
Learning Debuggable Models Through Multi-Objective NAS

Supplemental Material

Anonymous¹

¹Anonymous Institution

A NAS-Bench-201 Overview

The NAS-Bench-201 search space is comprised of a macro skeleton and a searched cell. An overview is shown in Figure A1.

The first layer of the macro skeleton is a 3×3 convolutional layer with $F = 16$ filters followed by a batch normalization layer. This is followed by a stack of five searched cells ($F = 16$). A basic residual block ($F = 32$) with a stride of two proceeds the stacked cell block. The shortcut connection is a 2×2 2D average pooling layer followed by a 1×1 convolutional layer. These blocks are alternated, cutting the image dimensions in half and doubling the filters for each set of blocks. The end of the network is a 2D global average pooling layer followed by a fully-connected (dense) classification layer with a softmax activation.

The searched cell can be expressed as a directed acyclic graph where nodes represent data and edges represent operations. The set of operations consists of 3×3 convolutional blocks, 1×1 convolutional blocks, 3×3 average pooling, “zeroize” (equivalent to dropping the edge), and “skip-connect” (equivalent to the identity operator). Note that each convolutional block is comprised of convolution, a rectified linear (ReLU) activation, and batch normalization. All of the convolutions and pooling layers use SAME padding. To prevent cycles, each node is assigned a rank and can only connect to higher-rank nodes. Since there are $V = 4$ nodes in a cell and five operation candidates in the operation set, the total size of the search space is $5^{(\sum_{i=0}^{V-1} i)} = 15,625$ architectures. There are two issues with the search space definition, which the NAS-Bench-201 authors also point out. First, different architecture encodings can result in the same graph. Like the authors, we do not consider isomorphism in the evaluation of architectures¹. Second, architectures can be disconnected due to the zeroize operation. In this case, the mating operations are reapplied to produce valid offspring.

We represent an architecture in the search space as \mathbf{x}_i , a fixed-size list of integers of size $\sum_{i=0}^{V-1} i = 6$ with each element in the range $[1..V]$. Each element of this *encoding* represents (i) a specific operator or operators, such as a convolutional or max pooling layer with specific parameters (e.g. kernel size, strides, etc.), or the lack of an operator (identity) and (ii) how that operator is connected to additional operators in the computational graph.

B Reproducibility: Experiment Setup and Hyperparameters

Setup. We use a cosine annealing Loshchilov and Hutter (2017) learning rate schedule to decay the learning rate from 0.1 to 0 at the end of the last epoch. We also take half an epoch to warm up the learning rate from 0 to 0.1 at midway through the first epoch. Multiple seeds are set for reproducibility – see the code for the NumPy and TensorFlow seed-setting procedure. The seed for each run is stored in each raw result.

Data preprocessing. Recall that each raw image \mathfrak{X}_i has a height of H pixels, width of W pixels, and C color channels. We first scale the image by 255 to map the input domain from $[0..255] \subset \mathbb{N}_0$ to $[0, 1] \subset \mathbb{R}$. Then, z-score normalization is applied, i.e. the channel-wise mean of the full dataset \mathfrak{X} is subtracted from each \mathfrak{X}_i and the result of which is divided by the channel-wise standard deviation of \mathfrak{X} . The resulting data has channel-wise means of zero and standard deviations of one.

¹The NAS-Bench-201 authors remark that there are 6,466 architectures with unique topology in the search space due to isomorphisms brought about by the “skip-connect” and “zeroize” operations.

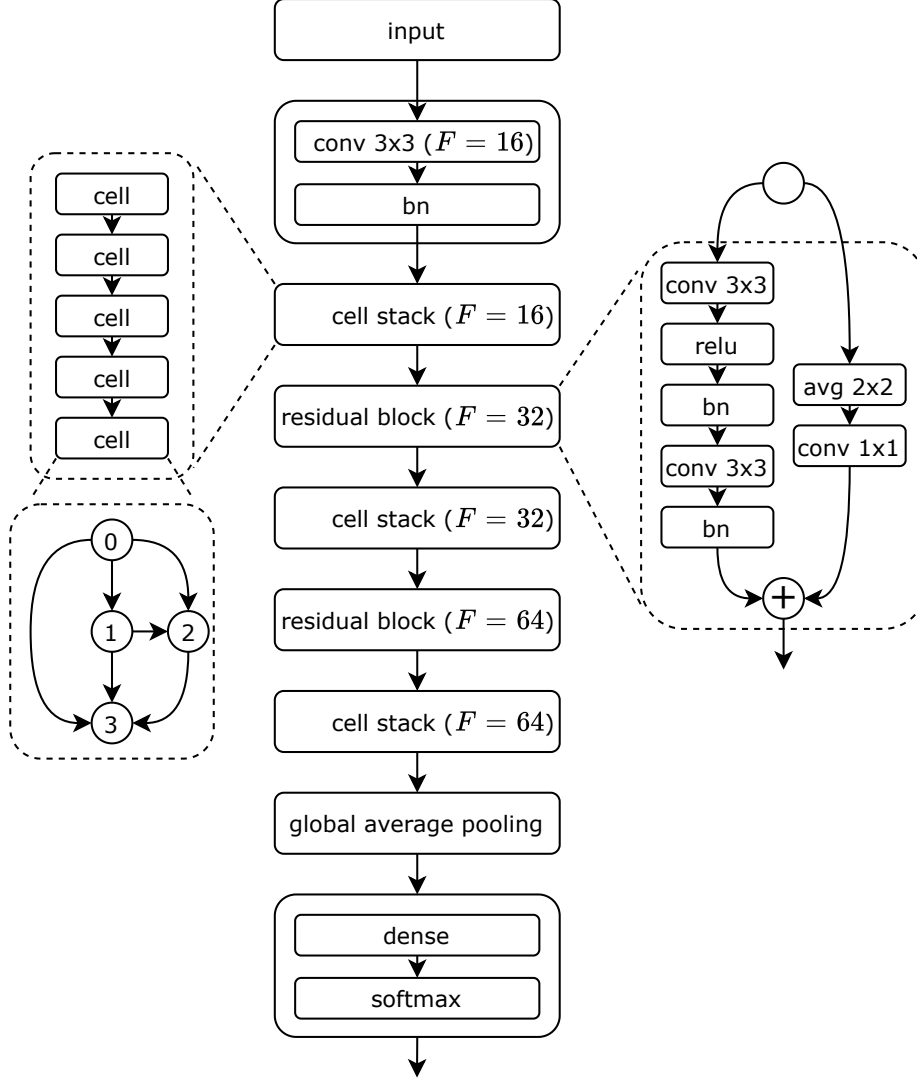


Figure A1: Visualization of the architectures generated by the search space. Note that the first convolutional layer in the main path and the average pooling layer in the shortcut path of each residual block has a stride of 2. conv: 2D convolutional layer. bn: batch normalization. relu: rectified linear (ReLU) activation layer. avg: 2D average pooling layer. F : the number of convolutional filters within each layer of a block.

Data augmentation. We zero-pad the left and right of each image with $\lceil H/8 \rceil$ pixels and the top and bottom of each image with $\lceil W/8 \rceil$ pixels. Then, each image is randomly cropped following a uniform distribution back to shape $H \times W \times C$. Next, the image is flipped horizontally with a probability of 0.5. The final augmentation applied is cutout Devries and Taylor (2017). Randomly centered rectangular windows with height $2\lceil H/8 \rceil$ and width $2\lceil W/8 \rceil$ are selected to be filled with zeros within the bounds of each image.

We do not allow offspring that have the same architecture as another offspring or a previously evaluated architecture. There are 6 integer variables in the optimization problem, so we set the probability of polynomial mutation per variable to $1/6$. Table B1 contains the summary of all hyperparameters used across the experiments.

	Hyperparameter	Value
Model Training	Loss	Cross Entropy
	Optimizer	SGD
	Learning Rate (LR)	0.1
	LR Schedule	Cosine Decay
	Nesterov	Yes
	Momentum	0.9
	Weight Decay	0.0005
	Batch Size	512
	Epochs	5 [*] , 12 [†] , 200 [‡]
	Data Normalization	Z-Score (Channel-Wise)
	Data Augmentation	See Text
NSGA-II	Population Size	64
	Sampling	Uniform Random
	Crossover	Simulated Binary $p = 0.9, \eta = 3$
	Mutation	Polynomial $p = 1/6, \eta = 3$

Table B1: Summary of hyperparameters used across each experiment. ^{*}MNIST; [†]CIFAR-10; [‡]ImageNet-16-120

ImageNet-16-120. The ImageNet-16-120 dataset, originally introduced in Chrabaszcz et al. (2017) and adapted by the NAS-Bench-201 benchmark Dong and Yang (2020), is a downsampled version of the ImageNet dataset. The dataset facilitates substantially faster experimentation while permitting satisfactory classification results – performance on ImageNet-16-120 has been shown to be indicative of performance across all of ImageNet. Each image in the dataset is resized to 16×16 pixels and only the data for the first 120 classes are retained.

Introspectability Regularizer. Introspectability can be used as a regularization term as it is differentiable. We add this as an auxiliary loss term and naively balance the term with cross entropy with a regularizer weight of 0.5 – this bounds introspectability to the range $[0, 1]$. Because we want to maximize introspectability, we take the cosine similarity instead of the distance. To accumulate activations grouped by classes in TensorFlow, the `tf.scatter_nd` operator is used in implementation. The remaining implementation is straightforward.

C Additional Activation Heat Maps

To gain a better qualitative understanding of the introspectability metric, we visualize the activations of the Pareto-optimal solutions of each task. In Figure C2, the solutions of the highest and lowest introspectability are shown for MNIST and ImageNet-16-120 (see main text for CIFAR-10). Within each layer, the activations are normalized using z-score normalization. The activations within each block per class are then averaged for the purpose of visualization. The differences between the highest- and lowest-scoring models are quite apparent; the activation patterns for each class in higher-scoring models have notable variance, whereas they are quite constant in lower-scoring models. The heat maps are best viewed digitally.

D Additional PCA Visualizations

The remaining 2D PCA activation visualizations are shown for the MNIST and CIFAR-10 tasks in Figure D1 and Figure D2, respectively. For MNIST, there is little discernible difference between the models with highest and lowest introspectability – this is expected as the difference between these introspectability scores is small (see the main text). For CIFAR-10, an apparent difference between the two models can be observed; the spread of points about the origin is more Gaussian with the higher-scoring model, which, empirically, should indicate a greater mean cosine distance between

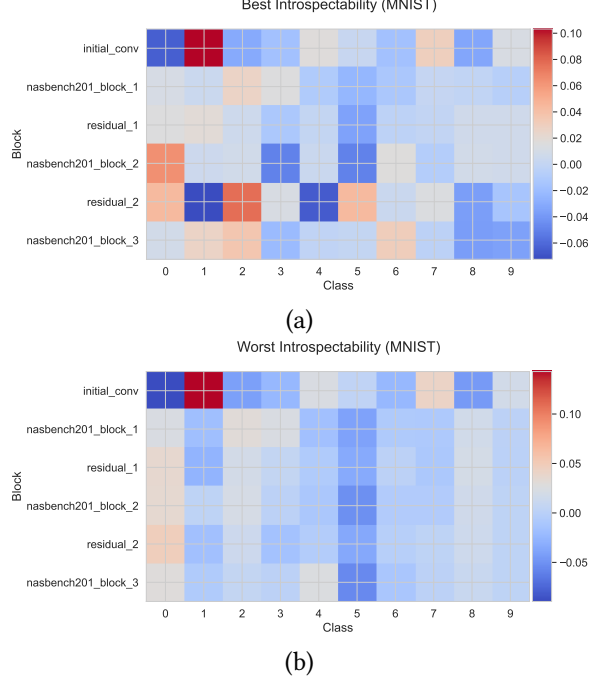


Figure C1: Mean activations heatmap of the models with (a) highest and (b) lowest introspectability on the MNIST task.

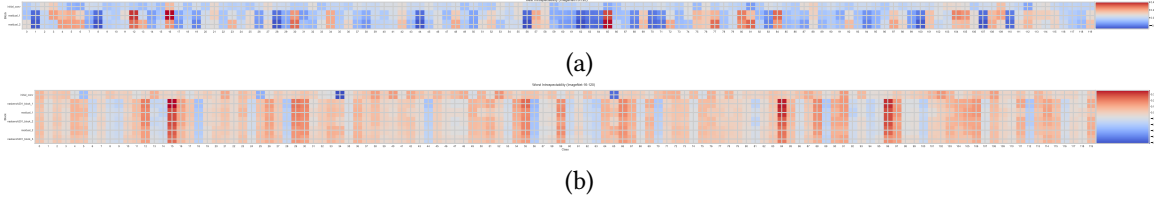


Figure C2: Mean activations heatmap of the model with highest introspectability on the ImageNet-16-120 task.

class representations. It is important to recall that the PCA projection eliminates thousands of dimensions used to represent activations. Naturally, this causes small changes in introspectability to be less apparent in visualizations.

E Analysis of Operator Selection

We show the operator-level normalized frequencies selected in the Pareto-optimal solutions of each task in Tables E1-E3. The 3x3 convolutions are most popular across all tasks, followed by either 3x3 average pooling or “zeroize” operators. The skip-connect and 1x1 convolutions are least frequent among these solutions.

F Frequentist Analysis of Motifs

We conduct analysis of the most common motifs across the Pareto-optimal solutions of each task, as shown in Tables F1-F3. Recall that the integer-coded cells are encoded as follows:

- 0: 3x3 Conv2D
- 1: 1x1 Conv2D
- 2: 3x3 AvgPool2D

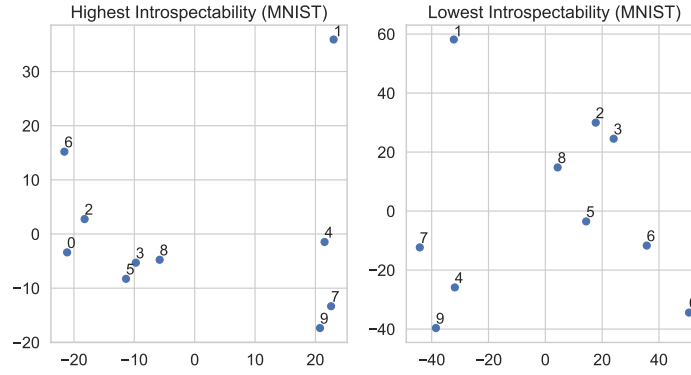


Figure D1: 2D PCA of the mean activations per class from the non-dominated models with highest and lowest introspectability on MNIST.

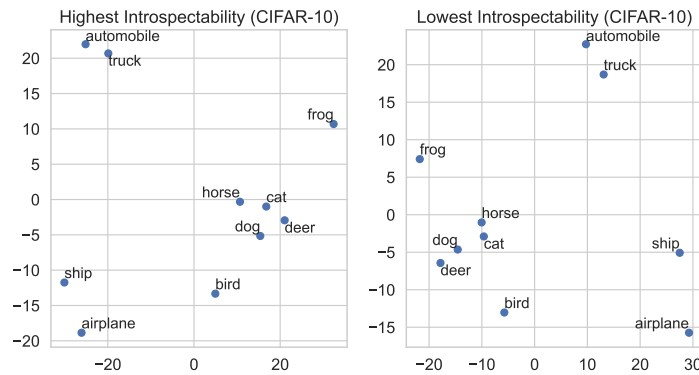


Figure D2: 2D PCA of the mean activations per class from the non-dominated models with highest and lowest introspectability on CIFAR-10.

- 3: Zeroize

95

- 4: Skip-Connect

96

We also use an asterisk (*) to match any operator. Within each table, the encodings of sizes 1 through 5 are shown alongside its normalized frequency of that size. Motifs of size 6 are not shown as we do not evaluate duplicate architectures (other than isomorphisms). The most common motifs reflect the operator frequencies discussed in the previous section. Interestingly, >67% of the Pareto-optimal solutions of each task all have a common motif of size 1, and >45% a common motif of size 2. This suggests that certain cell topologies exhibit inductive biases specific to the task.

97

98

99

100

101

102

G Comparing Motifs Across the Pareto Front

103

Motif Discovery.

104

1. Assemble the following data into a tabular structure: architecture encoding, accuracy, and introspectability for the Pareto front of the solutions
2. Sort the data by accuracy and then introspectability which results in data with ascending accuracy and descending introspectability
3. Record the count of each block for each architecture encoding
4. For each architecture encoding in the sorted data, enumerate all applicable motifs of size 1 to 5 (motifs of size 6 cannot exist as architectures are not evaluated multiple times). For example, some

105

106

107

108

109

110

111

Operation	Normalized Frequency
3x3 Conv2D	0.51515
3x3 AvgPool2D	0.16667
Zeroize	0.16667
1x1 Conv2D	0.09091
Skip-Connect	0.06061

Table E1: Frequency of operations of solutions in the Pareto front (normalized by total cell operations across the Pareto front models) on the MNIST task

Operation	Normalized Frequency
3x3 Conv2D	0.44444
Zeroize	0.24306
3x3 AvgPool2D	0.19097
Skip-Connect	0.06250
1x1 Conv2D	0.05903

Table E2: Frequency of operations of solutions in the Pareto front (normalized by total cell operations across the Pareto front models) on the CIFAR-10 task

architecture encoding $[a, b, c, d, e, f]$ has $\sum_{k=1}^5 \binom{6}{k}$ motifs, e.g. $[a, *, *, *, e, *]$ and $[*, b, c, *, *, f]$ 112
but not, say, $[*, b, c, *, *, e]$. This is nearly equivalent to its power set minus \emptyset and the original 113
sequence. An asterisk here implies a match with any other operator, and thus allows for the 114
comparison of motifs between different architectures 115

5. For each motif, compute the absolute value of the Spearman rank correlation coefficient between 116
the ranks of the solutions in the sorted data and whether each solution has the motif. If applicable, 117
the count of the operator is used instead of a simple indicator flag. The intuition here is that we 118
discover interesting architectures that demonstrably are favored more in one part of the Pareto 119
front than another, e.g. the high-accuracy vs. high-introspectability regions 120
6. In addition to each motif having a correlation score, we also record the support (the number of 121
solutions with the motif) and the motif size 122
7. Compute the Pareto front of the scored motifs (the costs being the correlation score, the support 123
and the motif size) to identify the most salient motifs. We heuristically eliminate motifs that 124
have support less than 3 or correlation less than 0.2 125

All figures from this discovery are present at the end of the appendices for the sake of space. 126

H Comparing Evolution of Single- and Multi-Objective Search 127

We illustrate the evolution of accuracy and introspectability of the models on the Pareto front over 128
each generation in Figure H1-Figure H3. Each figure contrasts single-objective with multi-objective 129
optimization to better understand the benefit of NSGA-II in our framework. Note that we do 130
not expect a strict increase in each objective at each generation, which would be expected for 131
population-level statistics, as opposed to statistics within the Pareto front. With single-objective 132
optimization, we can observe that solutions with higher introspectability tend to lie beyond the 133
95% confidence interval. This indicates fluke solutions, whereas multi-objective more confidently 134
produces higher-introspectability solutions. 135

Operation	Normalized Frequency
3x3 Conv2D	0.48039
3x3 AvgPool2D	0.21078
Zeroize	0.10784
Skip-Connect	0.10784
1x1 Conv2D	0.09314

Table E3: Frequency of operations of solutions in the Pareto front (normalized by total cell operations across the Pareto front models) on the ImageNet-16-120 task

Size	Normalized Frequency	Encoding
1	0.81818	[0 * * * *]
2	0.45455	[0 * * * 3 *]
2	0.45455	[0 * 0 * * *]
2	0.45455	[0 * * * * 0]
2	0.45455	[* * 0 * * 0]
3	0.36364	[0 * 0 * * 0]
4	0.27273	[0 * 0 4 * 0]
5	0.18182	[0 0 0 4 * 0]
5	0.18182	[0 * 0 4 3 0]

Table F1: Frequency of encodings of solutions in the Pareto front (normalized by the number of Pareto-optimal solutions) on the MNIST task. The top motif (motifs if tied frequency) for each size is shown only

I Comparing XNAS Accuracy with Related NAS Methods

136

We compare XNAS to other multi-objective approaches on the CIFAR-10 task. Building on the collected results and approach from Lu et al. (2019), we take the architecture with the best accuracy and increase the number of filters by a factor of four. We then perform full training on the CIFAR-10 dataset for 200 epochs. The comparison of results and methods is shown in Table I1. While XNAS does not achieve the best accuracy (nor was this the objective of this research), the result is still competitive, especially considering the trade-off between accuracy and introspectability.

137
138
139
140
141
142

J Additional Ablation Studies

143

We perform additional studies to understand the relationships between the objectives, accuracy and introspectability, and the generalization error, number of parameters, and training speed of architectures. Figure J1 demonstrates that introspectability and accuracy have an inverse relationship on the generalization error – this error increases with high-accuracy models and decreases with high-introspectability models. Likewise, the trend can be observed with the number of parameters and training speed as shown in Figures J2 and J3, respectively. These figures follow a similar trend as the number of parameters correlates with the number of FLOPs and thus the training time. As discussed in Section 4.4 of the main text, high-introspectability networks tend to have a more pooling layers whereas high-accuracy networks have more convolutional layers. This helps to explain the trends observed in the number of parameters. A takeaway from this analysis is that the trade-off between accuracy and introspectability also implies a trade-off in parameters (and FLOPs), training time, and generalization error.

144
145
146
147
148
149
150
151
152
153
154
155

Size	Normalized Frequency	Encoding
1	0.70833	[* * * 3 *]
2	0.50000	[* * * 3 0]
3	0.29167	[0 * 0 * 3 *]
3	0.29167	[* * 0 * 3 0]
4	0.18750	[0 * 0 * 3 0]
5	0.08333	[0 * 0 1 3 0]

Table F2: Frequency of encodings of solutions in the Pareto front (normalized by the number of Pareto-optimal solutions) on the CIFAR-10 task. The top motif (motifs if tied frequency) for each size is shown only

Size	Normalized Frequency	Encoding
1	0.67647	[* 0 * * * *]
2	0.47059	[* 0 * * * 0]
3	0.23529	[* 0 * 0 * 0]
4	0.11765	[2 0 * 1 * 0]
4	0.11765	[* 0 * 0 0 0]
4	0.11765	[0 0 * * 0 0]
4	0.11765	[0 0 * 0 * 0]
4	0.11765	[0 * * 0 0 0]
5	0.05882	[3 0 2 2 * 3]
5	0.05882	[2 0 1 1 * 0]
5	0.05882	[2 0 * 1 4 0]
5	0.05882	[2 0 4 0 0 *]
5	0.05882	[2 0 * 0 0 0]

Table F3: Frequency of encodings of solutions in the Pareto front (normalized by the number of Pareto-optimal solutions) on the ImageNet-16-120 task. The top motif (motifs if tied frequency, up to 5) for each size is shown only

K Model Debugging Experiments

156

Here, we study the ability of our activations calibration approach to correct bugs in models. We first demonstrate that there is a strong connection between the pairwise activation distances used in the formulation of introspectability and the ground truth confusion matrix. To make this comparison, the pairwise distances are negated as disentanglement (separation) between class representations is posited to correlate with confounding. Since the distance between the activations of a class and itself is 0, ideally, the distance between such and the activations of other classes is maximized. There is no information about ground truth available in computing pairwise distances, i.e. the computation is symmetrical and unconditioned. In turn, we compare this information to a confusion matrix folded along the diagonal. This means that element $x_{i,j}$, $i \neq j$ in the folded confusion matrix is equivalent to the sum $x_{i,j} + x_{j,i}$ in the original confusion matrix. On a higher level, each element $x_{i,j}$ is either the number of true positives for a class (when considering the diagonal), or the support of class i being predicted when class j were true and the support of the converse. To support this, we measure the correlation between the negated pairwise activation distances and the folded ground truth confusion matrix across Pareto optimal models trained on CIFAR-10. High-introspectability models achieve a correlation of $\rho = 0.85$ while low-introspectability models achieve a correlation

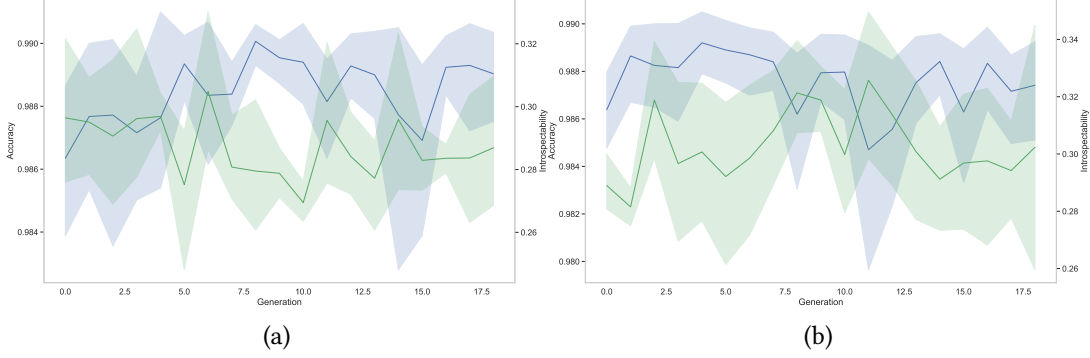


Figure H1: The mean accuracy (blue) and introspectability (green) of the Pareto front solutions each generation on the MNIST task. (a) single-objective; (b) multi-objective. The shaded region indicates the 95% confidence interval of solutions at each generation.

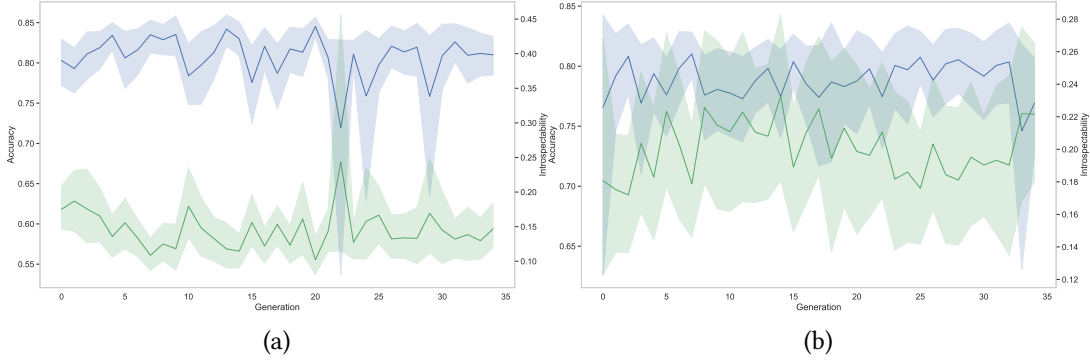


Figure H2: The mean accuracy (blue) and introspectability (green) of the Pareto front solutions each generation on the CIFAR-10 task. (a) single-objective; (b) multi-objective. The shaded region indicates the 95% confidence interval of solutions at each generation.

of $\rho = 0.59$. With this motivation, we demonstrate how model bugs can be identified and corrected in the following case study.

Case Study: Bug Identification and Correction. Figure K1 demonstrates for a random higher-introspectability model trained on CIFAR-10 that there is strong correlation between the negated pairwise activation distances and the ground truth confusion matrix folded along the diagonal ($\rho = 0.81$). Noticeably, the model confounds the classes 3 and 5, which is reflected in the pairwise activations as the smallest distance (largest negated distance).

With the bug in the model identified, we formulate a strategy to mitigate the issue. The key of our approach is to push the representations of classes 3 and 5 apart in order to reduce the confounding of one another. We accomplish this by using the introspectability regularizer approach with pairwise coefficients. The generalization of this to arbitrary pairs is formalized in Eq. (1).

$$\text{Introspectability}_{\text{reg}}(\mathcal{M}, \mathfrak{X}) = \frac{-1}{\binom{N_C}{2}} \sum_{c=1}^{N_C} \sum_{k=c+1}^{N_C} D(\bar{\Phi}^{(c)}, \bar{\Phi}^{(k)}) \times \omega_{i,j} \quad (1)$$

where $\omega_{i,j}$ is a weight for each class pair (i, j) . With every $\omega_{i,j} = 1$ this is equivalent to the untargated introspectability regularizer. If the aim is to target all confounded predictions, one can set all $\omega_{i,j}$ proportionally to the pairwise activation distances (or folded confusion matrix). However, we target a single pair in this case study. In our experiment, the model is trained with the regularization term for an additional 5 epochs, a learning rate of 0.001, $\omega_{3,5} = 25$, and all other

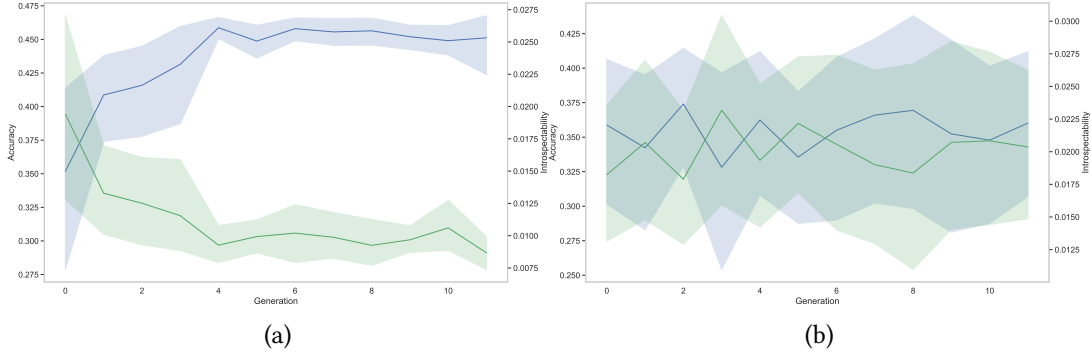


Figure H3: The mean accuracy (blue) and introspectability (green) of the Pareto front solutions each generation on the ImageNet-16-120 task. (a) single-objective; (b) multi-objective. The shaded region indicates the 95% confidence interval of solutions at each generation.

Method	Error	Other Objective	Compute
PPP-Net Dong et al. (2018)	4.36%	FLOPs, # parameters, or inference time	Nvidia Titan X
MONAS Hsu et al. (2018)	4.34%	Power	Nvidia 1080 Ti
NSGA-Net Lu et al. (2019)	3.85%	FLOPs	Nvidia 1080 Ti 8 GPU Days
XNAS	4.45%	Introspectability	Nvidia Tesla P100 6 GPU Days

Table I1: Multi-objective methods for CIFAR-10 (best accuracy for each method). Table adapted from Lu et al. (2019)

$\omega_{i,j} = 1$. The results are visualized in Figure K2. The approach is stronger in identifying bugs than mitigating them, although there is improvement without significant degradation of accuracy ($\pm 0.6\%$ across 10 trials). We leave the tuning of hyperparameters and alternative weighting schemes to future exploration.

L Extended Background

DNN Inspection within Explainable AI (XAI). The opaque nature of deep neural networks (DNNs) has ultimately led to the sub-field of explainable AI (XAI) Gunning (2019), which was denominated in 2016 by DARPA, although relevant work predates this by years. Relevant to the subject matter of this work are XAI methods of DNN inspection. This suite of methods enables the debugging of model behavior, the detection of dataset errors, and the development of adversarial attacks. The authors of Koh and Liang (2017) scale influence functions, a robust statistics method, to DNNs to understand the effect of training points on a prediction. DNN visualization tools have been proposed to provide qualitative modes of analysis. Notably, Erhan et al. (2009); Yosinski et al. (2015) provide tools for visualizations by gradient ascent, deconvolution for highlighting input images, and discovering preferred input patterns for each class. Probing-based methods aim to qualify the role of DNN internal elements (neurons, latent representations, etc.). In Kim et al. (2018); Bau et al. (2020), methods are proposed to relate DNN internals to semantic concepts, such as textures, shapes, colors, or even people. Another approach introduced in Ghorbani and Zou (2020) is to use Shapley values from game theory to quantify the influence each neuron has on overall DNN error.

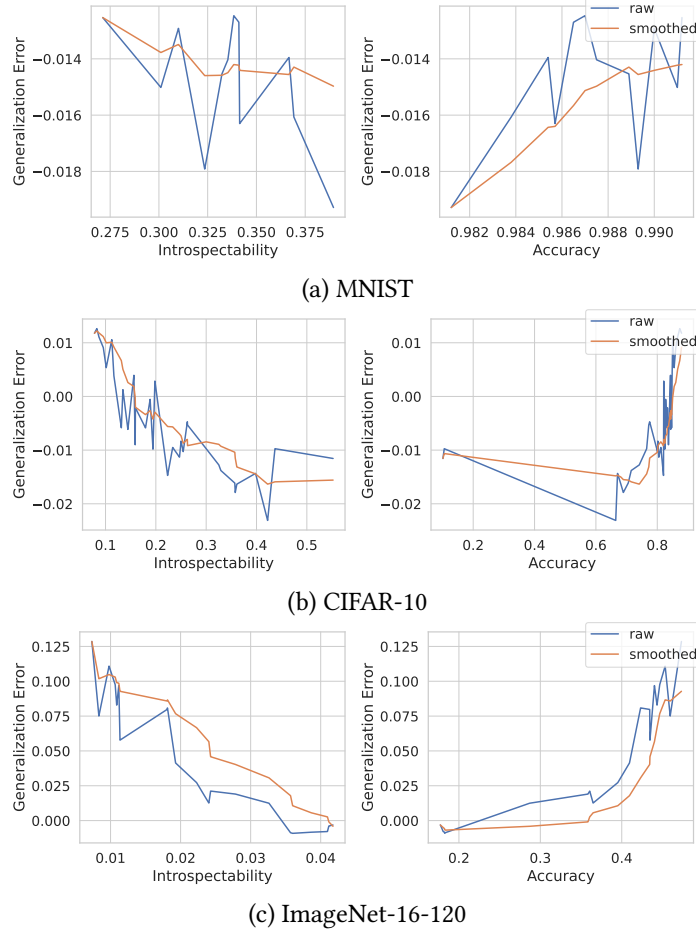


Figure J1: The effect of introspectability and accuracy on generalization error across the tasks.

More related to our work are those related to disentanglement, i.e. the separation of concept- or class-relevant information in a network. For instance, Zhang et al. (2018) proposes the learning of interpretable CNN filters by coercing feature maps to resemble hand-crafted templates. Moreover, the variational autoencoder (VAE) Kingma and Welling (2014) has been extended to produce a disentangled latent space by regularizing the bottleneck layer Higgins et al. (2017). In contrast, we optimize for DNNs with disentangled internal representations of classes without explicit constraints on the loss, modifications to the architecture, or hand-crafted activation patterns. This also allows for the use of non-differentiable objectives.

M Extended Crossover and Mutation Details

Mating comprises two core operations: crossover and mutation. The *crossover* operator produces offspring by combining the encodings of two parents. The operator combines the building blocks between successful parents to exploit the *implicit parallelism* of population-based search (Holland, 1992). Due to the integer-based encoding that we employ in this work, we elect to use simulated binary crossover (Deb et al., 2007), which uses a probability density function to simulate the single-point crossover of binary-coded genetic algorithms. The *mutation* operator produces offspring by modulating one or more of the variables of a single parent. We specifically select polynomial mutation, which follows the same probability distribution as simulated binary crossover. Both crossover and mutation also have a parameter p that controls the probability that the respective operator is applied to a member of the population.

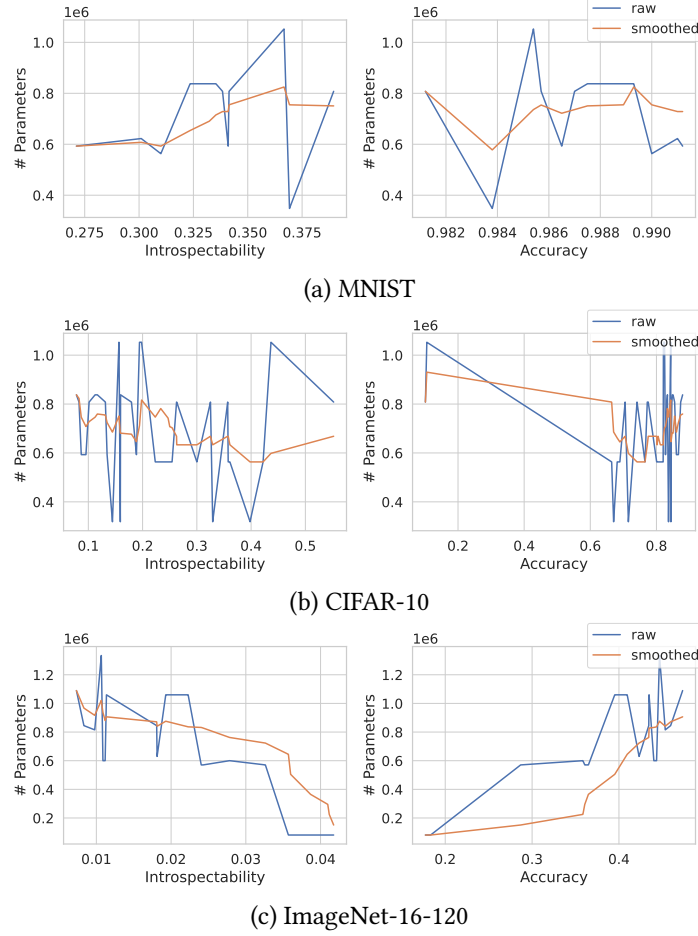


Figure J2: The effect of introspectability and accuracy on the number of parameters across the tasks.

N Dataset Information

MNIST. The MNIST LeCun et al. (2010) dataset is available at <https://yann.lecun.com/exdb/mnist/>. Yann LeCun and Corinna Cortes hold the copyright of MNIST dataset, which is a derivative work from the original NIST datasets. MNIST dataset is made available under the terms of the Creative Commons Attribution-Share Alike 3.0 license. The dataset does not contain personally identifiable information or offensive content.

CIFAR-10. The CIFAR-10 Krizhevsky (2009) dataset is available at <https://www.cs.toronto.edu/%7Ekriz/cifar.html>. There is no license provided for the dataset. The dataset does not contain personally identifiable information or offensive content.

ImageNet-16-120. The ImageNet-16-120 (Dong and Yang, 2020) dataset is a subset of ImageNet which is available at <https://www.image-net.org/download.php>. The terms of using the ImageNet dataset are also outlined at <https://www.image-net.org/download.php>. ImageNet does not own the copyright to the images, rather it compiles a list of web images per synset as described at <https://www.image-net.org/about.php>. The dataset is known to contain some personally identifiable information and potentially offensive content – see Asano et al. (2021) for further details.

O Scaling Up XNAS

We scale XNAS to clusters comprising an arbitrary number of compute nodes using the distributed framework, Ray (Moritz et al., 2018). Given a set of M nodes $\{n_i\}_{i=1}^M$, n_1 is treated as a head node

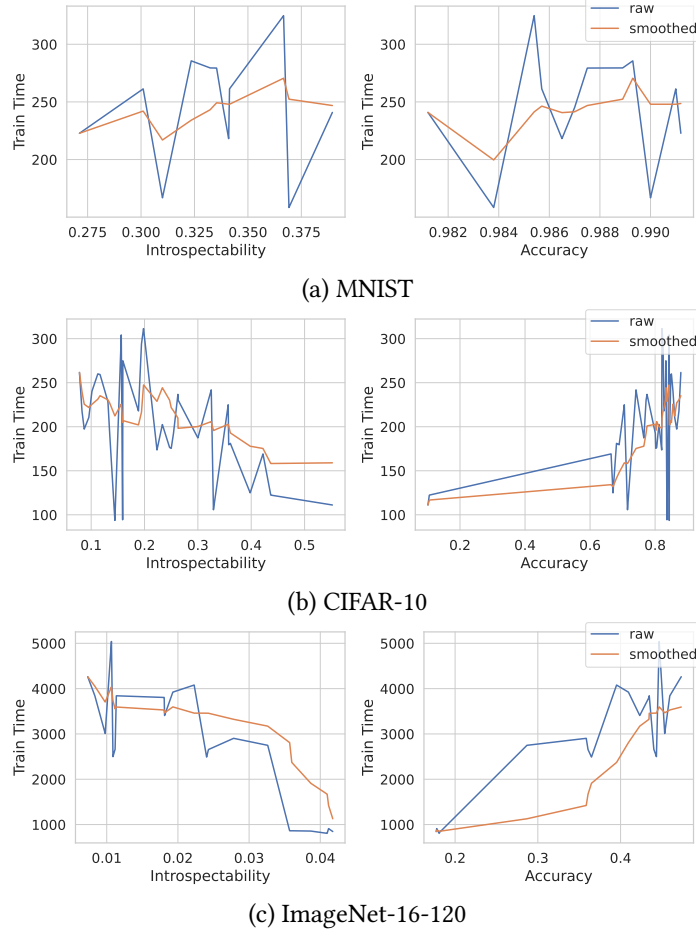


Figure J3: The effect of introspectability and accuracy on the training speed across the tasks.

that is responsible for running the core NSGA-II optimization loop and the core Ray server. The remaining nodes $\{n_i\}_{i=2}^M$ are configured as workers available to train and evaluate architectures on a dataset. When a new generation of architectures is created, each offspring is submitted for fitness evaluation to a queue by the head node. Each job in the queue is offloaded to a free worker until all workers complete their jobs and the queue is empty. The head node n_1 also is treated as an additional worker if it has free resources. Each worker node can execute in parallel as many jobs as it has GPUs.

P FLOPs Analysis

Here, we show the relationship between FLOPs and the two objectives, accuracy and introspectability. While fewer convolutional layers are preferred by the introspectability metric, there is not a direct relationship between either objective and FLOPs. Figure P1 shows the relationship for the MNIST, CIFAR-10, and ImageNet-16-120 datasets for the Pareto fronts discovered by multi-objective XNAS.

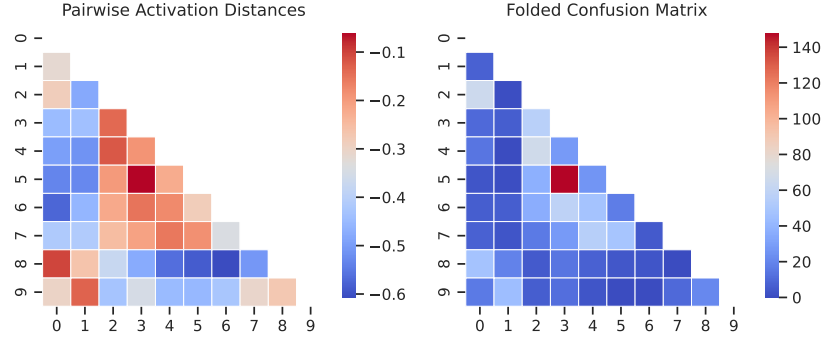


Figure K1: Heat maps of (left) the negated pairwise activation distances as part of the introspectability computation, and (right) the ground truth confusion matrix folded along the diagonal. The heat maps are shown for a random model trained on CIFAR-10. As can be seen, the model confounds the classes 3 and 5, which is reflected in the distance between activations.

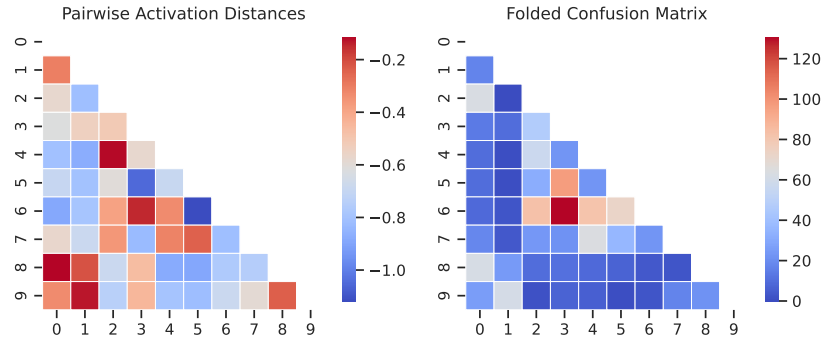
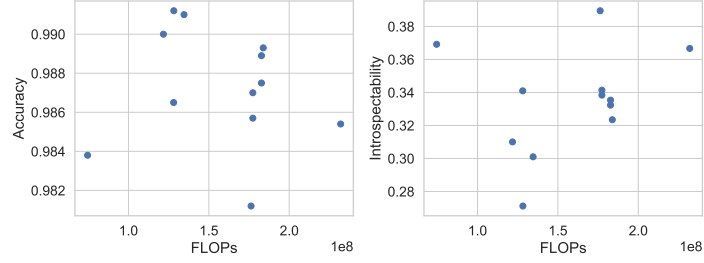
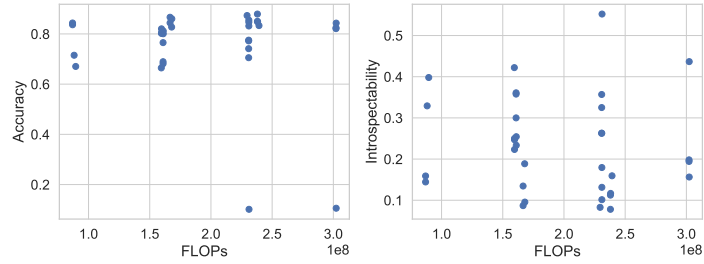


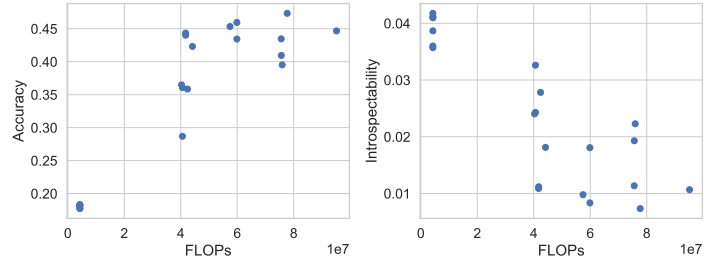
Figure K2: Heat maps after the correction procedure is run of (left) the negated pairwise activation distances as part of the introspectability computation, and (right) the ground truth confusion matrix folded along the diagonal. Note the difference in color scale from Figure K1.



(a) MNIST



(b) CIFAR-10



(c) ImageNet-16-120

Figure P1: Relationship between FLOPs and the two objectives, accuracy and introspectability, on the MNIST, CIFAR-10, and ImageNet-16-120 datasets for the Pareto fronts discovered by multi-objective XNAS.

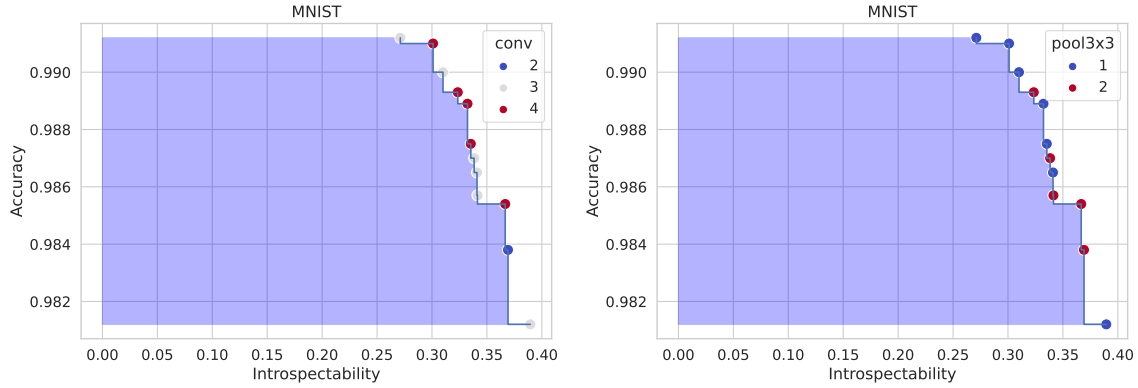


Figure Q1: The Pareto front for the MNIST task with solutions colored by the number of convolutional (left) and pooling (right) layers.

Q Motif Figures

257

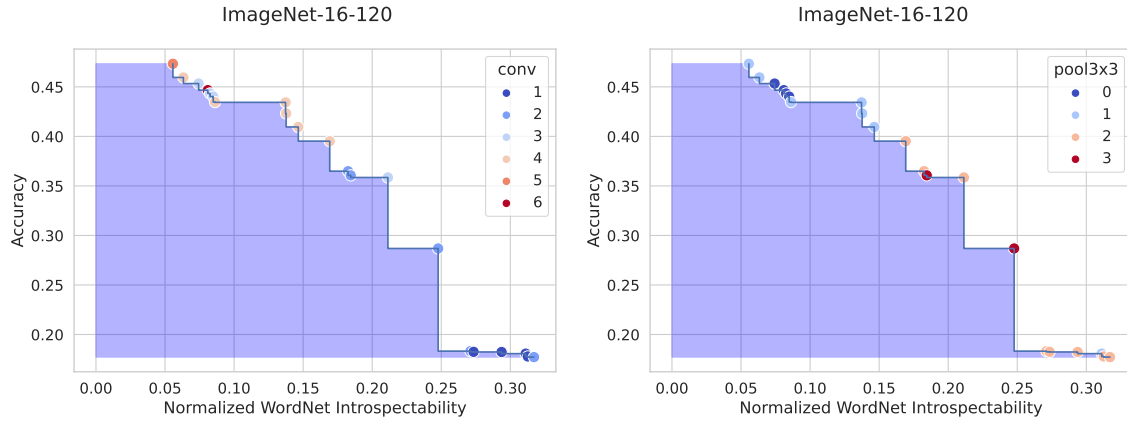


Figure Q2: The Pareto front for the ImageNet-16-120 task with solutions colored by the number of convolutional (left) and pooling (right) layers.

References

258

Asano, Y. M., Rupprecht, C., Zisserman, A., and Vedaldi, A. (2021). PASS: An imagenet replacement for self-supervised pretraining without humans. In *Proceedings of the NeurIPS Track on Datasets and Benchmarks 2021*, pages 1–26. 259 260 261

Bau, D., Zhu, J., Strobel, H., Lapedriza, À., Zhou, B., and Torralba, A. (2020). Understanding the role of individual units in a deep neural network. *Proc. Natl. Acad. Sci. USA*, 117(48):30071–30078. 262 263

Chrabaszcz, P., Loshchilov, I., and Hutter, F. (2017). A downsampled variant of imagenet as an alternative to the CIFAR datasets. *CoRR*, abs/1707.08819. 264 265

Deb, K., Sindhya, K., and Okabe, T. (2007). Self-adaptive simulated binary crossover for real-parameter optimization. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, GECCO '07*, page 1187–1194, New York, NY, USA. Association for Computing Machinery. 266 267 268 269

Devries, T. and Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. *CoRR*, abs/1708.04552. 270 271

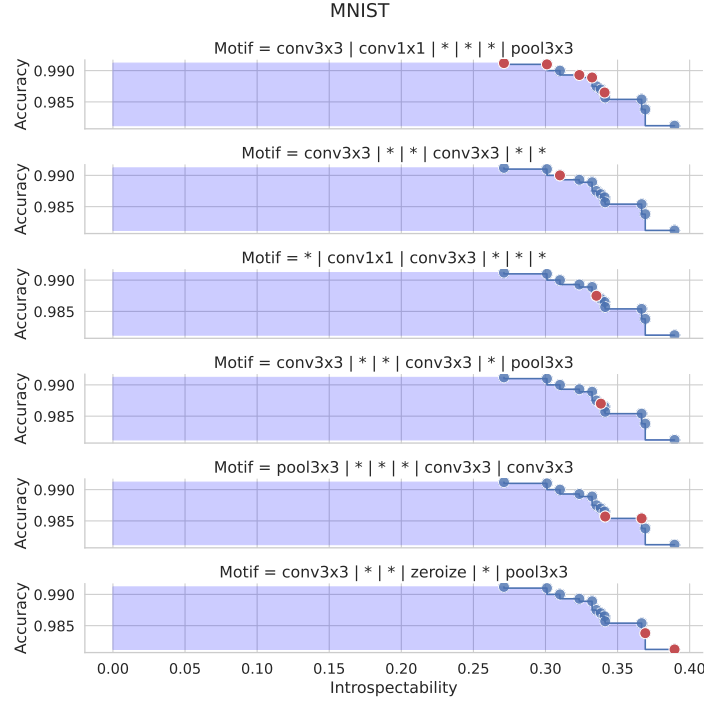


Figure Q3: All discovered motifs among the Pareto optimal solutions on the MNIST task. See text for description of the motif discovery process. Each red solution indicates that its architecture has the motif shown in the sub-plot title. The remaining solutions are shown in blue. For the N/A plot, none of the discovered motifs apply to the architecture.

- Dong, J., Cheng, A., Juan, D., Wei, W., and Sun, M. (2018). PPP-net: Platform-aware progressive search for Pareto-optimal neural architectures. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net.
- Dong, X. and Yang, Y. (2020). Nas-bench-201: Extending the scope of reproducible neural architecture search. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.
- Erhan, D., Bengio, Y., Courville, A., and Vincent, P. (2009). Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1.
- Ghorbani, A. and Zou, J. Y. (2020). Neuron shapley: Discovering the responsible neurons. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.
- Gunning, D. (2019). Darpa’s explainable artificial intelligence (xai) program. In *Proceedings of the 24th International Conference on Intelligent User Interfaces, IUI ’19*, page ii, New York, NY, USA. Association for Computing Machinery.
- Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017*. OpenReview.net.

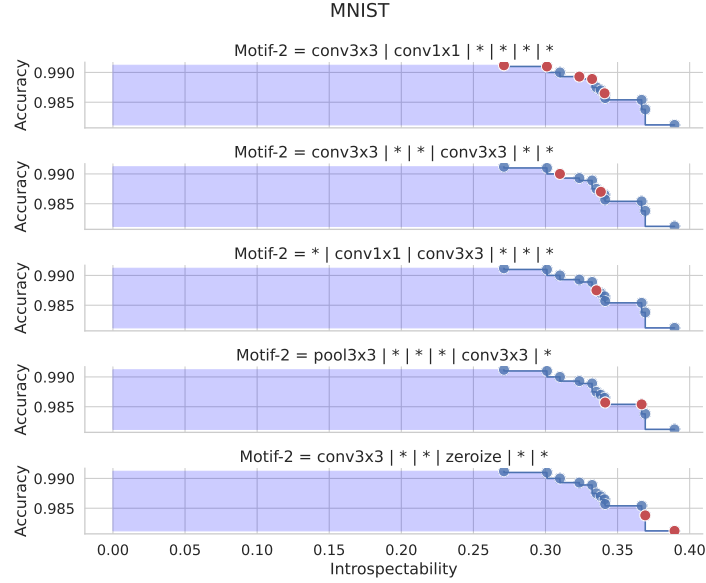


Figure Q4: Discovered motifs of size 2 among the Pareto optimal solutions on the MNIST task. See text for description of the motif discovery process. Each red solution indicates that its architecture has the motif shown in the sub-plot title. The remaining solutions are shown in blue. For the N/A plot, none of the discovered motifs apply to the architecture.

- Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press.
- Hsu, C., Chang, S., Juan, D., Pan, J., Chen, Y., Wei, W., and Chang, S. (2018). MONAS: multi-objective neural architecture search using reinforcement learning. *CoRR*, abs/1806.10332.
- Kim, B., Wattenberg, M., Gilmer, J., Cai, C. J., Wexler, J., Viégas, F. B., and Sayres, R. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In Dy, J. G. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2673–2682. PMLR.
- Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y., editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.
- Koh, P. W. and Liang, P. (2017). Understanding black-box predictions via influence functions. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR.
- Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto.
- LeCun, Y., Cortes, C., and Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, 2.
- Loshchilov, I. and Hutter, F. (2017). SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

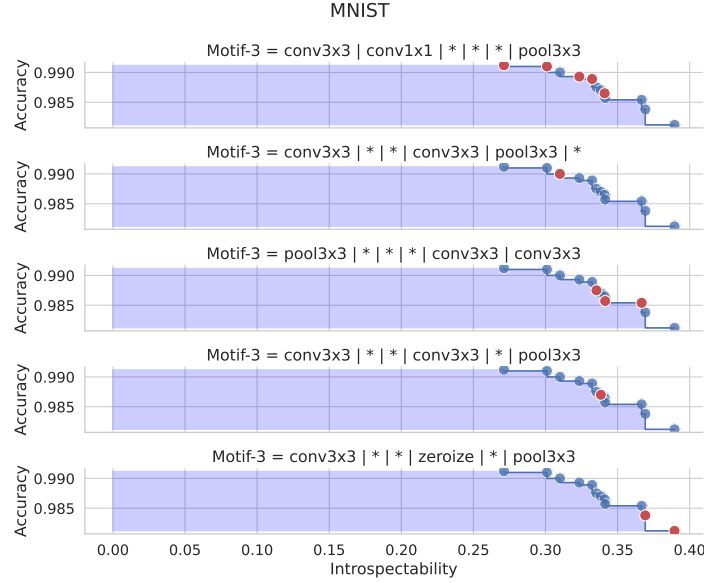


Figure Q5: Discovered motifs of size 3 among the Pareto optimal solutions on the MNIST task. See text for description of the motif discovery process. Each red solution indicates that its architecture has the motif shown in the sub-plot title. The remaining solutions are shown in blue. For the N/A plot, none of the discovered motifs apply to the architecture.

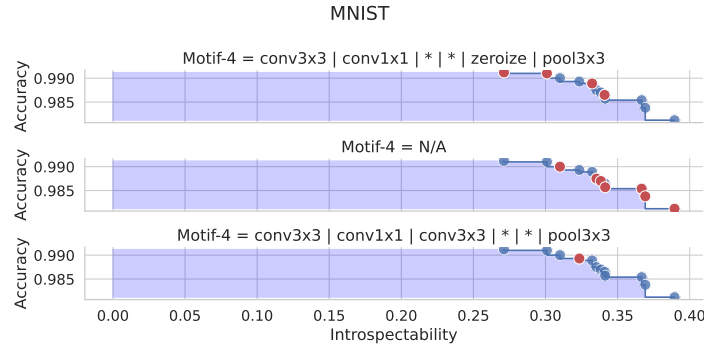


Figure Q6: Discovered motifs of size 4 among the Pareto optimal solutions on the MNIST task. See text for description of the motif discovery process. Each red solution indicates that its architecture has the motif shown in the sub-plot title. The remaining solutions are shown in blue. For the N/A plot, none of the discovered motifs apply to the architecture.

- Lu, Z., Whalen, I., Boddeti, V., Dhebar, Y. D., Deb, K., Goodman, E. D., and Banzhaf, W. (2019). NSGA-Net: neural architecture search using multi-objective genetic algorithm. In Auger, A. and Stützle, T., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 419–427. ACM. 314 315 316 317
- Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M. I., and Stoica, I. (2018). Ray: A distributed framework for emerging AI applications. In Arpaci-Dusseau, A. C. and Voelker, G., editors, *13th USENIX Symposium on Operating Systems Design and Implementation*, pages 561–577. USENIX Association. 318 319 320 321
- Yosinski, J., Clune, J., Nguyen, A. M., Fuchs, T. J., and Lipson, H. (2015). Understanding neural networks through deep visualization. *CoRR*, abs/1506.06579. 322 323

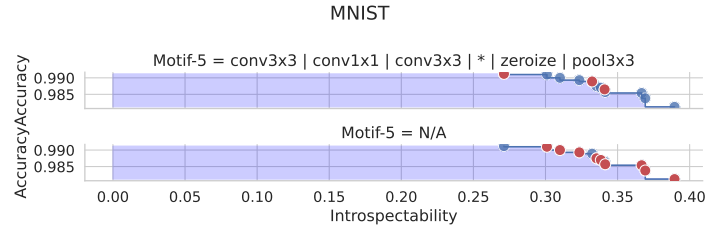


Figure Q7: Discovered motifs of size 5 among the Pareto optimal solutions on the MNIST task. See text for description of the motif discovery process. Each red solution indicates that its architecture has the motif shown in the sub-plot title. The remaining solutions are shown in blue. For the N/A plot, none of the discovered motifs apply to the architecture.

Zhang, Q., Wu, Y. N., and Zhu, S. (2018). Interpretable convolutional neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8827–8836. Computer Vision Foundation / IEEE Computer Society.

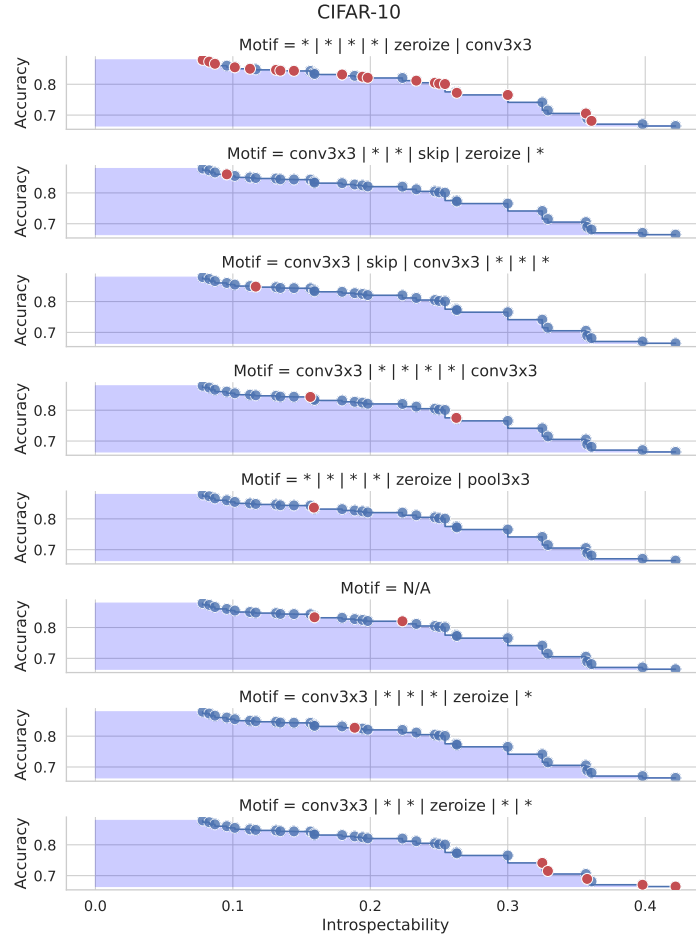


Figure Q8: All discovered motifs among the Pareto optimal solutions on the CIFAR-10 task. See text for description of the motif discovery process. Each red solution indicates that its architecture has the motif shown in the sub-plot title. The remaining solutions are shown in blue. For the N/A plot, none of the discovered motifs apply to the architecture.

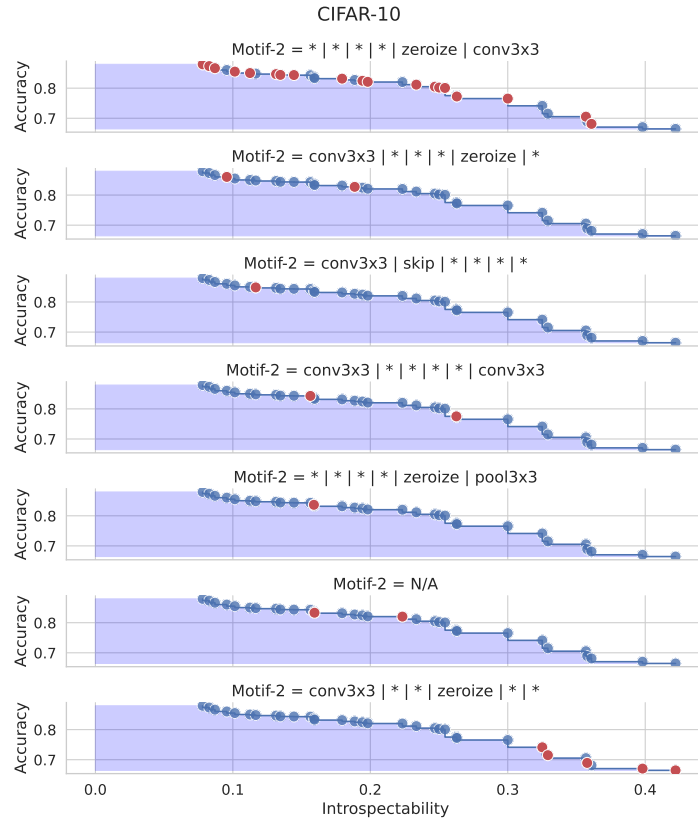


Figure Q9: Discovered motifs of size 2 among the Pareto optimal solutions on the CIFAR-10 task. See text for description of the motif discovery process. Each red solution indicates that its architecture has the motif shown in the sub-plot title. The remaining solutions are shown in blue. For the N/A plot, none of the discovered motifs apply to the architecture.

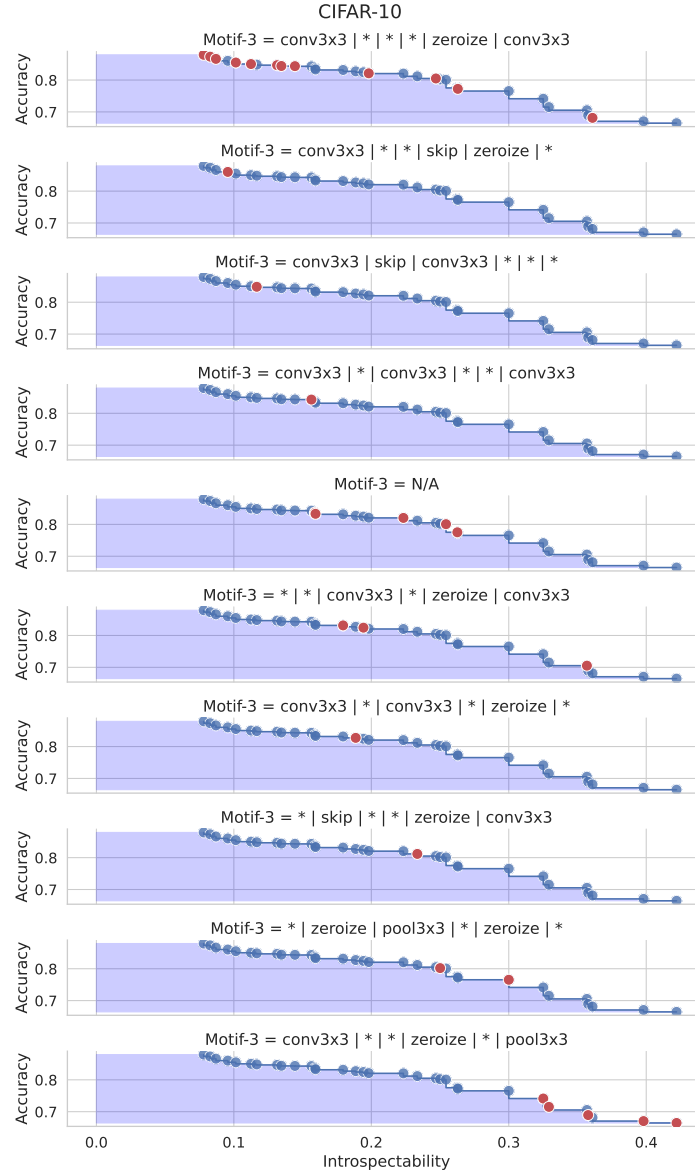


Figure Q10: Discovered motifs of size 3 among the Pareto optimal solutions on the CIFAR-10 task. See text for description of the motif discovery process. Each red solution indicates that its architecture has the motif shown in the sub-plot title. The remaining solutions are shown in blue. For the N/A plot, none of the discovered motifs apply to the architecture.

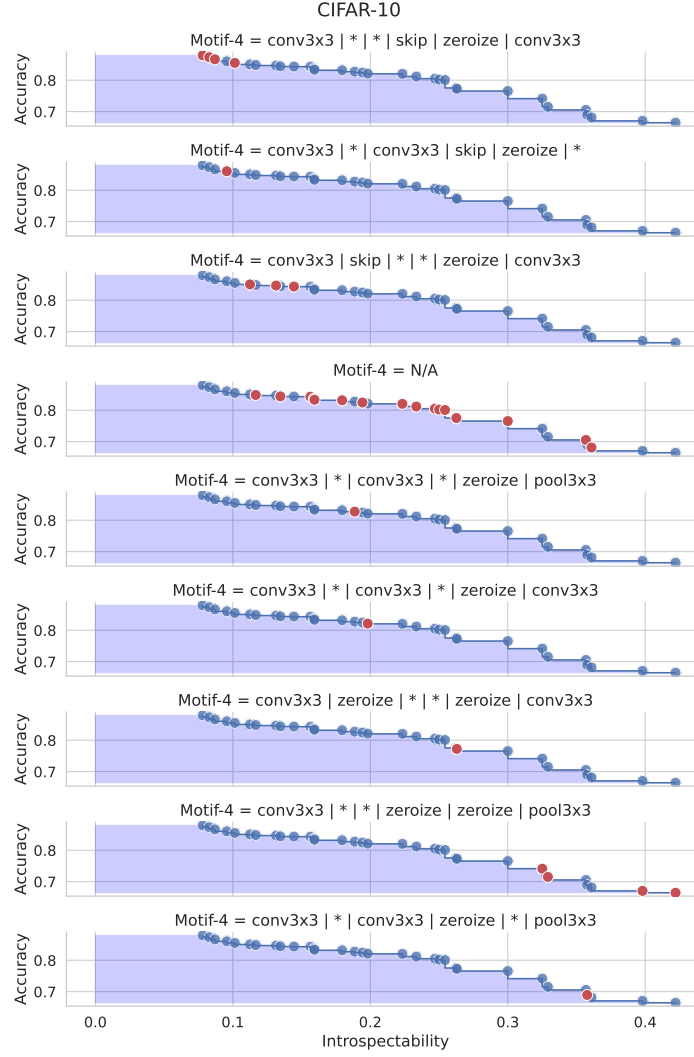


Figure Q11: Discovered motifs of size 4 among the Pareto optimal solutions on the CIFAR-10 task. See text for description of the motif discovery process. Each red solution indicates that its architecture has the motif shown in the sub-plot title. The remaining solutions are shown in blue. For the N/A plot, none of the discovered motifs apply to the architecture.

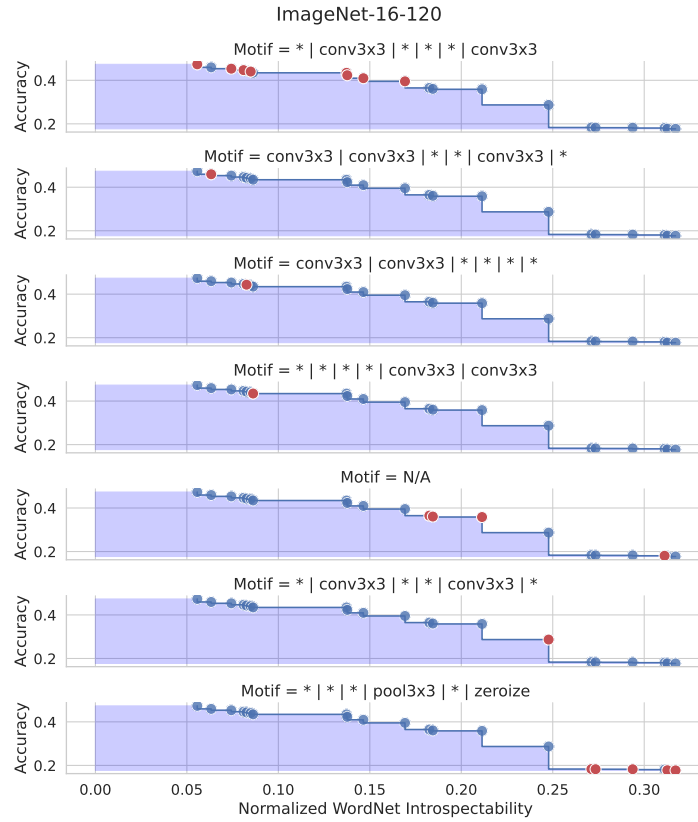


Figure Q12: All discovered motifs among the Pareto optimal solutions on the ImageNet-16-120 task. See text for description of the motif discovery process. Each red solution indicates that its architecture has the motif shown in the sub-plot title. The remaining solutions are shown in blue. For the N/A plot, none of the discovered motifs apply to the architecture.

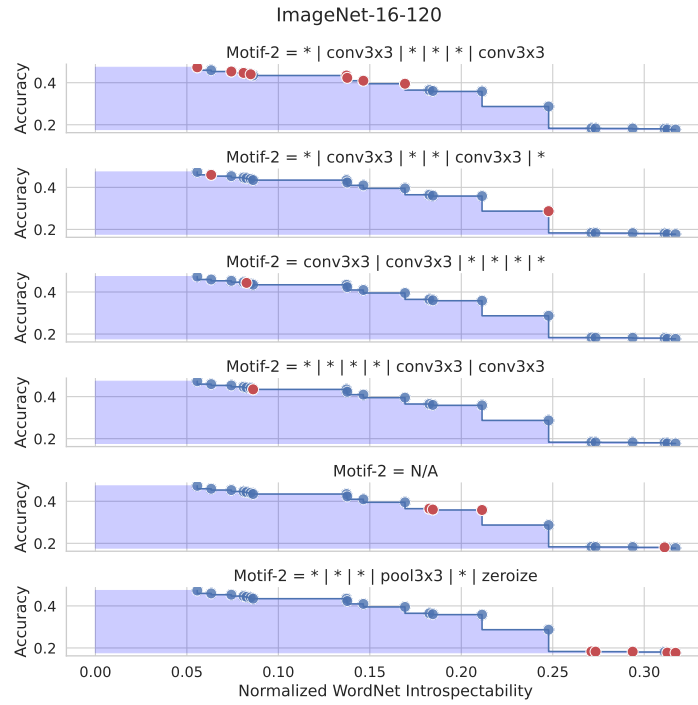


Figure Q13: Discovered motifs of size 2 among the Pareto optimal solutions on the ImageNet-16-120 task. See text for description of the motif discovery process. Each red solution indicates that its architecture has the motif shown in the sub-plot title. The remaining solutions are shown in blue. For the N/A plot, none of the discovered motifs apply to the architecture.

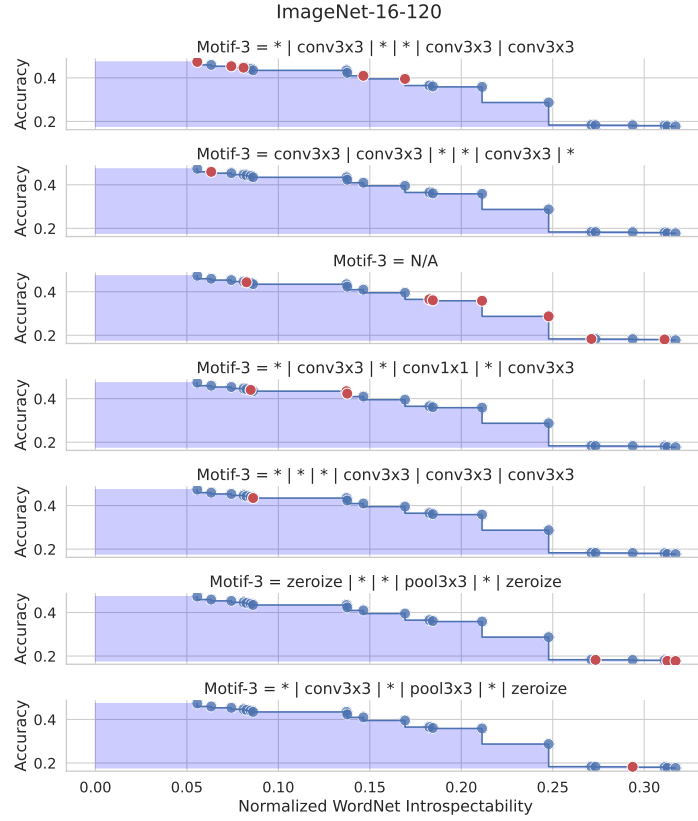


Figure Q14: Discovered motifs of size 3 among the Pareto optimal solutions on the ImageNet-16-120 task. See text for description of the motif discovery process. Each red solution indicates that its architecture has the motif shown in the sub-plot title. The remaining solutions are shown in blue. For the N/A plot, none of the discovered motifs apply to the architecture.

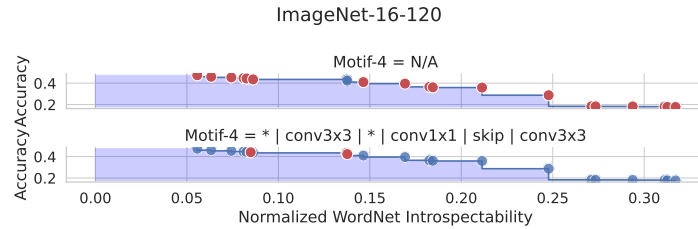


Figure Q15: Discovered motifs of size 4 among the Pareto optimal solutions on the ImageNet-16-120 task. See text for description of the motif discovery process. Each red solution indicates that its architecture has the motif shown in the sub-plot title. The remaining solutions are shown in blue. For the N/A plot, none of the discovered motifs apply to the architecture.