Supplementary Material for: Assessing the Impact of Diversity on the Resilience of Deep Learning Ensembles: A Comparative Study on Model Architecture, Output, Activation, and Attribution

Rafael Rosales, Pablo Munoz, Michael Paulitsch
Intel Labs, Intel Corporation
{rafael.rosales, pablo.munoz, michael.paulitsch}@intel.com

A. Training parameters for heterogeneous architectures

Table i shows the architecture and optimization hyper-parameters used for training the models used in our experiments.

id	Architecture	Optimizer	Parameters	Scheduler	Epochs	BatchSize
ResNext50_32_2	ResNext50 cardinality=32, blockWidth=2 layers=[3;4;6;3], dropout=0.2	SGD	lr=0.1 decay=0.0001, momentum=0.9	Step: γ=0.1, epochs=30	100	32
ResNext50_32_41	ResNext50 cardinality=32, blockWidth=4 layers=[2;2;3;2], dropout=0.2	SGD	lr=0.1 decay=0.0001, momentum=0.9	Step: γ=0.1, epochs=30	100	32
ResNext50_16_4	ResNext50 cardinality=16, blockWidth=4 layers=[3;4;6;3], dropout=0.2	SGD	lr=0.1 decay=0.0001, momentum=0.9	Step: γ=0.1, epochs=30	100	32
MNASNET_1	MNASNET ratio=1, dropout=0.2	SGD	lr=0.1 decay=0.0001, momentum=0.9	Step: γ=0.1, epochs=30	100	32
MNASNET_1p	MNASNET ratio=1, dropout=0.2	RMSprop	lr=0.256, decay=0.9, momentum=0.9	Step: γ=0.97, epochs=2.4	100	32
Squeezenet_512	Squeezenet version=1.1	SGD	lr=0.01 decay=0.0002, momentum=0.9	Step: γ=0.1, epochs=30	100	512
Squeezenet	Squeezenet version=1.1	SGD	lr=0.01 decay=0.0002, momentum=0.9	Step: γ=0.1, epochs=30	100	128
bootstrapNAS-B_0	ResNet50 depth=[0, 0, 0, 0, 1], width=[0, 0, 0, 2, 2, 2] expansion=[0.2, 0.2, 0.2, 0.25, 0.2, 0.25, 0.25, 0.25,0.2, 0.25, 0.25, 0.25]	SGD	See Listing 1	-	-	-
bootstrapNAS-B_1	ResNet50 d=[0, 0, 0, 0, 1], w=[0, 0, 0, 2, 2, 2] expansion=[0.25, 0.2, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25]	SGD	See Listing 1	-	-	-
deit_tiny_p16_224	DeiT size=tiny(3 heads), patchSize=16, embedding=192	AdamW	lr=0.0005	Cosine	300	256
dino_deit_tiny1	DINO DeiT size=tiny, patchSize=16, embedding=192, seed=0, localCropsScale={0.05,0.2}, globalCropsScale={0.4, 1.}	AdamW	lr=0.0005	Cosine	Backbone=300 Classifier=100	256
dino_deit_tiny2	DINO DeiT size=tiny, patchSize=16, embedding=192, seed=7, localCropsScale={0.05,0.2}, globalCropsScale={0.4, 1.}	AdamW	lr=0.0005	Cosine	Backbone=300 Classifier=100	256
dino_resnet1	DINO ResNet50 backboneSeed=7 localCropsScale={0.05,0.14} globalCropsScale={0.14, 1.}, classifierSeed=5	SGD	decay=0.0001, lr=0.03	-	Backbone=300 Classifier=100	256
dino_resnet2	DINO ResNet50 backboneSeed=7 localCropsScale={0.05,0.14} globalCropsScale={0.14, 1.}, classifierSeed=15	SGD	decay=0.0001, lr=0.03	-	Backbone=300 Classifier=100	256

Table i: Training architectures and parameters used to create ensembles of heterogeneous architectures

Listing 1 shows an example configuration file to create a super-network using a pre-trained model from Torchvision. The configuration parameters are used by BootstrapNAS in the Neural Network Compression Framework (NNCF) and specify the size and elasticity of the super-network. Once the super-network has been trained, the user can extract models of different sizes and performances.

Listing 1: Super-network configuration example for NNCF's BootstrapNAS

```
# Insert here model and dataset fields
# Insert here optimizer fields
"bootstrapNAS":{
    "training": {
        "algorithm":"progressive_shrinking",
        "progressivity_of_elasticity": ["depth", "width"],
        "batchnorm_adaptation": {
            "num_bn_adaptation_samples": 1500},
    "schedule": {
        "list_stage_descriptions": [
        {"train_dims": ["depth"], "epochs": 25,
        "depth_indicator": 1, "init_lr": 2.5e-6,
        "epochs_lr": 25},
        {"train_dims": ["depth"], "epochs": 40, "depth_indicator": 2, "init_lr": 2.5e-6, "
            epochs_lr": 40},
        {"train_dims": ["depth", "width"], "epochs": 50, "depth_indicator": 2, "
            reorg_weights": true, "width_indicator": 2, "bn_adapt": true, "init_lr": 2.5e-6,
             "epochs_lr": 50},
        {"train_dims": ["depth", "width"], "epochs": 50, "depth_indicator": 2, "
            reorg_weights": true, "width_indicator": 3, "bn_adapt": true, "init_lr": 2.5e-6,
             "epochs_lr": 50}
```

```
        ]
    },
    "elasticity": {
        "available_elasticity_dims": ["width", "depth"],
        "width": {
        "max_num_widths": 3,
        "min_width": 32,
        "width_step": 32,
        "width_multipliers": [1, 0.80, 0.60]
        }
    }
},
    "search": {
        "algorithm": "NSGA2",
        "num_evals": 1000,
        "population": 50,
        "ref_acc": 93.65
    }
}
```

The parameters shown in Table ii indicate the configuration of the subnetworks extracted from the super-network in our experiments. We used a previous version of BootstrapNAS in our experiments, which extends Once-for-all (OFA) super-networks from Cai et al. [4] and follows its conventions to describe the search space. In newer versions of BootstrapNAS, expansion ratios are handled by an elastic width handler, and the search space description follows a different convention.

id	Subnetwork Configurations
B_0	depth: [0, 0, 0, 0, 1], expansion: [0.2, 0.2, 0.2, 0.25, 0.2, 0.25, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25], width: [0, 0, 0, 2, 2, 2],
nB_0	depth: [0, 0, 0, 0, 1], expansion: [0.2, 0.2, 0.2, 0.25, 0.2, 0.25, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25], width : [0, 0, 0, 2, 2, 2]
nB_1	depth: [0, 0, 0, 0, 1], expansion: [0.25, 0.2, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25], width : [0, 0, 0, 2, 2, 2]
nB_2	depth: [0, 0, 0, 0, 1], expansion: [0.25, 0.2, 0.25, 0.2, 0.2, 0.25, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25], width : [0, 0, 0, 2, 2, 2]
nB_3	depth: [0, 0, 0, 0, 1], expansion: [0.25, 0.2, 0.25, 0.25, 0.2, 0.2, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25], width : [0, 0, 0, 2, 2, 2]
dB_1a	depth: [0, 0, 1, 0, 0], expansion: [0.25, 0.2, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25], width : [0, 0, 0, 2, 2, 2]
dB_1b	depth: [1, 0, 0, 0, 0], expansion: [0.25, 0.2, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25], width : [0, 0, 0, 2, 2, 2],
wB_1a	depth: [0, 0, 0, 0, 1], expansion: [0.25, 0.2, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25], width : [0, 0, 1, 1, 2, 2],
wB_1b	depth: [0, 0, 0, 0, 1], expansion: [0.25, 0.2, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25], width : [0, 1, 1, 1, 1, 2],
wB_1c	depth: [0, 0, 0, 0, 1], expansion: [0.25, 0.2, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25, 0.2, 0.25, 0.25, 0.25], width : [1, 1, 1, 1, 1, 1],

Table ii: Configuration of subnetworks extracted for the creation of ensembles of heterogeneous architectures

Table iii shows the transformations used during training for all individual models.

Transform	Parameters
RandomResizedCrop	size=(224, 224), scale=(0.08, 1.0), ratio=(0.75, 1.3333), interpolation=bilinear
RandomHorizontalFlip	p=0.5
Normalize	mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]

Table iii: Training data set transforms used for the training of all models

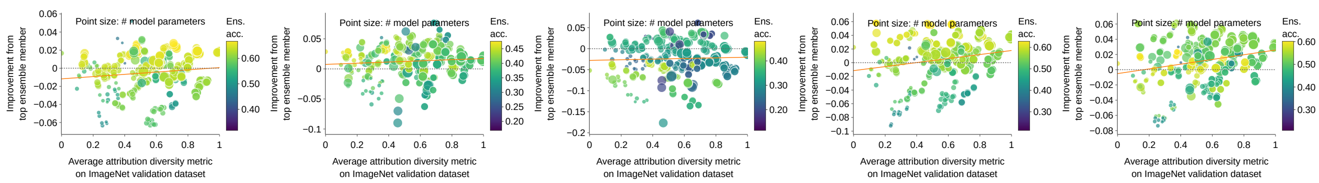
Figure A shows the final top1 accuracy scores of each model defined by Table i.

Model id	Top 1% accuracy
ResNext50_32_2	75.198
ResNext50_32_4l	75.072
ResNext50_16_4	75.432
MNASNET_1	54.408
MNASNET_1p	54.0
Squeezenet_512	41.918
Squeezenet	56.558
bootstrapNAS-B_0	76.342
bootstrapNAS-B_1	76.282
deit_tiny_p16_224	71.654
dino_deit_tiny1	67.932
dino_deit_tiny2	67.52
dino_resnet1	67.236
dino_resnet2	67.216

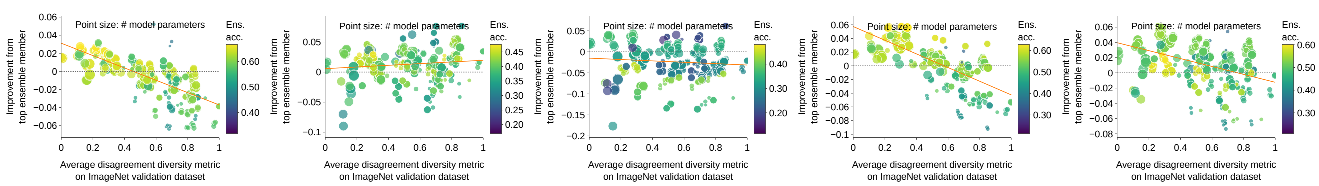
Model id	Top 1% accuracy
B_0	76.342
B_1	76.282
nB_0	76.301
nB_1	76.318
nB_2	76.191
nB_3	76.142
dB_1a	76.138
dB_1b	76.084
wB_1a	76.170
wB_1b	75.804
wB_1c	75.852

Figure A: Individual accuracies of individual trained models on ImageNet validation dataset

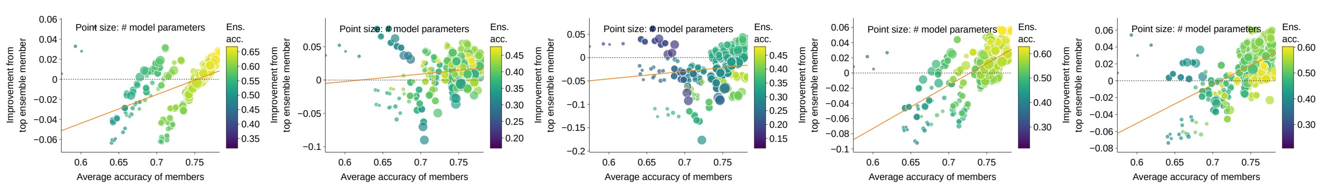
B. Comparison of prediction-based disagreement, input attribution diversity and average accuracy metrics in ensembles of heterogeneous architectures



(a) Attr. ImageNetv2 (b) Attr. Waterdrop (c) Attr. Lines (d) Attr. Plasma (e) Attr. Checkerb.



(f) Disagr. ImageNetv2 (g) Disagr. Waterdrop (h) Disagr. Lines (i) Disagr. Plasma (j) Disagr. Checkerb.



(k) Acc. ImageNetv2 (l) Acc. Waterdrop (m) Acc. Lines (n) Acc. Plasma (o) Acc. Checkerb.

Figure B: Comparison of **improvement correlation of three different metrics on five validation datasets** using averaging as consensus mechanism. Columns: Datasets. Rows: Metrics.