composition ratio parameter β of low-entropy examples. With respect to data entropy, bottom-βk examples and top-$(1-\beta)k$ examples are selected, where $0 < \beta < 1$. For probabilistic selection, the probability distribution of examples should be designed to satisfy the two aforementioned characteristics. Utilizing histogram information, which can be obtained using a pre-trained baseline model, we design and evaluate three probabilities, denoted by P_1, P_2, and P_3, for probabilistic selection. Let h_x denote a bin where example x belongs in data entropy histogram H, and $|h_x|$ denote the height of h_x, *i.e.*, the number of examples in h_x. The three probabilities are defined as follows:

$$P_{\{1,2,3\}}(x;H) = \text{norm}(W_{\{1,2,3\}}(h_x;H)/|h_x|), \qquad (2)$$

where norm() normalizes the inside term such that $\sum_{x\in D} P_{\{1,2,3\}}(x;H) = 1$ for target data D. In the inside term, selection weights denoted by $W_{\{1,2,3\}}(h_x;H)$ are defined as follows:

$$W_1(h_x;H) = \frac{\max_{h'\in H}|h'| - |h_x| + 1}{\sum_{h''\in H}\max_{h'\in H}|h'| - |h''| + 1}, \qquad (3)$$

$$W_2(h_x;H) = \frac{1}{\text{the number of bins in } H}, \qquad (4)$$

$$W_3(h_x;H) = \frac{1/|h_x|}{\sum_{h''\in H} 1/|h''|}. \qquad (5)$$

In Eq. 2 with $W_2(h_x;H)$, which places equal weights on all bins, examples from tail-ends of H are more likely to be selected. W_1 and W_3 further penalize middle examples by using the difference between height of h_x and the maximum height of the bin near the center in H.

For evaluation on NAS-Bench-1shot1, we execute the proposed selections using 10 different seeds. Among the deterministic selections with $\beta = \{0.9, 0.8, 0.7, 0.6, 0.5\}$, the search performance with $\beta = 0.9$ is the best; the other results are provided in the Appendix. For probabilistic selection, we quantify $|h_x|$ based on a data entropy histogram of CIFAR-10, which is the blue histogram in Figure 2. Figure 2(f) shows that the entropy distribution of examples selected by the proposed selection.

As shown in Figure 3, among the selections, the deterministic selection with $\beta = 0.9$ and the probabilistic selection based on $P_1(x;H)$ achieve the best search performance. In particular, in search space 1, the $P_1(x;H)$-based probabilistic selection achieves the original performance with only $k = 1.5$K examples. Although the deterministic selection with $\beta = 0.9$ achieves a competitive performance as well, finding the optimal β is nontrivial because the optimal β can be dependent on the target data or pre-trained baseline models. By contrast, probabilistic selection does not require additional hyperparameters; as such, an exhaustive hyperparameter search is not necessary for selecting proxy data. Therefore, we set the $P_1(x;H)$-based probabilistic selection as our main method for the remainder of the study.

As shown in Figure 1, the proposed proxy data selection demonstrates better search performance compared with the other existing selections. We include evaluation results on NATS-Bench [Dong *et al.*, 2020] in the Appendix, where the results also show our method is valid on NAS algorithms offered by NATS-Bench. We further demonstrate its effectiveness on an enlarged cell-based search space [Liu *et al.*, 2019] and various NAS algorithms in Section 5.

4.1 Discussion Regarding Efficacy of Proposed Selection

In this section, the factor contributing to the effectiveness of the proposed selection method particularly for NAS is discussed. Many differentible NAS algorithms focus on training a super-network [Xie *et al.*, 2020]. When a super-network is trained to fit only the easier examples, it will naturally converge faster than when it is attempting to fit difficult examples. The side effect of this phenomenon is that the gradient of the loss will become small only after a few epochs of training [Chang *et al.*, 2017] and hence will no longer backpropagate useful signals for the super-network. Therefore, when deriving an architecture from such super-network, it is likely that the resulting architecture will have limited generalization capacity to difficult examples. Using difficult examples allows the super-network to learn more meaningful information, which is difficult to be obtained from easy examples. Using the t-SNE [Maaten and Hinton, 2008] visualization of different proxy data, we can speculate that the missing information from the easy examples is related to the decision boundaries obtained from the pre-trained network and the dataset. As mentioned in Section 4, we use ResNet-20 to

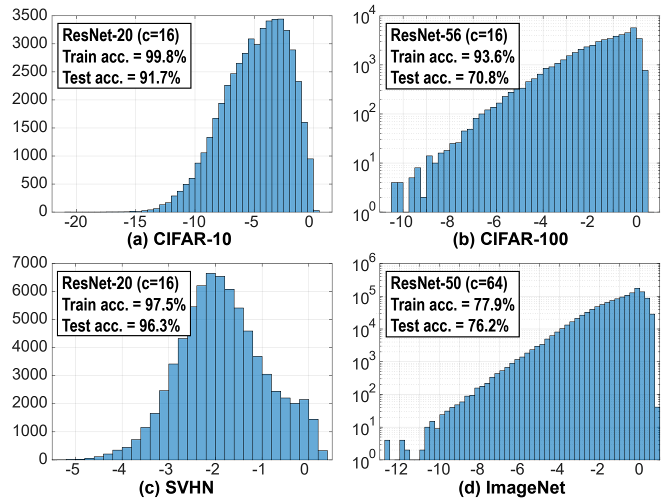

Figure 4: Histograms of data entropy (log scale in x-axis). In each histogram, **c** indicates the number of filters in the first convolutional layer in each ResNet. The histograms from (a) to (d) are obtained by setting a bin width as 0.5, 0.25, 0.2, and 0.25, respectively.

extract features from different proxy data of CIFAR-10; the corresponding t-SNE visualization results are shown in the Appendix. The easy examples tend to be distant from the decision boundaries, unlike the difficult ones. Therefore, for a super-network to learn such decision boundaries, proxy data with difficult examples is required.

Meanwhile, if proxy data is comprised primarily of difficult examples, the stable training of a super-network may be hindered, which is consistent with the results of D_{top} in Figure 1. This issue can be resolved using a sufficient number of easy examples. Consequently, the proposed selection method that satisfies the characteristics identified in Section 3 yields a super-network that is similar to that trained with the entire dataset while the size of the proxy data is minimized.

5 Experiments and Results

We used the pre-trained ResNet-50 in Pytorch model zoo for ImageNet, and trained the three models for CIFAR-10, CIFAR-100, and SVHN; the training took 0.015, 0.033, and 0.022 GPUdays, where the additional cost is negligible. We prepared proxy data using log-scaled data entropy histograms in Figure 4. Although the data entropy distributions of CIFAR-100 and ImageNet show different patterns than those of CIFAR-10 and SVHN, our experimental results consistently indicate that the proposed proxy data selection is valid for CIFAR-100 and ImageNet, as it is for the other two datasets. For the evaluation, we used two types of GPUs: Tesla V100 for searching and training neural architectures on ImageNet, and GeForce RTX 2080ti for the remaining datasets. We execute the search processes using three different seeds and report the averaged values. More details regarding the experimental settings are included in the Appendix.

5.1 Comparison with Random Selection

Based on Section 3, it is apparent that random selection is an effective, reasonable baseline. Hence, we compare the proposed selection with random selection in the cell-based search space using DARTS [Liu *et al.*, 2019] with CIFAR-10; search results with the other selections evaluated in Section 3 are included in the Appendix. We change the size of proxy data from 5K to 25K; the search cost decreases proportionally to the size of proxy data. As shown in Table 1, on CIFAR-10, searching with the proposed selection is usually superior to that using random selection. Furthermore, the search performance with the random selection fluctuates with varying sizes of proxy data. Result of searching with 128K training examples chosen from ImageNet by the proposed selection, is also superior to that of random selection.

Selection	**CIFAR-10**					**IN**
	5K	**10K**	**15K**	**20K**	**25K**	**128K**
Random	3.21	2.95	2.99	2.72	3.22	25.2
Proposed	2.94	2.92	2.88	2.78	2.76	24.6

Table 1: Evaluation (top-1 test error (%)) of DARTS with varying sizes of proxy data of CIFAR-10 and ImageNet denoted by **IN**.

NAS algorithm	**Base**		**Proposed**	
	Err. (%)	**Cost**	**Err. (%)**	**Cost**
CIFAR-10 (Base: 50K, Proposed: 5K)				
DARTS	3.00	0.26	2.94	0.03
PC-DARTS	2.67	0.08	2.91	0.01
EcoDARTS-c4r2	2.80	0.23	2.81	0.02
SDARTS-RS	2.67	0.23	2.83	0.03
SGAS-Cri.1	2.66	0.19	2.72	0.02
ImageNet (Base: 1.28M, Proposed: 128K)				
DARTS	26.7	-	24.6	0.32
PC-DARTS	24.2	3.8[†]	24.3	0.26

Table 2: Evaluation of various NAS methods used on cell-based search space using proposed proxy data selection. Search cost is GPU days and single 2080ti GPU and V100 GPU are used for searching on CIFAR-10 and ImageNet, respectively. [†]Authors of PC-DARTS reported that search process on ImageNet required 11.5 hours with eight V100 GPUs, *i.e.*, 3.8 GPU days.

5.2 Applicability to NAS Algorithms

Recently, various differentiable NAS algorithms based on a cell-based search space have been proposed [Xie *et al.*, 2020]. We apply the proposed proxy data selection to the recently proposed NAS, *i.e.*, DARTS [Liu *et al.*, 2019], PC-DARTS [Xu *et al.*, 2020], SDARTS [Chen and Hsieh, 2020], SGAS [Li *et al.*, 2020], and EcoDARTS that is a DARTS-based variant of EcoNAS [Zhou *et al.*, 2020], respectively. As shown in Table 2, all of the tested NAS algorithms achieve the comparable performance to their respective original search performance. While on CIFAR-10, PC-DARTS with the propose selection experiences a slight decrease in performance, on ImageNet, it succesfully achieves the original search performance with significantly reduced search cost.

None of the existing NAS algorithms searched directly on ImageNet, with the exception of PC-DARTS. To perform the direct search on ImageNet, we incorporate the proposed selection with PC-DARTS and DARTS. DARTS with the pro-

NAS alg.	Data	Search space	Base 100%	Proposed 10%	Proposed 20%
DARTS	CIFAR-10	S1	3.84	3.60	2.96
		S2	4.85	3.54	3.46
		S3	3.34	2.71	2.72
		S4	7.20	6.60	5.82
	CIFAR-100	S1	29.46	26.41	28.79
		S2	26.05	21.65	22.66
		S3	28.90	22.10	23.51
		S4	22.85	98.91	25.74
	SVHN	S1	4.58	3.12	4.11
		S2	3.53	2.81	3.03
		S3	3.41	2.77	3.11
		S4	3.05	3.06	2.42
RobustDARTS (L2)	CIFAR-10	S1	2.78	2.79	2.86
		S2	3.31	3.33	2.98
		S3	2.51	2.74	2.80
		S4	3.56	3.41	3.43
	CIFAR-100	S1	24.25	26.13	23.67
		S2	22.24	22.21	21.39
		S3	23.99	21.71	22.31
		S4	21.94	27.83	21.10
	SVHN	S1	4.79	2.46	2.60
		S2	2.51	2.45	2.49
		S3	2.48	2.53	2.42
		S4	2.50	5.16	2.62

Table 3: Evaluation (top-1 test error (%)) in four restricted cell-based search spaces and three datasets.

posed selection discovers a better architecture than the original DARTS, which transfers the architecture searched on CIFAR-10 to ImageNet. The original PC-DARTS searched on ImageNet with 12.5% of examples randomly sampled from the dataset. It required 3.8 GPU days, *i.e.*, 11.5 hours with eight V100 GPUs, with a batch size of 1024; we speculate that the parallel execution on the eight GPUs resulted in a non-negligible overhead. By contrast, PC-DARTS with the proxy data which consists of 10% of examples constructed using the proposed selection, can discover the competitive architecture using a single V100 GPU with a batch size of 256 in approximately 0.26 GPU days, *i.e.*, 14.6 times less cost than that of the original PC-DARTS.

5.3 Applicability to Datasets

To demonstrate the general applicability of the proposed selection to datasets, we test it on CIFAR-10, CIFAR-100, and SVHN using DARTS and RobustDARTS [Zela *et al.*, 2020a] in four different search spaces. These search spaces were modified from the cell-based search space by reducing the types of candidate operations (S1-S3) and inserting harmful noise operation (S4); the details are provided in the Appendix. Following the experimental protocols in RobustDARTS, the weight decay factors for DARTS and RobustDARTS (L2) during search are set to be 0.0003 and 0.0243, respectively.

As shown in Table 3, most results of the two NAS algorithms using the proposed selection are within a reasonable range of the original search performance. However, when DARTS is executed on S4 with 10% of examples from CIFAR-100, a significant search failure occurs. This failure is caused because noise operations in S4 occupy most of the edges in the cell structure after search. Note that the noise operation is intended for inducing failure in DARTS [Zela *et al.*, 2020a] and is generally not used in practice. Nevertheless, the original search performance on CIFAR-100 can be obtained when using 20% of examples.

5.4 Inverse Transferability

Typically, in most NAS algorithms, the transferability of architectures discovered using CIFAR-10 is demonstrated by their performance on ImageNet. Using DARTS with the proposed selection, the computational cost of searching with ImageNet is reduced by $\frac{1}{10}$, *i.e.*, 0.26 GPU days. The resulting search time on the proxy data of ImageNet is similar to those of other NAS algorithms on the entire CIFAR-10. Therefore, granted the similar amounts of search cost for fair comparison, architectures discovered on ImageNet using the proposed selection can be evaluated on CIFAR-10, which is the inverse way from conventional studies.

Consequently, the architecture searched on ImageNet using the proposed selection yields a top-1 test error of **2.4%** on CIFAR-10, *i.e.*, the best performance among cell-based NAS algorithms; the results with recent NAS algorithms are provided in the Appendix. It is noteworthy that we do not utilize additional techniques introduced in recent studies, and that the architecture above is discovered only by executing DARTS on the proxy data of a large-scale dataset. We speculate that the use of ImageNet provides DARTS with more helpful visual representations than CIFAR-10. We refer to such an approach of transferring an architecture from a large-scale dataset to a smaller dataset as *inverse transfer*. Our study reveals that if the search cost on a large-scale dataset is reasonably low, then the inverse transfer of an architecture can provide new directions for NAS research.

6 Conclusion

For the first time in NAS research, we introduced proxy data for accelerating NAS algorithms without sacrificing search performance. After evaluating existing selection methods on NAS-Bench-1shot1, we obtained the insights and proposed a novel selection method for NAS, which prefers examples in tail-end of entropy distribution of the target data. We thoroughly demonstrated the NAS acceleration and applicability of the proposed probabilistic selection on various datasets and NAS algorithms. Notably, a direct search on ImageNet was completed in 7.5 GPU hours, suggesting that the inverse transfer approach is valid. We expect other studies on NAS to benefit from the significant reduction in the search cost through the use of proxy data.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) [2018R1A2B3001628], Samsung Electronics Co., Ltd, and the Brain Korea 21 Plus Project in 2021.

References

[Beluch *et al.*, 2018] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *Proceedings of CVPR*, pages 9368–9377, 2018.

[Chang *et al.*, 2017] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Proceedings of NeurIPS*, pages 1002–1012, 2017.

[Chen and Hsieh, 2020] Xiangning Chen and Cho-Jui Hsieh. Stabilizing differentiable architecture search via perturbation-based regularization. In *Proceedings of ICML*, 2020.

[Coleman *et al.*, 2020] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *Proceedings of ICLR*, 2020.

[Dong and Yang, 2019] Xuanyi Dong and Yi Yang. Searching for a robust neural architecture in four gpu hours. In *Proceedings of CVPR*, pages 1761–1770, 2019.

[Dong and Yang, 2020] Xuanyi Dong and Yi Yang. Nas-bench-201: Extending the scope of reproducible neural architecture search. In *Proceedings of ICLR*, 2020.

[Dong *et al.*, 2020] Xuanyi Dong, Lu Liu, Katarzyna Musial, and Bogdan Gabrys. NATS-Bench: Benchmarking nas algorithms for architecture topology and size. *arXiv preprint arXiv:2009.00437*, 2020.

[Graves *et al.*, 2017] Alex Graves, Marc Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *Proceedings of ICML*, pages 1311–1320, 2017.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of CVPR*, pages 770–778, 2016.

[Li *et al.*, 2020] Guohao Li, Guocheng Qian, Itzel C Delgadillo, Matthias Muller, Ali Thabet, and Bernard Ghanem. Sgas: Sequential greedy architecture search. In *Proceedings of CVPR*, pages 1620–1630, 2020.

[Liu *et al.*, 2019] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *Proceedings of ICLR*, 2019.

[Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[Pham *et al.*, 2018] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. In *Proceedings of ICML*, pages 4095–4104, 2018.

[Real *et al.*, 2019] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of AAAI*, volume 33, pages 4780–4789, 2019.

[Sener and Savarese, 2018] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *Proceedings of ICLR*, 2018.

[Settles, 2009] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[Shannon, 1948] Claude Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[Sinha *et al.*, 2020] Samarth Sinha, Han Zhang, Anirudh Goyal, Yoshua Bengio, Hugo Larochelle, and Augustus Odena. Small-gan: Speeding up gan training using core-sets. In *Proceedings of ICML*, 2020.

[Toneva *et al.*, 2019] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *Proceedings of ICLR*, 2019.

[Xie *et al.*, 2020] Lingxi Xie, Xin Chen, Kaifeng Bi, Longhui Wei, Yuhui Xu, Zhengsu Chen, Lanfei Wang, An Xiao, Jianlong Chang, Xiaopeng Zhang, and Qi Tian. Weight-sharing neural architecture search: A battle to shrink the optimization gap, 2020.

[Xu *et al.*, 2020] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. Pc-darts: Partial channel connections for memory-efficient architecture search. In *Proceedings of ICLR*, 2020.

[Ying *et al.*, 2019] Chris Ying, Aaron Klein, Eric Christiansen, Esteban Real, Kevin Murphy, and Frank Hutter. Nas-bench-101: Towards reproducible neural architecture search. In *Proceedings of ICML*, pages 7105–7114, 2019.

[Zela *et al.*, 2020a] Arber Zela, Thomas Elsken, Tonmoy Saikia, Yassine Marrakchi, Thomas Brox, and Frank Hutter. Understanding and robustifying differentiable architecture search. In *Proceedings of ICLR*, 2020.

[Zela *et al.*, 2020b] Arber Zela, Julien Siems, and Frank Hutter. Nas-bench-1shot1: Benchmarking and dissecting one-shot neural architecture search. In *Proceedings of ICLR*, 2020.

[Zhou *et al.*, 2020] Dongzhan Zhou, Xinchi Zhou, Wenwei Zhang, Chen Change Loy, Shuai Yi, Xuesen Zhang, and Wanli Ouyang. Econas: Finding proxies for economical neural architecture search. In *Proceedings of CVPR*, pages 11396–11404, 2020.

[Zoph and Le, 2017] Barret Zoph and Quoc Le. Neural architecture search with reinforcement learning. In *Proceedings of ICLR*, 2017.

[Zoph *et al.*, 2018] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of CVPR*, pages 8697–8710, 2018.