Appendix

A. Experimental Details

A.1. Hyperparameters for training and distillation

All reference models for each search space are **trained from scratch** for 450 epochs on 8 GPUs up to state-of-the-art accuracy using the hyperparameters given in [37] for EfficientNet-B0 [33]. More specifically, we use a total batch size of 1536 with an initial learning rate of 0.096, RMSprop with momentum of 0.9, RandAugment data augmentation [6], exponential weight-averaging, dropout [30] and stochastic depth [14] of 0.2, together with a learning rate decay of 0.97 every 2.4 epochs.

Blockwise knowledge distillation (BKD) is done by training every block for a single epoch. During this epoch, we apply a cosine learning rate schedule [21] considering 20 steps, an initial learning rate of 0.01, a batch size of 256, the Adam [17] optimizer, and random cropping and flipping as data augmentation.

Finetuning is done via end-to-end knowledge distillation (EKD) by using hard ground truth labels and the soft labels of the reference model, see Figure 3(b). We use the same hyperparameters used for training from scratch with the following changes: a decay of 0.9 every 2 epochs, the initial learning rate divided by 5 and no dropout, stochastic depth nor RandAugment. Depending on the reference model and the complexity of the search space, finetuning achieves full from-scratch accuracy in 15-50 epochs, see Figure 10.

A.2. Hardware measurements

All complexity measurements used throughout the text, either hardware-aware or hardware-agnostic, are gathered as follows:

- **Nvidia V100 GPU** latency measurements are done in Pytorch 1.4 with CUDNN 10.0. In a single loop, 20 batches are sent to GPU and executed, while the GPU is synced before and after every iteration. The first 10 batches are treated as a warm-up and ignored; the last 10 are used for measurements. We report the fastest measurement as the latency.

- Measurements on the **Samsung S20 GPU** are always done with a batch-size of 1, in a loop running 30 inferences, after which the system cools down for 1 minute. The average latency is reported.

- The **number of operations** and **number of parameters** are measured using the ptflops framework (https://pypi.org/project/ptflops/).

- Latency measurement on the **simulator** targeting tensor compute units is done with a batch-size of 1. We report the fastest measurement as latency.

All complexity metrics for the reference models shown throughout the text are measured using this same setup.

A.3. Accuracy of baseline models

Accuracy is taken to be the highest reported in [37], the highest reported in the paper, or trained from scratch using the EfficientNet-B0 hyperparameters used in the [37] repository, see Table 3. This is the case for EfficientNet-B0 (our training), MobileNetV2, MnasNet, SPNASNet and FBNet. OFA/Scratch is the "flops@389M_top1@79.1_finetune@75" model from [2] trained from scratch using the hyperparameters used for EfficientNet-B0 in [37]. Note that these baselines are competitive. MobileNetV2 for example, typically has an accuracy of around 72%, while the training in [37] pushes that to 73%. ResNet50 is typically at 76%, but reaches 79% using the training proposed in [37]. ProxylessNas [4] and DNA's [18] accuracy is taken from their respective papers.

A.4. Comments on Accuracy Predictors

A.4.1 Size of the Architecture Library

Tables 4 and 5 show the impact of the size of the Architecture Library used to fit the linear predictor. The tables show how performance varies on a test set of finetuned models for the MobileNetV3 (1.2×) and DONNA search spaces, respectively. Note how the ranking quality, as measured by Kendall-Tau (KT) [16], is always better in this work than in DNA [18]. On top of that, DNA [18] only ranks models within the search space and does not predict accuracy itself. Another metric to estimate the accuracy predictor's quality is the Mean-Squared-Error (MSE) in terms of predicted top-1 accuracy on the ImageNet validation set. Note that for the MobileNetV3 (1.2×) search space, 20 target accuracies are sufficient for a good predictor, as shown in Table 4. We use the same amount of targets for the EfficientNet-B0, Mobilenet V3 (1.0×) and ProxylessNas (1.3×) search spaces. For the DONNA search space, we use 30 target accuracies, see Table 5. Note that the linear accuracy predictor can improve overtime, whenever the Architecture Library is expanded. As predicted Pareto-optimal architectures are finetuned to full accuracy, those results can be added to the library and the predictor can be fitted again using this extra data.

Table 3: Top-1 ImageNet validation accuracy of architectures used throughout the text, with the references indicating the source for the listed accuracy. * are models found by us using OFA [2] for a specific target complexity metric.

Architecture	ImageNet Top-1 [%]	Reference
EfficientNet-B0	77.7	Ours, using [37]
SPNASNet-100	74.084	From [37]
MNasNet-B1-1.0×	74.658	From [37]
MNasNet-A1-1.0×	75.448	From [37]
MNasNet-A1-1.4×	77.2	From [32]
FBNet-C-100	78.124	From [37]
MobileNetV2 (1.0x)	72.970	From [37]
MobileNetV2 (1.4x)	76.516	From [37]
MobileNetV3 (Large)	75.766	From [37]
ProxyLessNas CPU	75.3	From [4]
ProxyLessNas GPU	75.1	From [4]
ProxyLessNas Mobile	74.6	From [4]
ResNet34	75.1	From [37]
ResNet50	79.0	From [37]
OFA/Scratch	77.5	Ours, with [37]
OFA-flops-A*	77.3	Ours, with [37]
OFA-flops-B*	77.5	Ours, with [37]
OFA-flops-C*	78.6	Ours, with [37]
OFA-sim-A*	77.1	Ours, with [37]
OFA-sim-B*	78.1	Ours, with [37]
OFA-sim-C*	78.5	Ours, with [37]
DNA-A	77.1	From [18]
DNA-B	77.5	From [18]
DNA-C	77.8	From [18]
DNA-D	78.4	From [18]

Table 4: Ranking quality for MobileNetV3 (1.2×) using DONNA, as function of the size of the Architecture Library. 'X'T indicates that 'X' targets were used to fit the predictor.

Metric	DNA [18]	10T	20T	30T	40T
Kendall-Tau [16]	0.74	0.79	0.79	0.8	0.82
MSE [top-1%]	NA	0.07	0.09	0.09	0.08

Table 5: Ranking quality for DONNA, as a function of the size of the Architecture Library. 'X'T indicates that 'X' targets were used to fit the predictor.

Metric	DNA [18]	10T	20T	30T	40T
Kendall-Tau [16]	0.77	0.87	0.87	0.9	0.9
MSE [top-1%]	NA	0.28	0.18	0.2	0.19

A.4.2 Choice of Quality Metrics

Apart from using the Noise-To-Signal-Power-Ratio (NSR) (See Section 3), other quality metrics can be extracted and used in an accuracy predictor as well. All quality metrics are extracted on a held-out validation set, sampled from the ImageNet training set, which is different from the default ImageNet validation set in order to prevent overfitting. Three other types of quality metrics are considered on top of the metric described in equation 1: one other block-level metric based on L1-loss and two network-level metrics. The block-level metric measures the normalized L1-loss between ideal feature map Y_n and the block $B_{n,m}$'s output feature map $\bar{Y}_{n,m}$. It can be described as the Noise-to-Signal-Amplitude ratio:

$$\mathcal{L}(W_{n,m}; Y_{n-1}, Y_n) = \frac{1}{C}\sum_{c=0}^{C}\frac{Y_{n,c} - \bar{Y}_{n,m,c_1}}{\sigma_{n,c}} \quad (3)$$

The two network-level metrics are the loss and top-1 accuracy extracted on the separate validation set. The network-level metrics are derived by replacing only block B_n in the reference model with the block-under-test $B_{n,m}$ and then validating the performance of the resulting network. Table 6 compares the performance of the 4 different accuracy predictors built on these different styles of features. Although they are conceptually different, they all lead to a very similar performance on the test set with NSR outperforming the others slightly. Because of this, the NSR metric from equation 1 is used throughout the text.

Table 6: Comparing different quality metrics: NSR (Equation 1), L1, network-level loss and top-1 accuracy for DONNA.

Ranking Metric	DNA [18]	NSR	L1	Loss	Top-1
Kendall-Tau [16]	0.77	0.9	0.89	0.89	0.88
MSE [top-1%]	NA	0.19	0.23	0.41	0.44

A.4.3 Accuracy predictors for different search-spaces

Similar to the procedures discussed in section 3, accuracy models are built for different reference architectures in different search spaces: EfficientNet-B0, MobileNetV3 (1.0×), MobileNetV3 (1.2×) and ProxyLessNas (1.3×). The performance of these models is illustrated in Table 7. Note that we can generate reliable accuracy predictors for all of these search spaces, with very high Kendall-Tau ranking metrics and low MSE on the prediction. The Kendall-Tau value on the MobileNetV3 (1.2×) search space is lower than the others, as the test set is larger for this space than for the others. The model is still reliable, as is made apparent by the very low MSE metric.

A.4.4 Ablation on accuracy predictor

Throughout this work, we use the Ridge regression from scikit-learn [27] as an accuracy predictor. Other choices can also be valid, although the Ridge regression model has proven stable across our experiments. Table 8 compares a non-exhaustive list of accuracy predictors from scikit-learn

Table 7: Comparing the quality of accuracy predictors for different search spaces. Predicted accuracy is the top-1 validation accuracy on ImageNet.

Search-Space	Kendall Tau[16]	MSE [top-1%]
DONNA	0.9	0.19
EfficientNet-B0	0.91	0.15
MobileNetV3 (1.0×)	0.97	0.13
MobileNetV3 (1.2×)	0.82	0.08
ProxyLessNas (1.3×)	0.95	0.04

Table 8: Comparing the quality of different accuracy predictors for the DONNA search space.

Predictor	Kendall Tau[16]	MSE [top-1%]
DONNA	0.9	0.19
Ridge	0.91	0.15
Adaboost	0.89	0.16
Lasso	0.91	0.20
SVM	0.86	1.84
Grad. Boosting Ensemble	0.86	0.53
LARS	0.16	15.9
BaggingRegressor	0.89	0.23

and their performancde on the DONNA architectural test set.

A.5. Finetuning speed

Depending on the search space's complexity, the used reference model in BKD, and the teacher in end-to-end knowledge distillation (EKD), finetuning can be faster or slower in terms of epochs. We always calibrate the finetuning process to be on-par with training from scratch for a fair comparison, but networks can be trained longer for even better results. With the hyperparameters for EKD given in Appendix A.1, Figure 10 shows that finetuning rapidly converges to from-scratch training accuracy for a set of sub-sampled models in different search spaces. Typically, 50 epochs are sufficient for most of the examples. Finetuning speed also depends on the final accuracy of the sub-sampled model. With an accuracy very close to the accuracy of the reference model, larger models typically converge slower using EKD than smaller models with a lower accuracy. For the smaller models, the teacher's guidance dominates more, which leads to faster finetuning.

A.6. Models for various search-spaces

Figure 11 illustrates predicted and measured performance of DONNA models in terms of number of operations, number of parameters, on an Nvidia V100 GPU and on a simulator targeting tensor operations in a mobile SoC. On top of this, predicted Pareto curves for a variety of other search-spaces are shown: MobileNetV3 (1.0×) and MobileNetV3 (1.2×). For these other search-spaces, we perform predictor-based searches in each of the scenarios, illustrating their respective predicted Pareto-optimal trendlines. The quality of these predictors is given in Table 7. For the extra search spaces, some optimal models have been finetuned to verify the predicted curve's validity. For every search space, the same accuracy predictor is used across all scenarios.

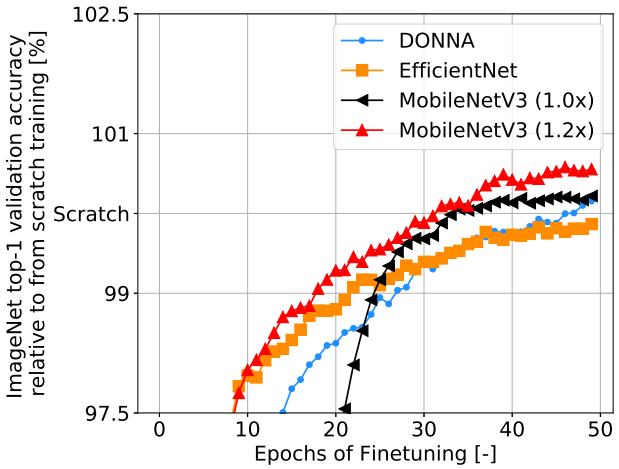


Figure 10: Speed at which BKD-initialized subsampled models can be finetuned for different search spaces. Models in DONNA, EfficientNet and converge to the accuracy of 450-epoch from scratch training in less than 50 epochs using the BKD initialization point, a $9 \times speedup$.

MobileNetV3 (1.0×) and MobileNetV3 (1.2×) are confirmed in terms of number of operations in Figure 11 (mid-left). ProxyLessNass (1.3×) is confirmed on an Nvidia V100 GPU in Figure 11 (mid-right). In the MobileNetV3 (1.0×) space, we find networks that are on-par with the performance of MobileNetV3 [12] in terms of accuracy for the same number of operations, which validates that DONNA can find the same optimized networks as other methods in the same or similar search spaces. Note that the DONNA outperforms all other search spaces on hardware platforms and in terms of number of parameters, which motivates our choice to introduce the new design space. The DONNA space is only outperformed in terms of Pareto-optimality when optimizing for the number of operations, a proxy metric.

B. Model Transfer Study

In this section, we further investigate the transfer properties of DONNA backbones in an object detection task. Our data hints towards two conclusions: (1) ImageNet top-1 validation is a good predictor for COCO mAP if models are sampled from a similar search space and if they are trained using the same hyperparameters and starting from

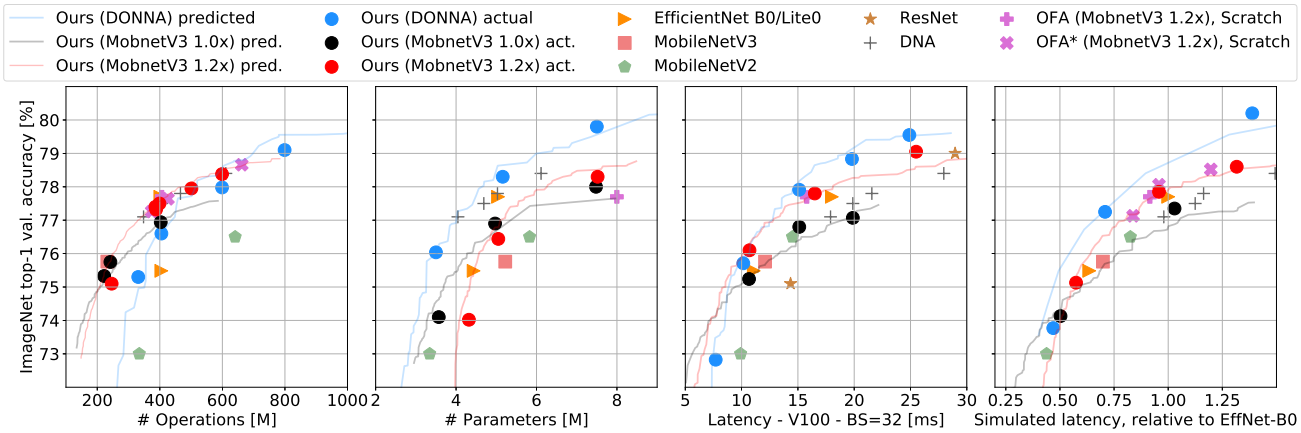


Figure 11: Trendlines and models found by DONNA optimizing for the number of operations (left), the number of parameters (mid left), inference time on an Nvidia V100 GPU (mid right) and a simulator targetting tensor compute units (right). Best viewed in color. This Figure shows the DONNA pipeline finds models of the same quality as OFA [3] when searching in the same search space and optimizing for the same complexity metric (left, right). Second, it shows networks in the DONNA search-space outperform models in the MobileNetV3-1.0× and MobileNetV3-1.2× spaces when targeting the number of parameters, or latency on the discussed hardware platforms. When optimizing for the number of operations, the MobileNetV3-style spaces outperform the DONNA space at accuracies lower than 79%.

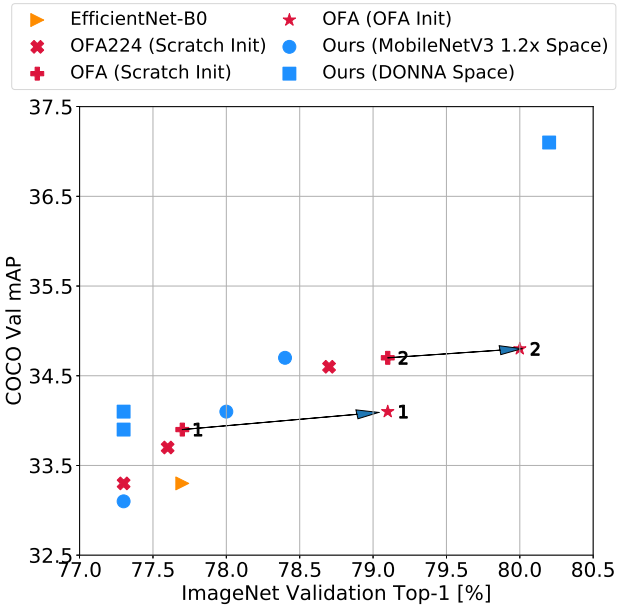


Figure 12: Transfer performance of DONNA backbones to object detection. For DONNA models, COCO validation mAP correlates well with the ImageNet Validation Top-1 accuracy. This is also the case for OFA models, if they are pretrained on ImageNet under the same or similar circumstances. If the OFA models are trained through progressive shrinking, their higher ImageNet accuracy does not transfer to a higher performance on MS-COCO.

the same initialization and (2) higher accuracies on ImageNet achieved through progressive shrinking in OFA do not transfer to significantly higher COCO mAP. The models under study are the same set as in Section 4.2.

These conclusions are apparent from Figure 12. Here, we plot the COCO Val mAPs of the detection architectures against the ImageNet Val top-1 accuracies of their respective backbones. First, we see that OFA models trained from scratch (OFA Scratch and OFA224) and models found in the similar MobileNetV3 (1.2×) search space through DONNA, transfer very similarly to COCO. Models found in the DONNA search space reach higher COCO mAP than expected based on their ImageNet top-1 accuracy. We suspect that such bias occurs because instead of strictly relying on depthwise convolutions, which is the case for MobileNetV3 (1.2×) space, grouped convolutions are used in the DONNA search space. Second, we find that while OFA models with OFA training obtain around 1.0-1.5 percent higher accuracy on ImageNet [8] than the same models trained from scratch, this increased accuracy does not transfer to a meaningful gain in downstream tasks such as object detection. This phenomenon is illustrated in Figure 12, where the same OFA models are trained on MS-COCO, either starting from weights trained on ImageNet from scratch or starting from weights obtained through progressive shrinking on ImageNet. For one of these models, the 1.4% gain in ImageNet validation accuracy only translates into 0.1% higher mAP on COCO. This observation motivates the choice that throughout the text, we compare to OFA-models which are trained from scratch rather than

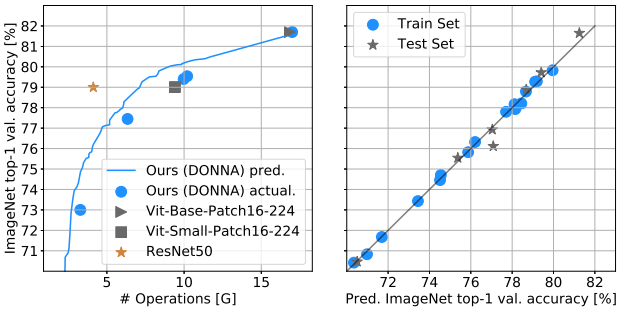


Figure 13: (left) Pareto Optimal models found through a DONNA search in a search space based on vit-base-patch16-224 finetuned on ImageNet from ImageNet21k. vit-base-patch16-224 is pretrained on ImageNet21k and finetuned on ImageNet. vit-small-patch16-224 is taken from [2], and trained using the same pipeline as the DONNA models. (right) Performance of accuracy predictor for the ViT compression case.

through progressive shrinking.

C. DONNA for Vision Transformers

DONNA can be trivially applied to Vision Transformers [9], without any conceptual change to the base algorithm. In this experiment, we use vit-base-patch16-224 from [2] as a teacher model for which we define a related hierarchical search space. Vit-base-patch16-224 is split into 4 DONNA-blocks, each containing 3 ViT blocks (self-attention+MLP) as defined in the original paper [9]. For every block, we vary the following parameters:

- Vit-block *depth* varies $\in$ {1,2,3}
- The *embedding dimension* can be scaled down to 50% of the original embedding dimension $\in${50%,75%,100%}, equivalent to $\in${384,576,768} internally in the DONNA-block.
- The *number of heads* used in attention varies from 4-to-12 $\in$ {4,8,12}.
- The *mlp-ratio* can be varied from 2-4 $\in$ {2,3,4}. Larger mlp-ratios indicate larger MLP's per block.

Potentially, sequence length can be searched over as well, but this is not done in this example. The *Block Library* is built using the BKD process, requiring $4\times3\times3\times3 = 135$ epochs of total training to model a fairly small search space of .5M architectures. The *Architecture Library* exists out of 23 uniformly sampled architectures in this search space, finetuned for 50 epochs on ImageNet [8], using a large CNN model as a teacher until convergence. The latter process is calibrated such that the original teacher model (vit-base-patch16-224), initialized with weights from the *Block Library* achieves the accuracy of the teacher model after these 50 epochs. Note that our reliance on such finetuning and knowledge distillation allows extracting knowledge without access to full datasets, in this case ImageNet21k. Finally, we use the Block- and Architecture libraries to train an accuracy predictor and execute an evolutionary search targeting minimization of the number of operations. Figure 13(left) illustrates the results of this search, showing that our search in this space allows finding a pareto set of models. In terms of number of operations, this ViT-based search space does not outperform ResNet-50. Figure 13(right) illustrates the quality of the accuracy predictor, on a limited set of ViT architectures.

D. Search space extension to Quantized Networks

The DONNA accuracy predictor extends to search spaces different from the one it has been trained for, see Section 4.1.2. This is a major advantage of DONNA , as it enables us to quickly extend pre-existing NAS results without the need to create an extended Architecture Library and without retraining the accuracy predictor. For details on this, see Section 4.1.2 and Fig. 4 for a discussion on this using ShiftNets [39]. This section illustrates that the DONNA accuracy predictor is not only portable across layer types, but also across different compute precisions, i.e. when using quantized INT8 operators.

To demonstrate this, let us consider the MobileNetV3 ($1.2\times$) search space. First, we build and train a DONNA accuracy predictor for full-precision (FP) networks and then test this predictor for networks with weights and activations quantized to 8 bits (INT8). The search space includes k $\in \{3,5,7\}$; expand $\in \{3,4,6\}$; depth $\in \{2,3,4\}$; activation $\in \{ReLU/Swish\}$; attention $\in \{None/SE\}$; and channel-scaling $\in \{0.5\times, 1.0\times\}$. We build a complete Block Library in FP; sampling 43 FP networks as an Architecture Library and finetuning them to collect the training data for the FP accuracy predictor model. Second, we quantize the Block Library using the Data-Free-Quantization (DFQ) [26] post training quantization method using 8 bits weights and activations (INT8). The quantized Block Library now provides the quality metrics for quantized blocks, which can be used as inputs to the FP accuracy predictor to predict INT8 accuracy. Finally, we test the FP accuracy predictor model on a test set of INT8 networks. For this, we sample 20 networks whose INT8-block quality is within the range of the train set of the accuracy predictor. These networks are first finetuned in FP using the procedure outlined in section 3 and then quantized to INT8 using DFQ [26].

Figure 14 illustrates the FP predictor can be used to directly predict the performance of INT8 networks, indicating

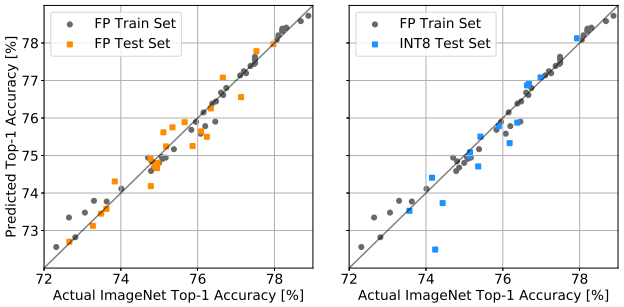


Figure 14: Validation of the accuracy prediction model trained on FP networks and tested on FP networks (right) and INT8 networks (left). Kendal-Tau values are 0.85, and 0.86 respectively for the Test-FP and Test-INT8 sets.

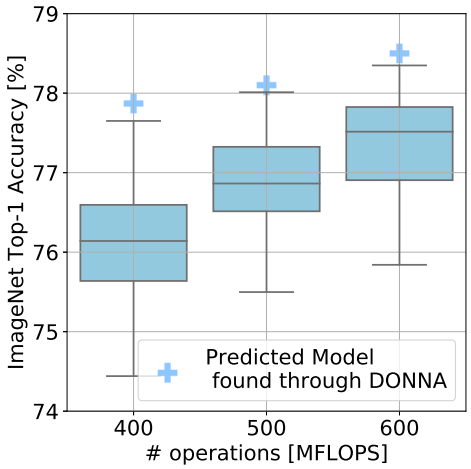


Figure 15: Comparing statistics of random architectures in the MobileNetV3 (1.2×) search-space, as predicted by the DONNA accuracy predictor, to the predicted accuracy of models found through DONNA at the same number of operations.

that DONNA search spaces can indeed be trivially extended to include INT8 precision. Fig. 14(left) shows FP train and test data for the accuracy predictor model. Fig. 14(right) shows FP train and INT8 test data using the same FP accuracy predictor. Formally, we compare the performance of this predictor on the FP and INT8 test set by comparing the achieved prediction MSE and Kendal-Tau (KT) [16]. We can observe that there are no outliers when using the predictor to predict the accuracy of INT8 networks. MSE for the FP test set is 0.13 and 0.34 for the INT8 test set. MSE for INT8 is higher because of the noise introduced by the quantization process. Nonetheless the KT-ranking is 0.85 for FP test set and 0.86 for the INT8 test set demonstrating that the accuracy predictor can be used for INT8-quantized models.

E. Comments on random search

DONNA clearly outperforms random search. In random search, networks are sampled randomly with some latency or complexity constraint and trained from scratch. This can be very costly if the accuracy of these architectures varies widely, as is the case in a large and diverse search space. On top of that, any expensive random search would have to be repeated for every target accuracy or latency on any new hardware platform. This is in stark contrast with DONNA, where the accuracy predictor is reused for any target accuracy, latency and hardware platform.

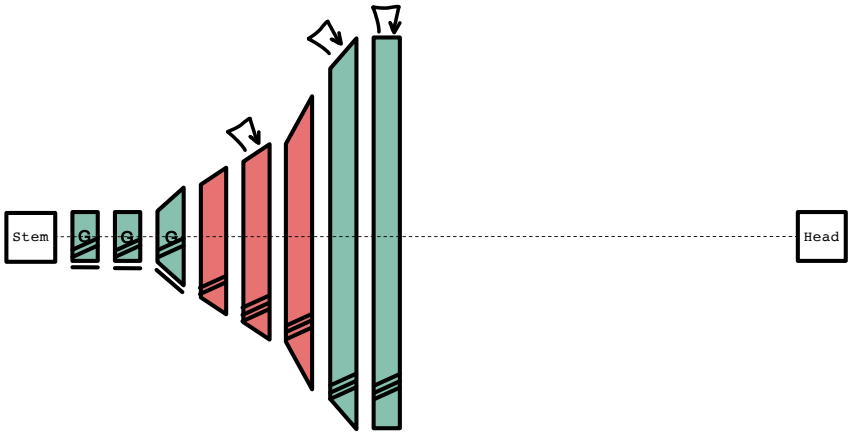
Fig. 15 illustrates box-plots for the predicted accuracy on ImageNet-224 for networks randomly sampled in the MobileNetV3 (1.2×) search space, at 400 +/-5 (190 samples), 500 +/- 5 (77 samples) and 600 +/- 5 (19 samples) million operations (MFLOPS). The box shows the quartiles of the dataset while the whiskers extend to show the rest of the distribution. According to the accuracy predictor, randomly sampled architectures at 400M operations are normally distributed with a mean and standard deviation of 76.2% and 0.7% respectively. Based on this, only around 2% of the randomly sampled architectures will have an accuracy exceeding 77.6%. So, when performing true random search for the 400M operation target, training 100 architectures for 450 epochs (45000 epochs in total) will likely yield 2 networks exceeding 77.6%. In contrast, after building the accuracy predictor for MobileNetV3 (1.2×) in 1500 epochs, DONNA finds an architecture achieving 77.5% at 400M operations in just 50 epochs, see Figure 11(mid-left). This is close to a 900× advantage if the start up cost is ignored, a reasonable assumption at a large amount of targets. In summary, the total cost of random search scales as $N \times 450 \times$ #latency-targets×#platforms, where N is the number of trained samples for every latency-target on every platform. DONNA scales as $50 \times$ #latency-targets×#platforms when many latency-targets and hardware platforms are being considered, meaning the initial costs of building the reusable accuracy predictor can be ignored.

Predictor-based random search could also be used as a replacement for the NSGA-II evolutionary search algorithm [7] in DONNA. However, NSGA-II is known to be more sample efficient than random search in a multi-objective setting [15]. This is also illustrated in Figure 15, where NSGA-II finds networks with a higher predicted accuracy than random search, given the 190 (400M), 77 (500M) and 19 (600M) samples for every target. In this NSGA-II, a total of 2500 samples was generated and measured during the search, covering the full search-space ranging from 150-800M operations.

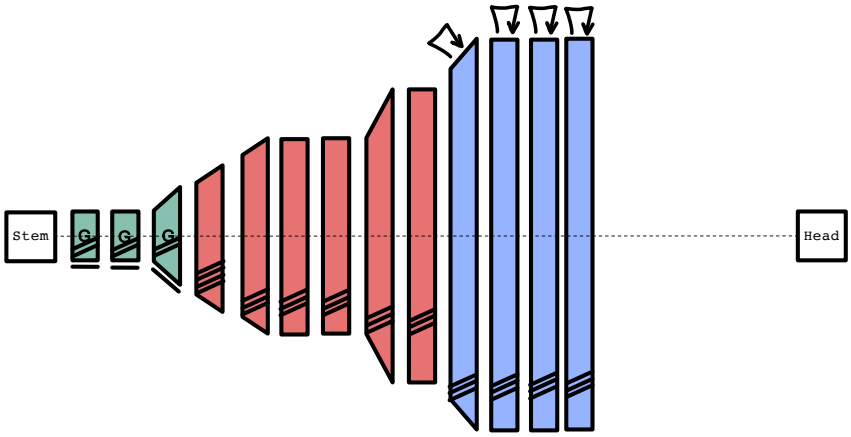
F. Model Visualizations

Figures 16, 17, 18, 19 and 20 visualize some of the diverse network architectures found through DONNA in the DONNA search space. Results are shown for a simulator, the Nvidia V100 GPU, the number of operations, the number of parameters, and the Samsung S20 GPU. Note that all of these networks have different patterns of Squeeze-and-Excite (SE [13]) and activation functions (whenever SE is used, Swish is also used), channel scaling, expansion rates, and kernel factors, as well as varying network depths. In Figure 16, grouped convolutions are also used as parts of optimal networks as a replacement of depthwise separable kernels.

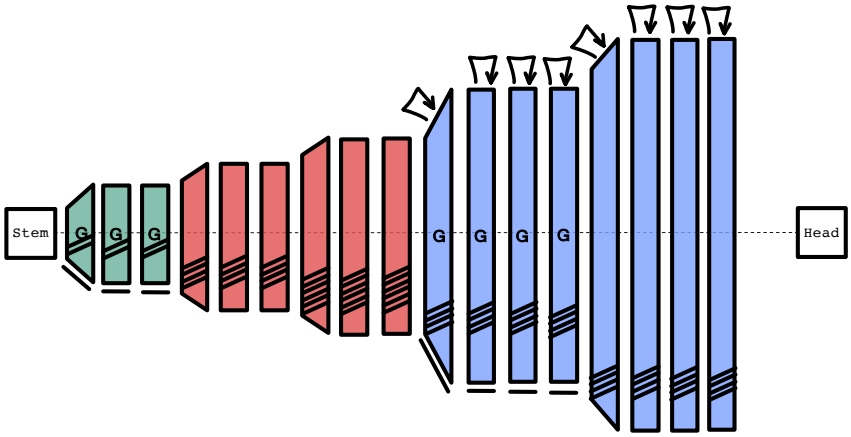
Figure 21 and 22 illustrate optimal EfficientNet-Style networks for the number of operations and the Samsung S20 respectively, as taken from Figure 8. Note how these networks are typically narrower, with higher expansion rates than the DONNA models, which makes them faster or more efficient in some cases. However, EfficientNet-Style models cannot achieve higher accuracy than 77.7% top-1 on ImageNet validation using 224×224 images, while the DONNA search space can achieve an accuracy higher than 80% in that case.



(a) DONNA for a simulator targeting tensor compute in a mobile SoC, 73.7% at 0.45× the latency of EfficientNet-B0.

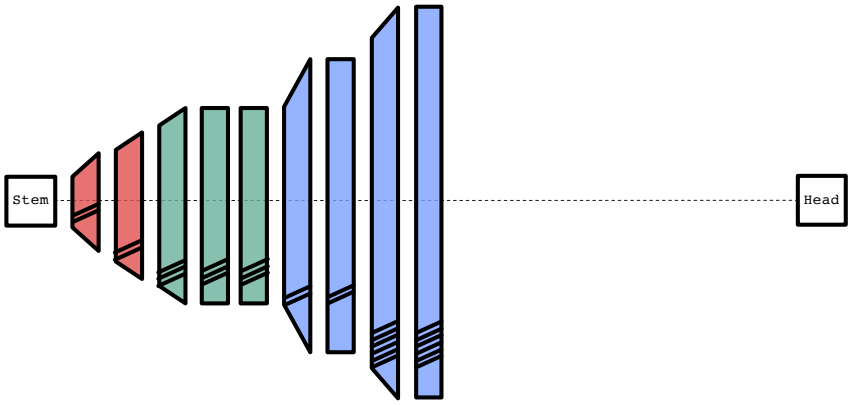


(b) DONNA for a simulator targeting tensor compute in a mobile SoC, 77.25% at 0.60× the latency of EfficientNet-B0.

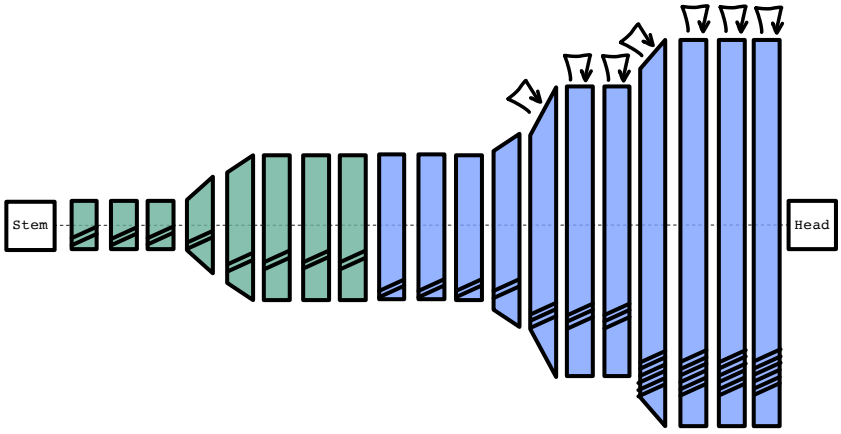


(c) DONNA for a simulator targeting tensor compute in a mobile SoC, 80.2% at 1.25× the latency of EfficientNet-B0.

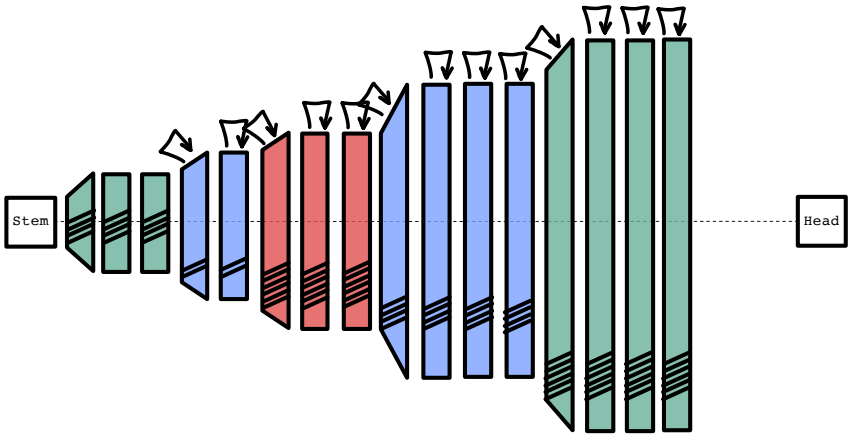
Figure 16: Example models found through DONNA in the DONNA search space, Pareto-optimal on ImageNet for a simulator targeting tensor compute units in a mobile SoC. The Box-color indicates kernel-size: green (3), blue (5) and red (7). Every box is an inverted residual bottleneck with depthwise-separable (plain) or grouped (line under the box) layers. The box height is related to the number of channels. The number of dashed lines per box indicate the expansion rate, the arrows on top indicate whether or not Squeeze-and-Excite and Swish is used.



(a) DONNA for Nvidia V100, batch-size 32, 73% top-1 @7.7ms on ImageNet.

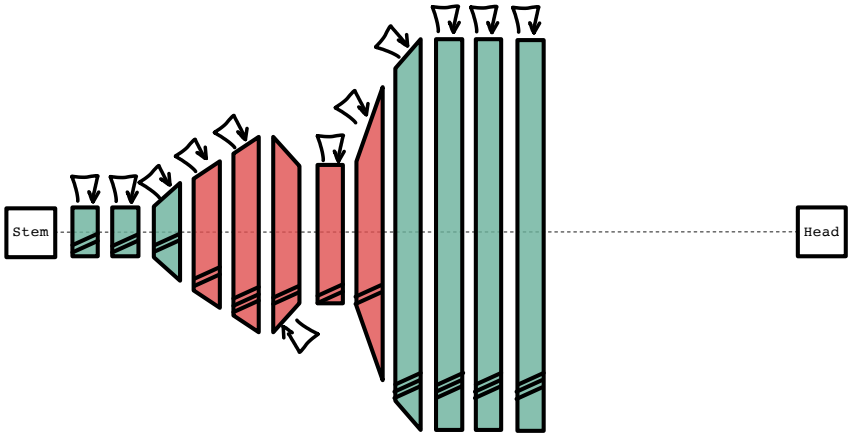


(b) DONNA for Nvidia V100, batch-size 32, 75.7% top-1 @10.15ms on ImageNet.

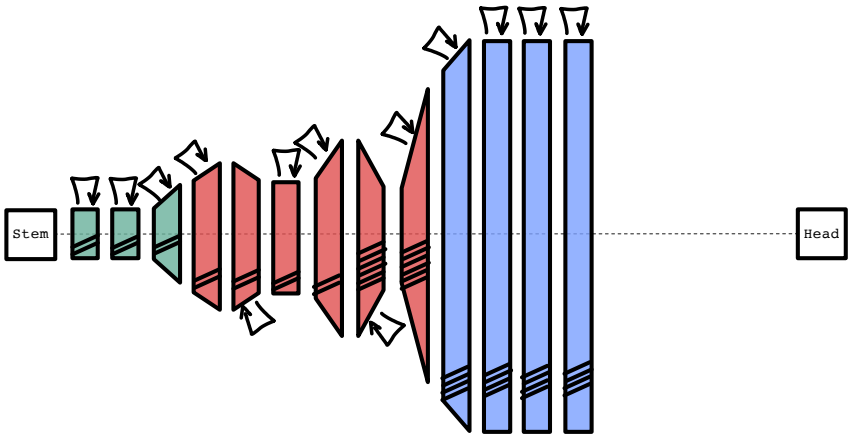


(c) DONNA for Nvidia V100, batch-size 32, 79.5% top-1 @24.9ms on ImageNet.

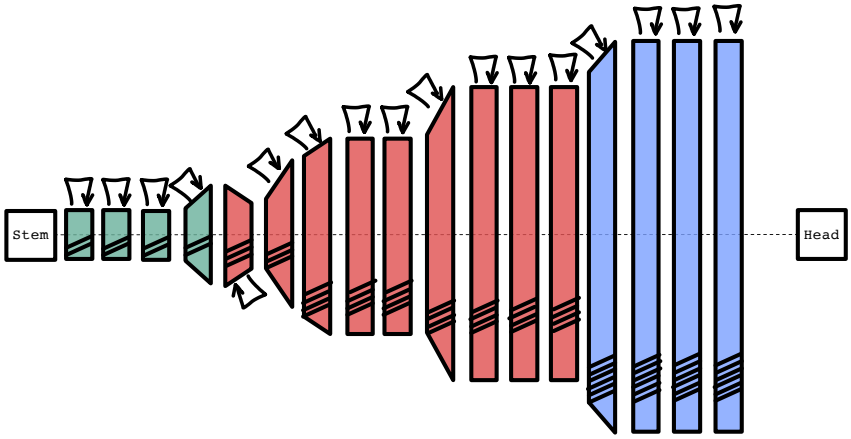
Figure 17: Example models found through DONNA in the DONNA search space, Pareto-optimal on ImageNet for the Nvidia V100 GPU, with a Batch-Size of 32. The Box-color indicates kernel-size: green (3), blue (5) and red (7). Every box is an inverted residual bottleneck with depthwise-separable (plain) or grouped (line under the box) layers. The box height is related to the number of channels. The number of dashed lines per box indicate the expansion rate, the arrows on top indicate whether or not Squeeze-and-Excite and Swish is used.



(a) DONNA for number of operations, 75.3% @ 331 MFLOP

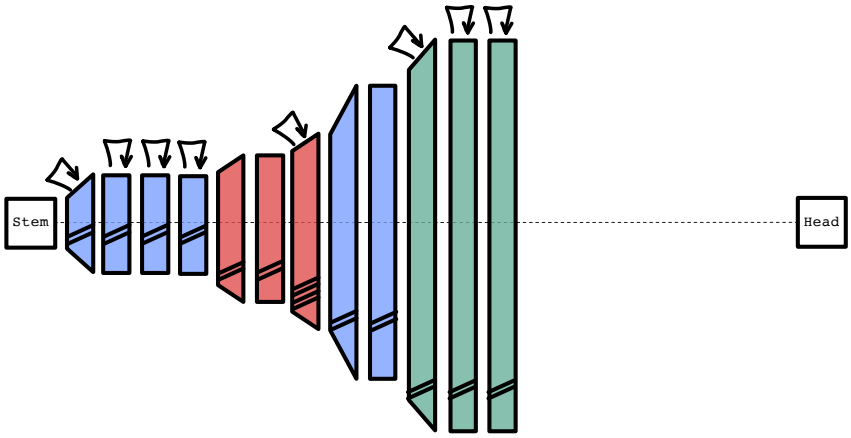


(b) DONNA for number of operations, 76.6% @ 405 MFLOP

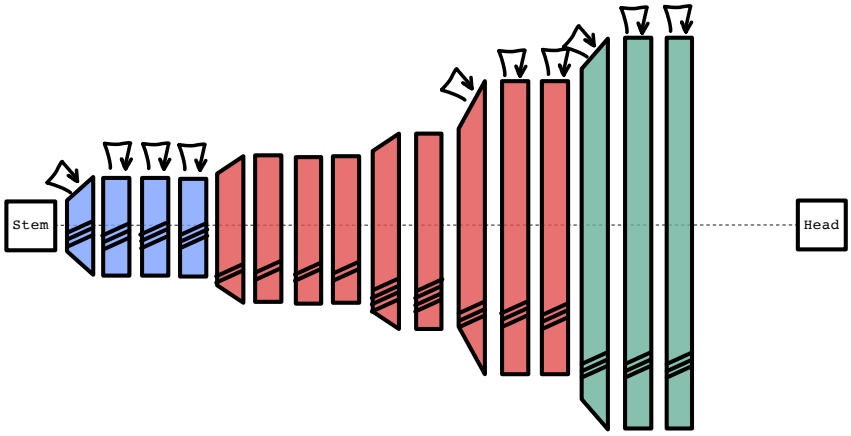


(c) DONNA for number of operations, 79.1% @ 800 MFLOP

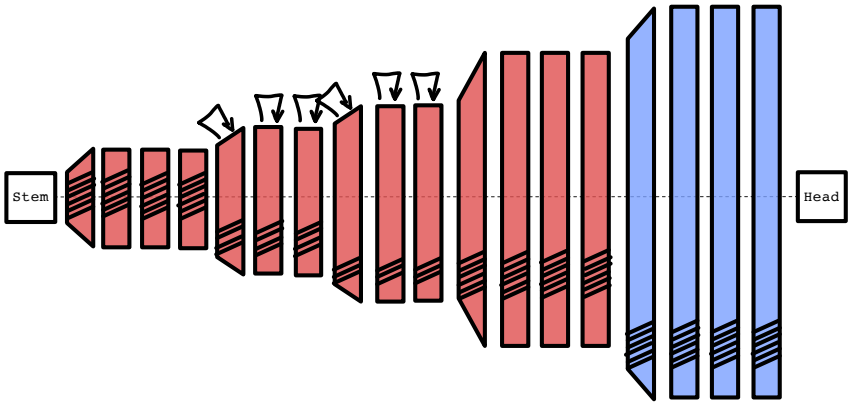
Figure 18: Example models found through DONNA in the DONNA search space, Pareto-optimal on ImageNet optimized for the number of operations. The Box-color indicates kernel-size: green (3), blue (5) and red (7). Every box is an inverted residual bottleneck with depthwise-separable (plain) or grouped (line under the box) layers. The box height is related to the number of channels. The number of dashed lines per box indicate the expansion rate, the arrows on top indicate whether or not Squeeze-and-Excite is used. Note that it is optimal to use SE in every block.



(a) DONNA for number of parameters, 76.0% @ 3.5 million parameters.

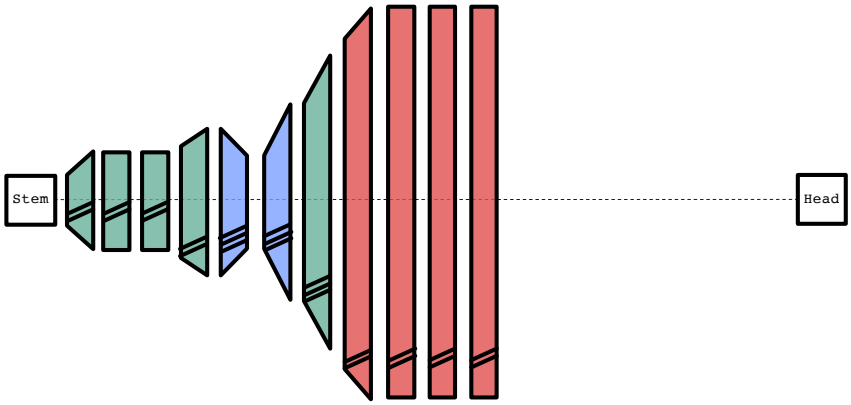


(b) DONNA for number of parameters, 78.3% @ 5.16 million parameters.

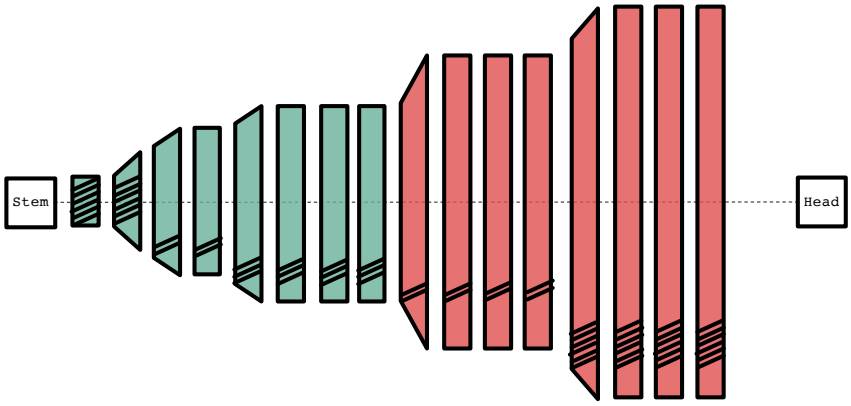


(c) DONNA for number of parameters, 80.0% @ 7.5 million parameters.

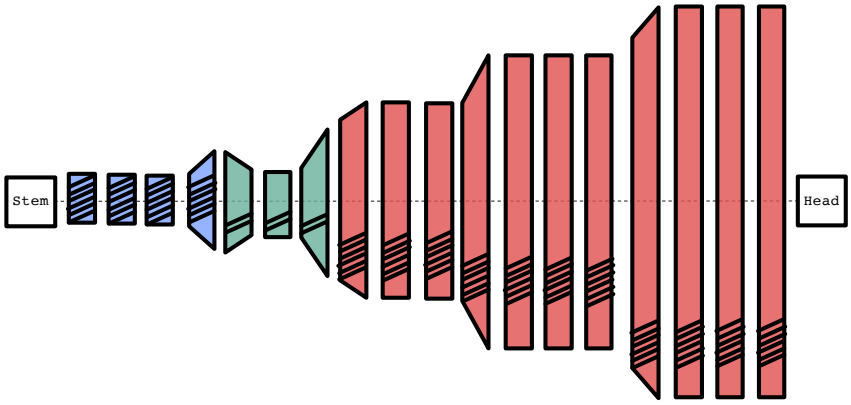
Figure 19: Example models found through DONNA in the DONNA search space, Pareto-optimal on ImageNet optimized for the number of parameters. The Box-color indicates kernel-size: green (3), blue (5) and red (7). Every box is an inverted residual bottleneck with depthwise-separable (plain) or grouped (line under the box) layers. The box height is related to the number of channels. The number of dashed lines per box indicate the expansion rate, the arrows on top indicate whether or not Squeeze-and-Excite and Swish is used.



(a) DONNA for the Samsung S20 GPU, 72.7% @ 6.5 ms.

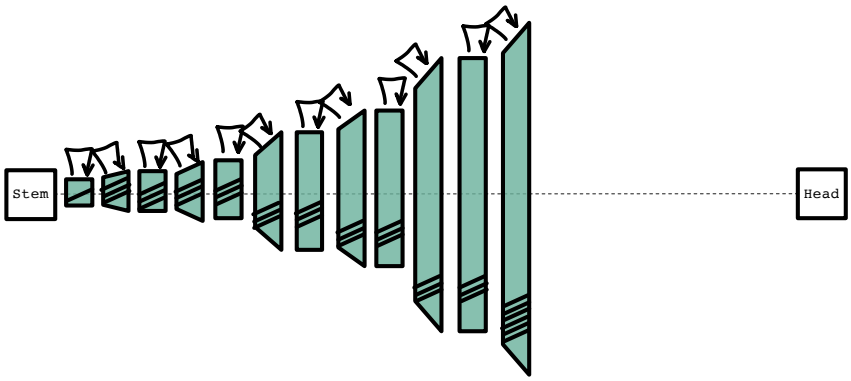


(b) DONNA for the Samsung S20 GPU, 76.96% @ 10.5 ms.

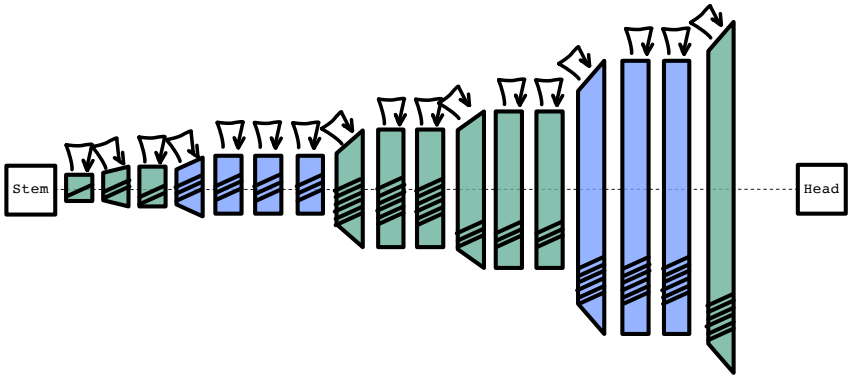


(c) DONNA for the Samsung S20 GPU, 78.9% @ 16.2 ms.

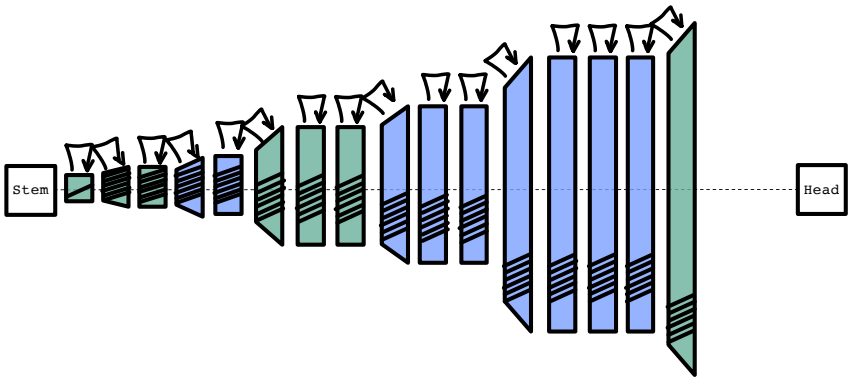
Figure 20: Example models found through DONNA in the DONNA search space, Pareto-optimal on ImageNet optimized for the Samsung S20 GPU. The Box-color indicates kernel-size: green (3), blue (5) and red (7). Every box is an inverted residual bottleneck with depthwise-separable (plain) or grouped (line under the box) layers. The box height is related to the number of channels. The number of dashed lines per box indicate the expansion rate, the arrows on top indicate whether or not Squeeze-and-Excite and Swish is used.



(a) DONNA for number of operations, 73.4% @ 185 MFLOP

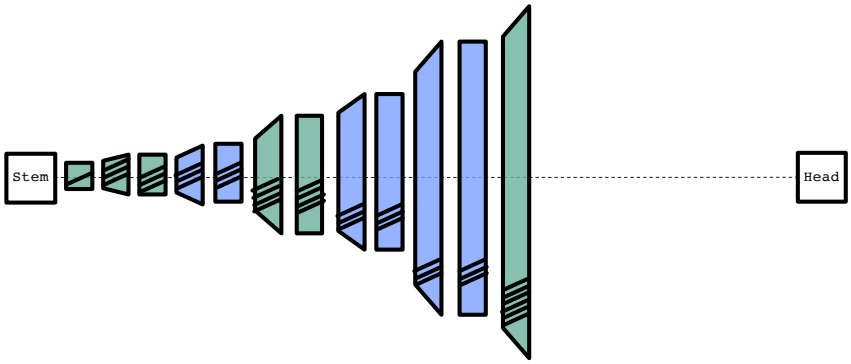


(b) DONNA for number of operations, 76.6% @ 301 MFLOP

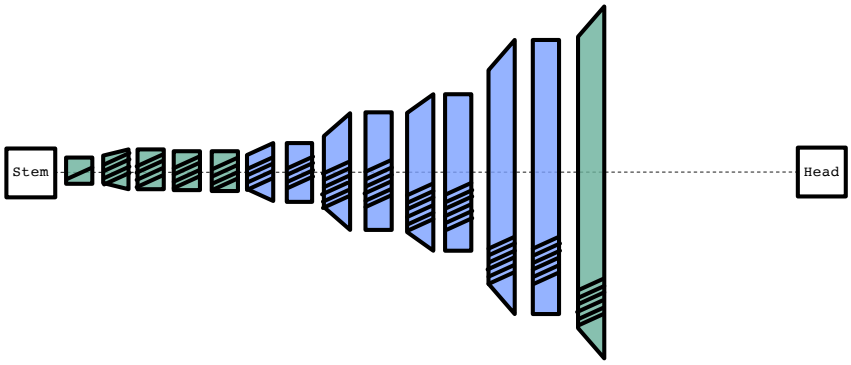


(c) DONNA for number of operations, 77.7% @ 405 MFLOP

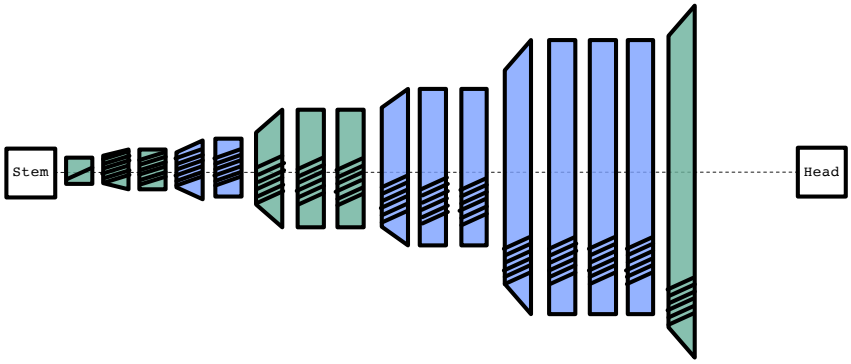
Figure 21: Example models found through DONNA in the EfficientNet-B0 search space, Pareto-optimal on ImageNet optimized for the number of operations. The Box-color indicates kernel-size: green (3), blue (5) and red (7). Every box is an inverted residual bottleneck with depthwise-separable layers. The box height is related to the number of channels. The number of dashed lines per box indicate the expansion rate, the arrows on top indicate whether or not Squeeze-and-Excite is used. Note that it is optimal to use SE in every block.



(a) DONNA for the Samsung S20 GPU, 71.5% @ 5.2 ms.



(b) DONNA for the Samsung S20 GPU, 73.6% @ 5.95 ms.



(c) DONNA for the Samsung S20 GPU, 75.4% @ 8.8 ms.

Figure 22: Example models found through DONNA in the EfficientNet search space, Pareto-optimal on ImageNet optimized for the Samsung S20 GPU. The Box-color indicates kernel-size: green (3), blue (5) and red (7). Every box is an inverted residual bottleneck with depthwise-separable layers. The box height is related to the number of channels. The number of dashed lines per box indicate the expansion rate, the arrows on top indicate whether or not Squeeze-and-Excite and Swish is used.