Supplementary Materials for FlexiBO: Cost-Aware Multi-Objective Optimization of Deep Neural Networks

Md Shahriar Iqbal
University of South Carolina

Jianhai Su
University of South Carolina

Lars Kotthoff
University of Wyoming

Pooyan Jamshidi
University of South Carolina

Abstract

In this paper, we propose FlexiBO, a flexible Bayesian multi-objective optimization algorithm to address this issue. We formulate a new acquisition metric based on the improvement of Pareto hyper-volume that includes the measurement cost of each objective. Our hyper-volume improvement calculation uses the acquisition metric to select the next sample and objective with maximum information gain per cost for evaluation, rather than all objectives. We apply FlexiBO to optimize 8 state-of-the-art DNN architectures for object detection, natural language processing, and speech recognition. Our results indicate that, for the same evaluation cost, the Pareto-front obtained using FlexiBO has a 46.79% higher contribution to the true Pareto-front and a 33.45% better diversity compared to other state-of-the-art methods.

1 Introduction

Default hyperparameters used for different architectures in FlexiBO experiments are shown in Table 3. Hardware and OS/Kernel level configuration options are chosen using the causal graphical models shown in Figure 1, Figure 2 and Figure 3 for Image recognition, natural language processing (NLP) and speech recognition, respectively. If a configuration option has a direct connection to a performance objective (energy consumption) then the options is said to causally influence performance.

11th ACM/SPEC International Conference on Performance Engineering (ICPE 2020), Edmonton, Canada.

Table 1: Cost after every 20th iteration for optimizing accuracy (bolded value indicates total cost).

Iteration	**Cost**	
	Cost-Aware	Cost-Unaware
1	17.6	17.6
20	35.2	35.2
40	35.2	52.8
60	35.2	70.4
80	52.8	88
100	70.4	105.6
120	88	123.2
140	88	140.8
160	105.6	158.4
180	123.2	176
200	**140.8**	**193.6**

Table 2: Cost after every 20th iteration for optimizing energy consumption. bolded value indicates total cost

Iteration	**Cost**	
	Cost-Aware	Cost-Unaware
1	17.6	17.6
20	17.6	35.2
40	35.2	52.8
60	35.2	70.4
80	35.2	88
100	52.8	105.6
120	70.4	123.2
140	88	140.8
160	88	158.4
180	88	176
200	**88**	**193.6**

Table 3: Network hyperparameters for different architectures of object detection, NLP and speech recognition DNN systems.

Hyperparameter	Object detection					NLP	Speech recognition
	Xception	MobileNet	LeNet	resnet	SqueezeNet	BERT-base	Deepspeech
Num. channels, $\boldsymbol{\zeta}$	3	3	3	3	3	-	-
Num. classes, $\lvert\boldsymbol{C}\rvert$	100	1000	1000	10	10	-	-
Epochs, $\boldsymbol{\epsilon}$	200	100	200	200	200	-	-
Batch size, $\boldsymbol{b}$	20	32		32	32	-	-
Learning rate, $\boldsymbol{\eta}$	0.00001	0.0001	0.0001	0.0001	0.0001	0.00003	0.0001
Decay, $\boldsymbol{\lambda}$	$1e^{-6}$	$1e^{-6}$	$1e^{-6}$	$1e^{-6}$	$1e^{-6}$	$3e^{-6}$	$1e^{-6}$
Dropout, $\boldsymbol{p}$	0.5	0.5	0.5	0.5	0.5	0.248	0.2367
Loss function, $\boldsymbol{L}$	categorical cross-entropy	categorical cross-entropy	categorical cross-entropy	categorical cross-entropy	categorical cross-entropy	categorical cross-entropy	categorical cross-entropy
Width multiplier, $\boldsymbol{\alpha}$	1	1	1	1	1	-	-
Data augmentation	Yes	Yes	Yes	Yes	Yes	-	-
Default standard deviation, σ	-	-	-	-	-	-	0.046875
Doc stride	-	-	-	-	-	128	-

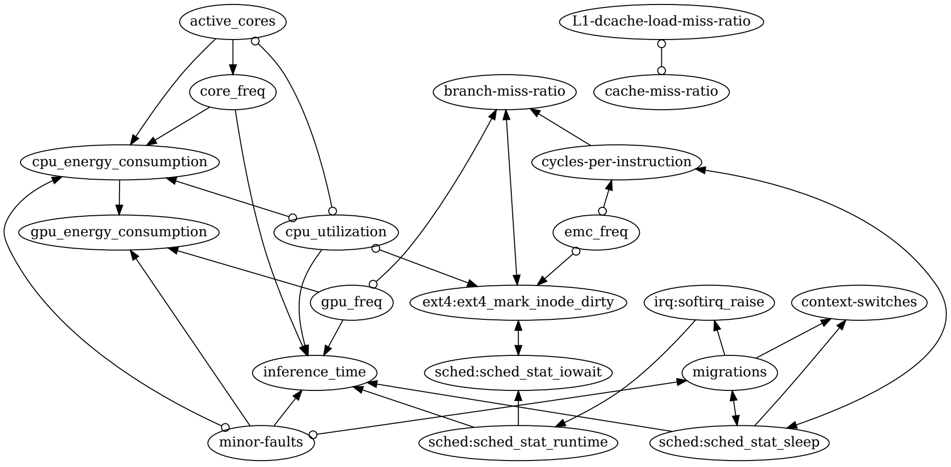

Figure 1: A partial causal model of the NLP DNN system. Performance nodes are inference_time, cpu_energy_consumption and gpu_energy_consumption.

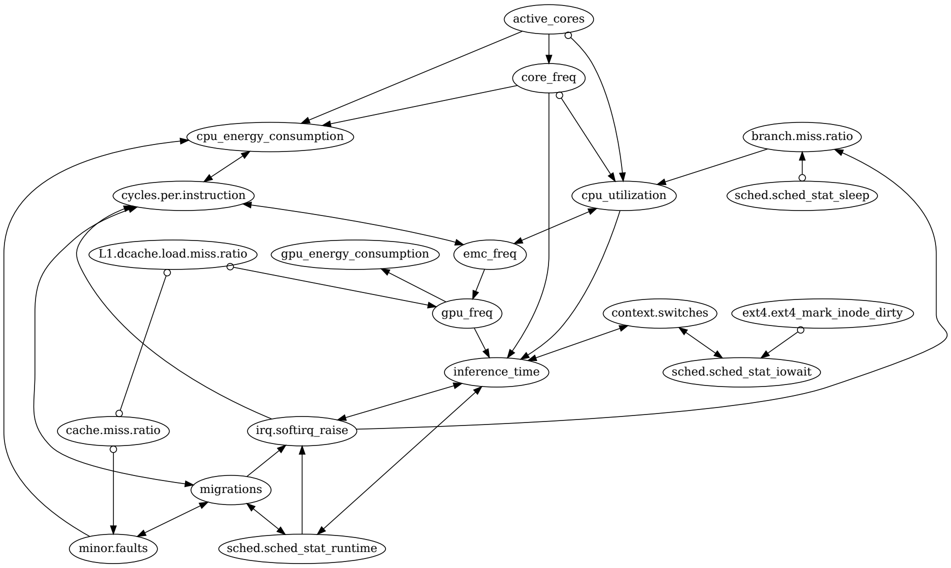

Figure 2: A partial causal model of the NLP DNN system. Performance nodes are inference_time, cpu_energy_consumption and gpu_energy_consumption.

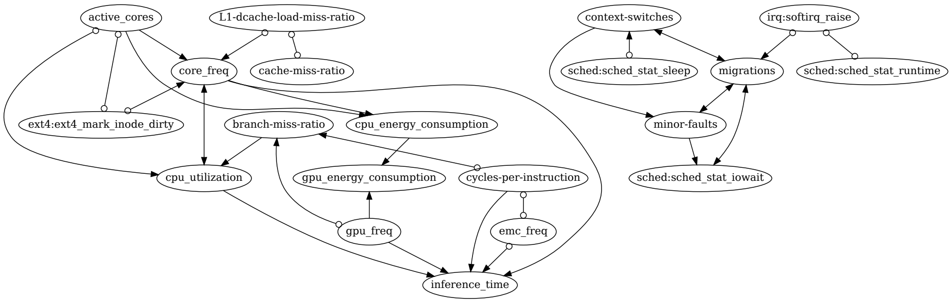

Figure 3: A partial causal model of the NLP DNN system. Performance nodes are inference_time, cpu_energy_consumption and gpu_energy_consumption.

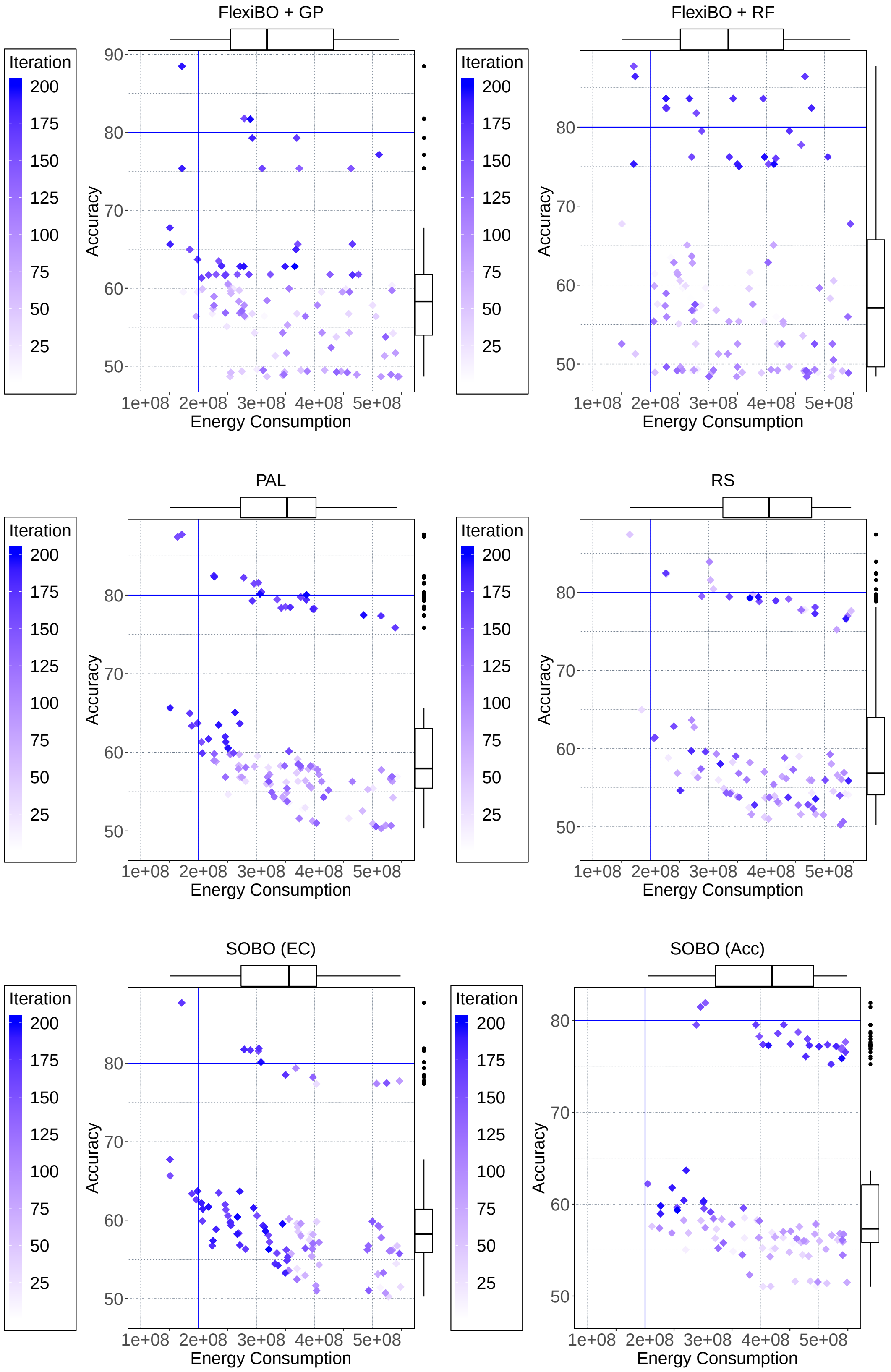

Figure 4: Exploration results in BERT-IMDB. It is desirable to have more evaluations to the top left half of the figure for optimization. FlexiBO outperforms other methods, finding more accurate and more efficient DNN architectures.

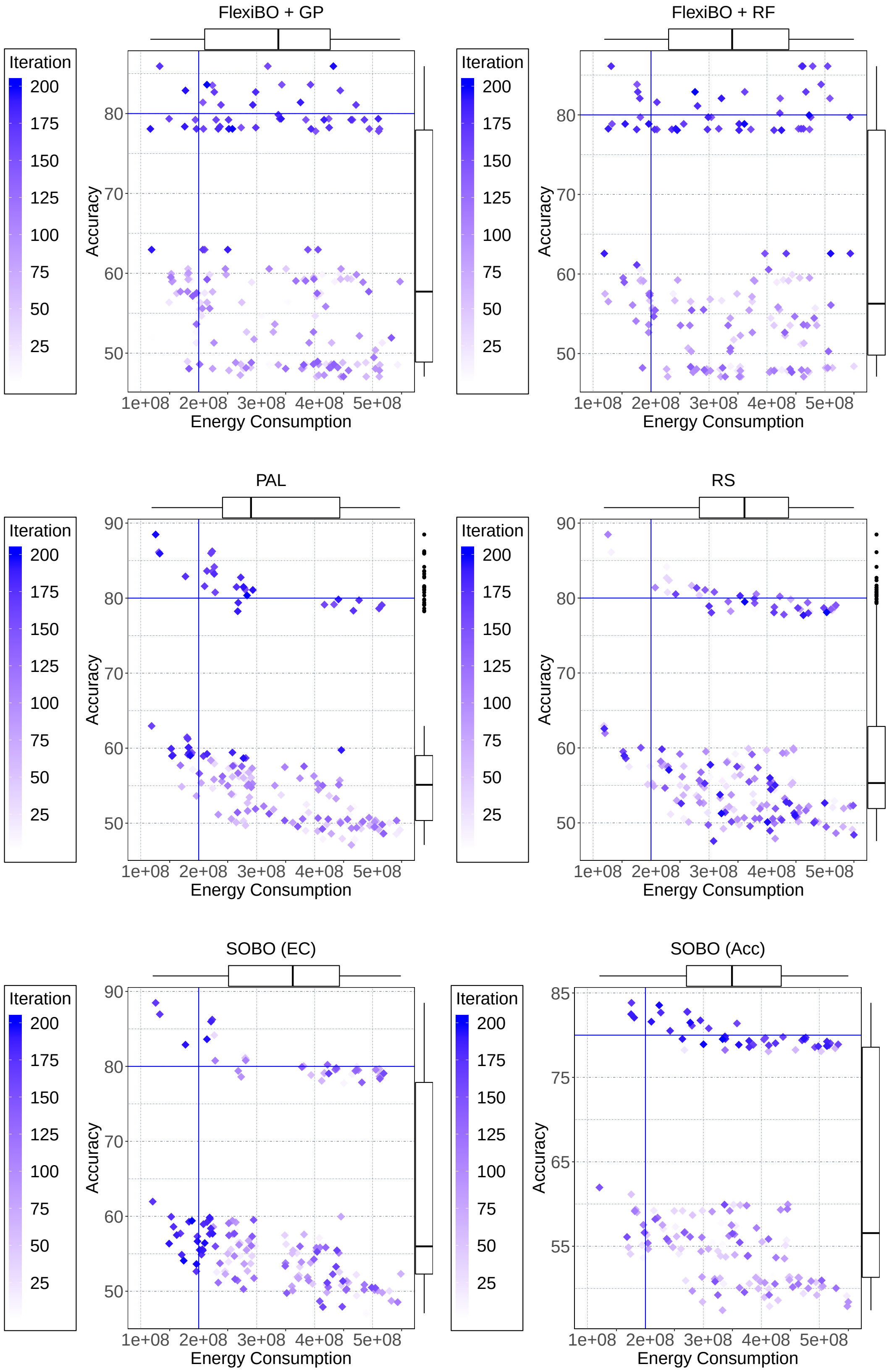

Figure 5: Exploration results in BERT-SQuAD. It is desirable to have more evaluations to the top left half of the figure for optimization. FlexiBO outperforms other methods, finding more accurate and more efficient DNN architectures.

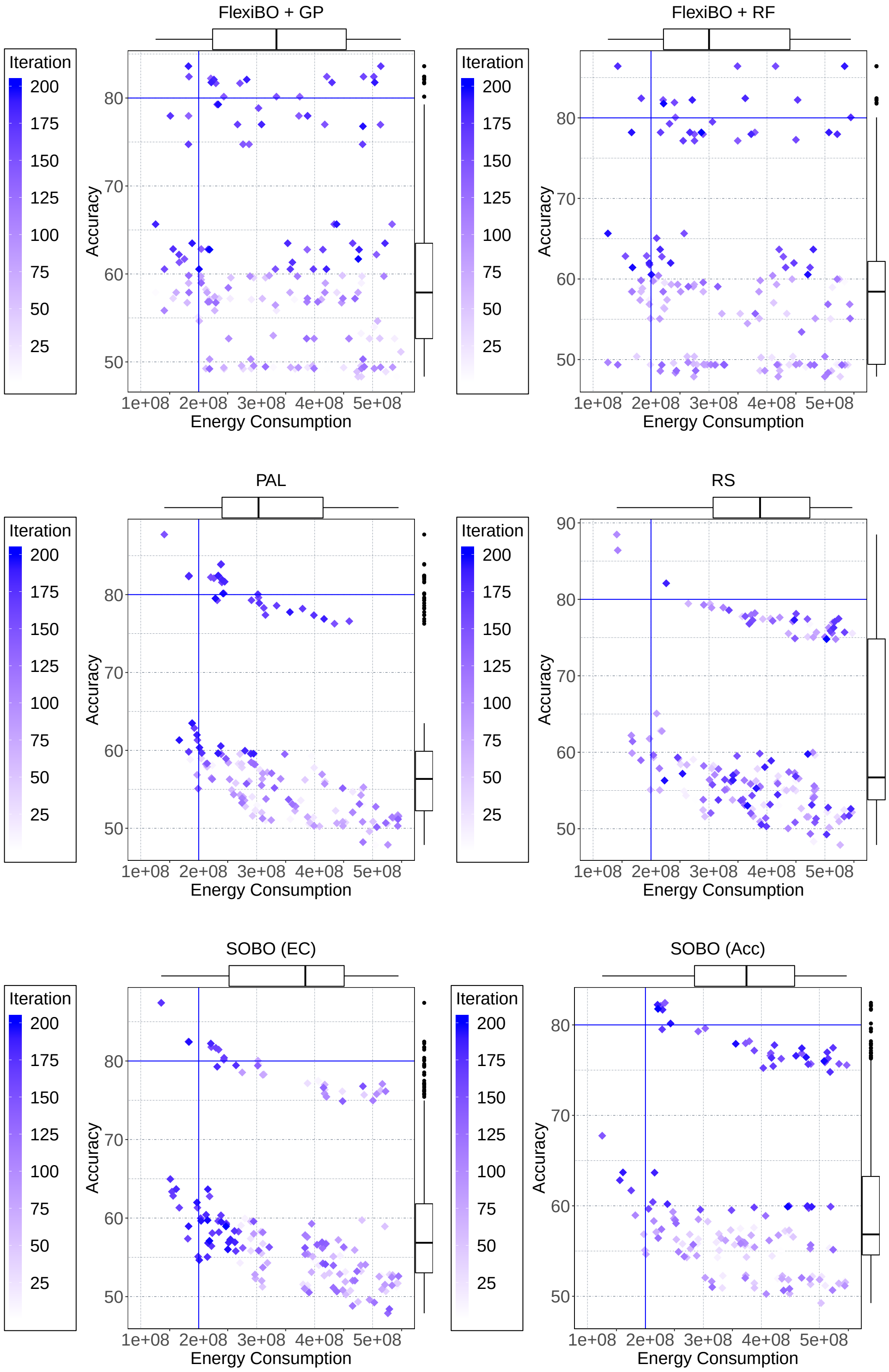

Figure 6: Exploration results in Deepspeech. It is desirable to have more evaluations to the top left half of the figure for optimization. FlexiBO outperforms other methods, finding more accurate and more efficient DNN architectures.

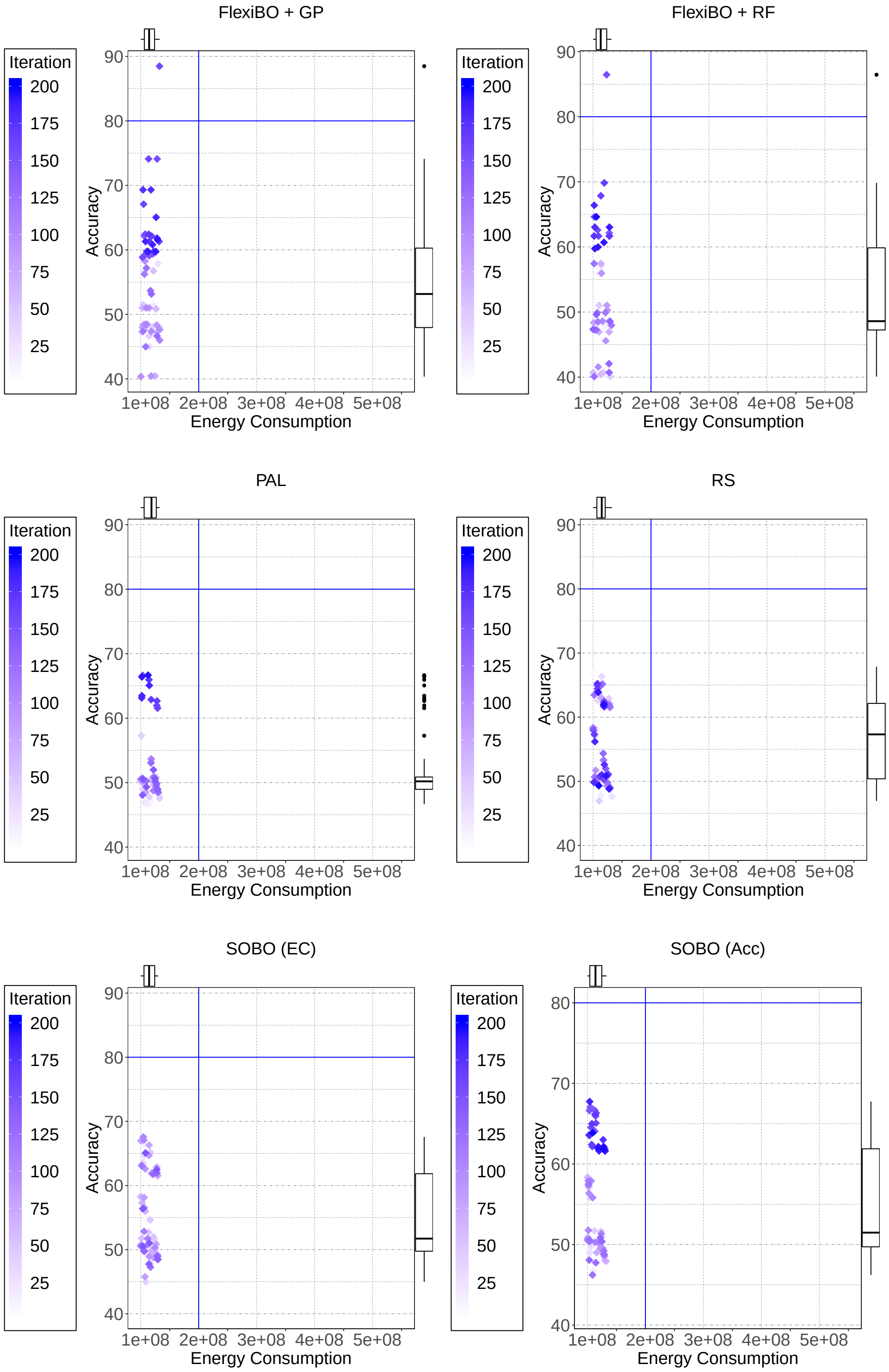

Figure 7: Exploration results in Lenet. It is desirable to have more evaluations to the top left half of the figure for optimization. FlexiBO outperforms other methods, finding more accurate and more efficient DNN architectures.

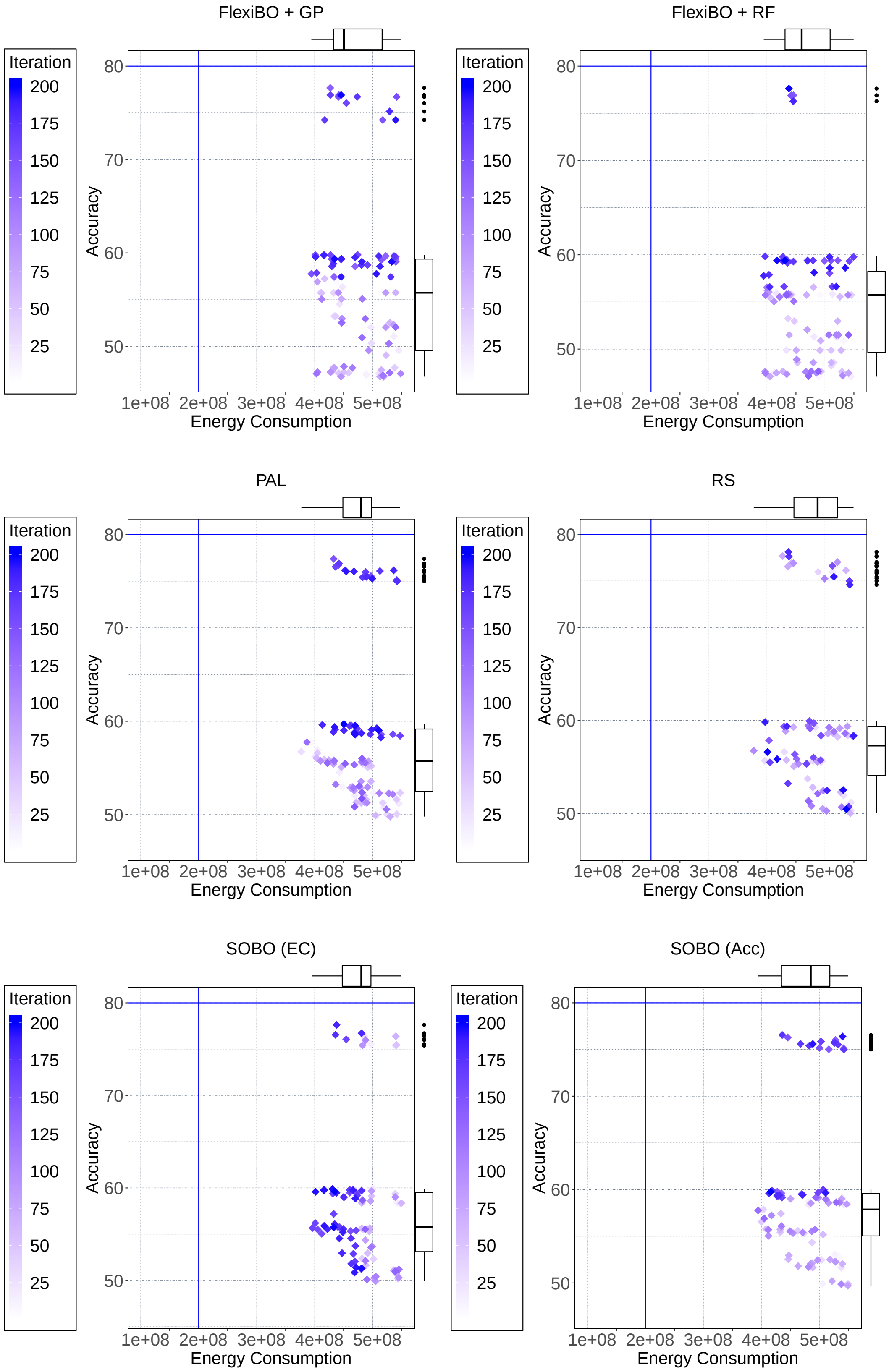

Figure 8: Exploration results in Mobilenet. It is desirable to have more evaluations to the top left half of the figure for optimization. FlexiBO outperforms other methods, finding more accurate and more efficient DNN architectures.

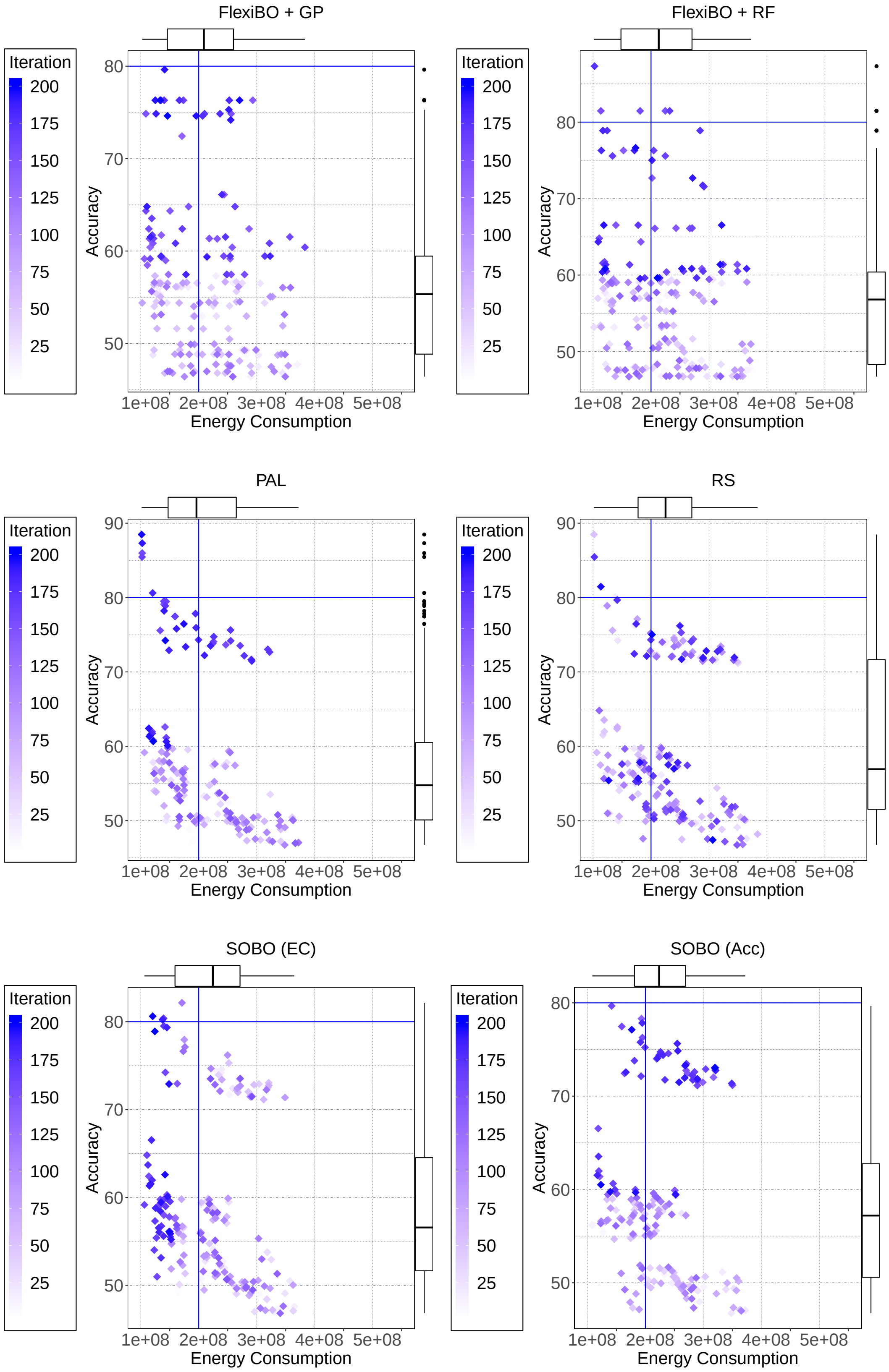

Figure 9: Exploration results in Resnet. It is desirable to have more evaluations to the top left half of the figure for optimization. FlexiBO outperforms other methods, finding more accurate and more efficient DNN architectures.

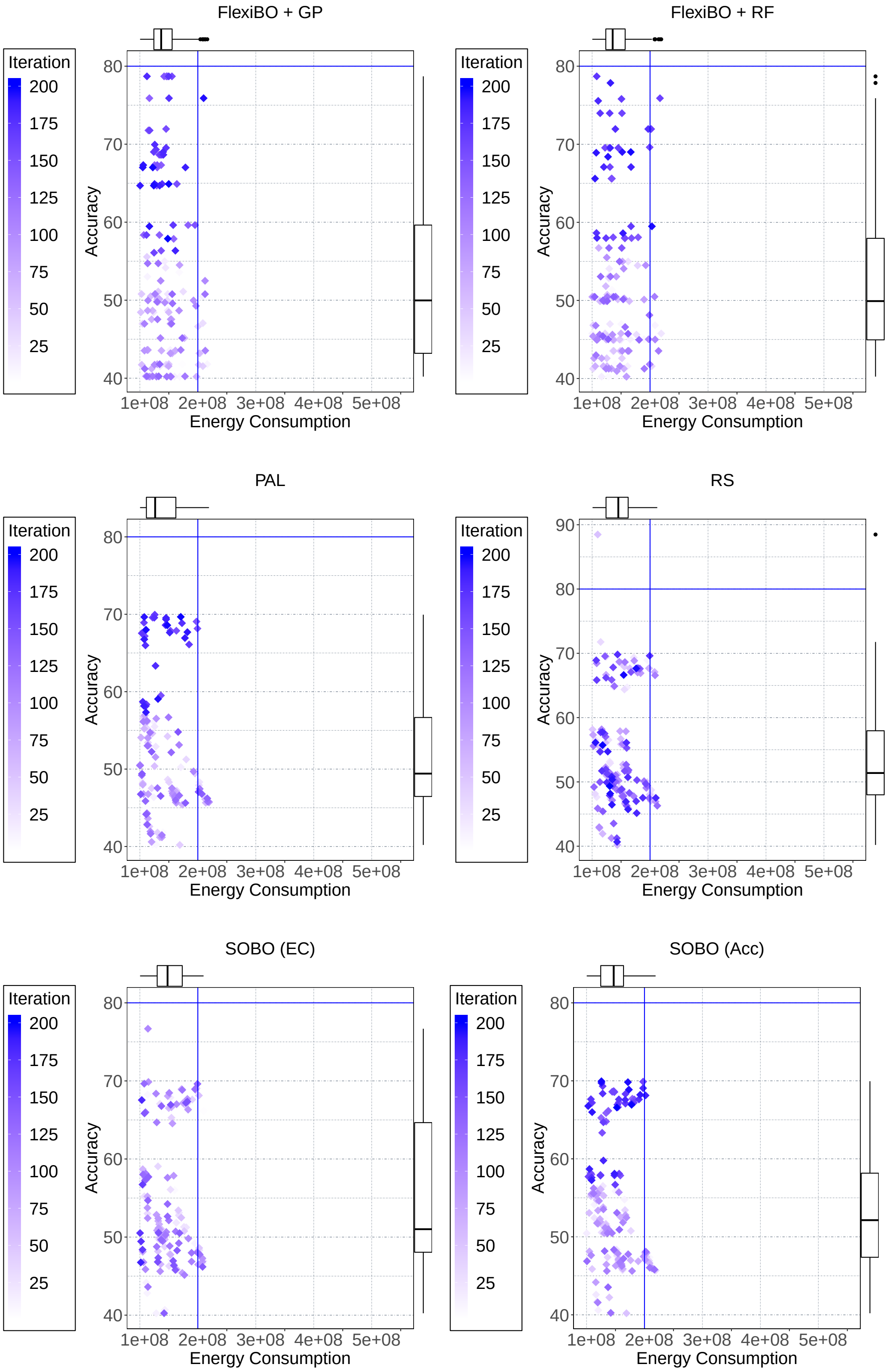

Figure 10: Exploration results in Squeezenet. It is desirable to have more evaluations to the top left half of the figure for optimization. FlexiBO outperforms other methods, finding more accurate and more efficient DNN architectures.