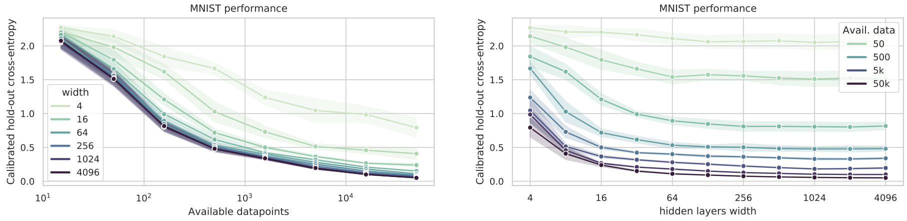
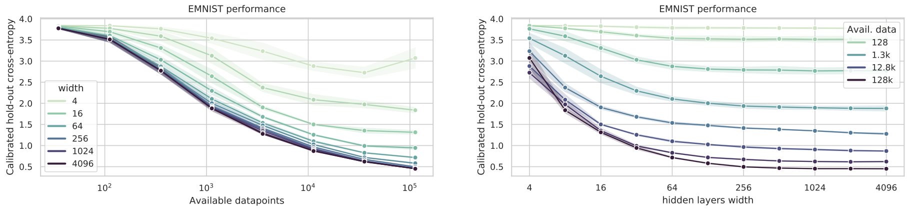
1 Scaling the Model Size

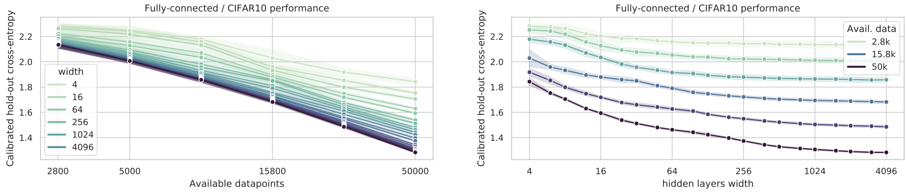
Fully Connected MLP on MNIST



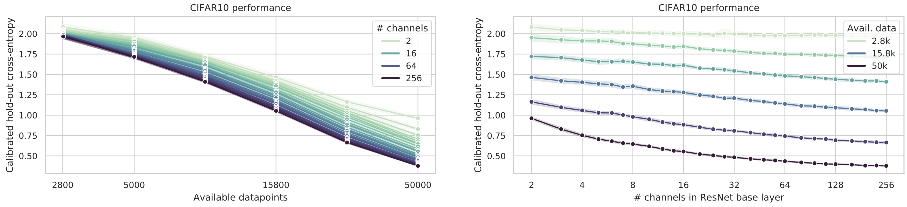
Fully Connected MLP on EMNIST



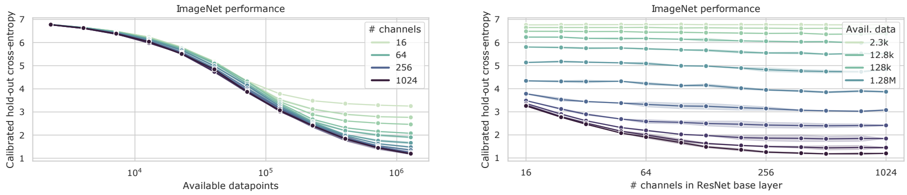
Fully Connected MLP on CIFAR10



ResNet-20 on CIFAR10



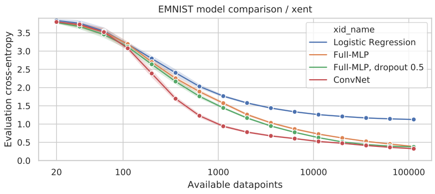
ResNet-101 on ImageNet

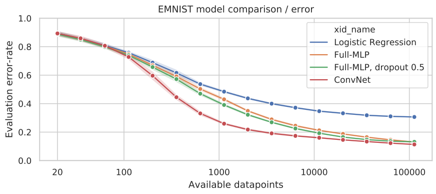


Cross-entropy performance profiles for various models and datasets. We always use 90% of the available data for training, 10% for calibration and evaluate on the (official) validation/development sets provided by the different datasets.

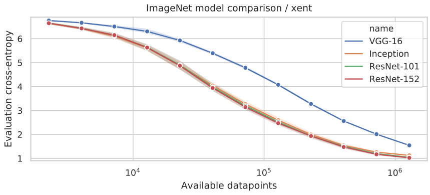
2 Calibrated Cross-Entropy vs. Error-rates

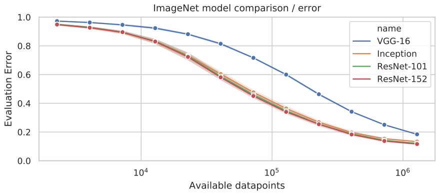
2.1 EMNIST model comparison with either cross-entropy or error-rate





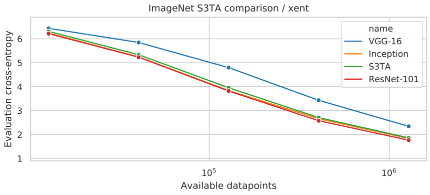
2.2 ImageNet model comparison with either cross-entropy or error-rate

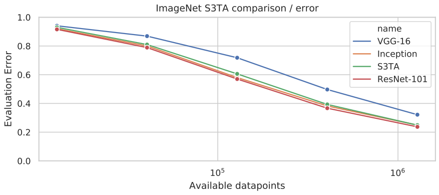




3 S3TA on ImageNet

We here compare the sequential, attention based *S3TA* model against various standard architectures for ImageNet. We use 8 sequential attention steps and a Resnet101-based feature extractor.





4 Compute Infrastructure and Experimental details

We implemented all experiments in Tensorflow and use existing, publicly available code wherever possible. E.g. we use the existing open-source implementation of the NASBench-101 architectures and of Inception; custom implementations of MLPs, ConvNets and ResNets. MLPs where executed on CPUs, simple ConvNets on single GPUs and bigger ResNet, Inception and S3TA models on 4 or 8 TPUs synchronously in parallel. The total batch-size was always fixed to 256.

5 NASBench architectures

These are the NAS-Bench 101 architectures considered in the paper, with their corresponding hashes. We picked architectures equidistantly in terms of performance from the BASBench database, after disregarding the worst 10%.

5.0.1 75ddc0891320c863ec5f148ae675947e

[’input’, ’conv1x1-bn-relu’, ’conv3x3-bn-relu’, ’conv1x1-bn-relu’, ’maxpool3x3’, ’maxpool3x3’, ’output’]

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{1}$$

5.0.2 b5a2bfe35a8f6a21364a992d4dadad31

[’input’, ’maxpool3x3’, ’maxpool3x3’, ’conv3x3-bn-relu’, ’maxpool3x3’, ’conv3x3-bn-relu’, ’output’]

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2}$$

5.0.3 63e9304e6a2aa542eb273dce26477c38

[’input’, ’maxpool3x3’, ’conv3x3-bn-relu’, ’conv3x3-bn-relu’, ’maxpool3x3’, ’maxpool3x3’, ’output’]

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

5.0.4 9f5da3119e80518bd23f9c115c7a18d6

['input', 'conv3x3-bn-relu', 'conv1x1-bn-relu', 'conv1x1-bn-relu', 'conv1x1-bn-relu', 'conv3x3-bn-relu', 'output']

$$M = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

5.0.5 0da48e9f9faecf504244c65b82e0ba71

['input', 'conv3x3-bn-relu', 'conv3x3-bn-relu', 'conv3x3-bn-relu', 'conv3x3-bn-relu', 'output']

$$M = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5}$$