Supplementary Materials for "Learning to Learn Parameterized Classification Networks for Scalable Input Images"

Duo Li[1,2⋆], Anbang Yao[2✉], and Qifeng Chen[1✉]

[1] The Hong Kong University of Science and Technology
[2] Intel Labs China
duo.li@connect.ust.hk   anbang.yao@intel.com   cqf@ust.hk

A Algorithmic Description

We summarize the optimization pipeline of our SAN framework in Algorithm 1. The two methods for inference mentioned in the Section 3.3 of the main paper are also presented in Algorithm 2 and 3 respectively.

Algorithm 1: Optimization of Scale Adaptive Network

```
Input: Training dataset $\mathcal{D}$, maximal iteration $M$, training resolution range
       $\mathbb{S} = \{S_1, S_2, \cdots, S_k\}$, main network architecture $\mathcal{N}$.
Output: Parameters of the meta network $\mathcal{M}$, $\text{BN}^{(S_i)}$ for each input
        resolution $S_i, i = 1, 2, \cdots, k$ and the FC layer.
Initialize the network parameters $\mathcal{M}$, $\text{BN}^{(S_i)}, i = 1, 2, \cdots, k$ and FC.
for $t \leftarrow 1$ to $M$ do
    Randomly draw a batch of samples $\mathcal{B}$ from $\mathcal{D}$.
    foreach $\mathbf{X}$ in $\mathcal{B}$ do
        for $i \leftarrow 1$ to $k$ do
            Randomly crop and resize the mini-batch data to the training
              resolution $S_i$, obtaining $\mathcal{B}^{(S_i)}$ with the prereserved labels.
            Derive scale encoding $\varepsilon^{(S_i)}$.
            Generate convolutional layers $\mathcal{M}(\varepsilon^{(S_i)})$ via Eqn. 2[a].
            Parameterize $\mathcal{N}^{(S_i)}$ with $\mathcal{M}(\varepsilon^{(S_i)})$, $\text{BN}^{(S_i)}$ and FC.
            Foward and compute cross-entropy loss $\mathcal{L}_{\text{CE}}^{(S_i)}$ via Eqn. 1.
            Compute scale distillation loss $\mathcal{L}_{\text{SD}}^{(S_i)}$ via Eqn. 3.
        end
        Compute the total loss via an un-weighted summation in Eqn. 4.
        Update network parameters with the SGD optimizer.
    end
end
return $\mathcal{M}$, $\text{BN}^{(S_i)}, i = 1, 2, \cdots, k$ and FC.
```

[a] All the referred equations in this algorithm are presented in the main paper.

⋆ indicates intern at Intel Labs China. ✉ indicates corresponding authors.

Algorithm 2: Ideal Inference of Scale Adaptive Network

```
Input: Test image $\mathcal{I}$ with the resolution $T$, training dataset $\mathcal{D}$, training
       resolution range $\mathbb{S} = \{S_1, S_2, \cdots, S_k\}$, main network architecture $\mathcal{N}$,
       meta network $\mathcal{M}$, $\mathrm{BN}^{(S_i)}$ for each training resolution
       $S_i, i = 1, 2, \cdots, k$ and FC.
Output: Category prediction of the test image.
Search the nearest resolution $S(T) \in \mathbb{S}$ for $T$.
Derive scale encoding $\varepsilon^{(T)}$.
Generate convolutional layers $\mathcal{M}(\varepsilon^{(T)})$ for main network $\mathcal{N}^{(T)}$.
Calibrate statistics of $\mathrm{BN}^{(S(T))}$ using data set $\mathcal{D}^{(T)}$ with test resolution $T$.
Parameterize $\mathcal{N}^{(T)}$ with $\mathcal{M}(\varepsilon^{(T)})$, calibrated $\mathrm{BN}^{(S(T))}$ and FC.
Feed the image $\mathcal{I}$ to $\mathcal{N}^{(T)}$ to get its category prediction $\mathcal{N}^{(T)}(\mathcal{I})$.
return $\mathcal{N}^{(T)}(\mathcal{I})$.
```

Algorithm 3: Proxy Inference of Scale Adaptive Network

```
Input: Test image $\mathcal{I}$ with the resolution $T$, training resolution range
       $\mathbb{S} = \{S_1, S_2, \cdots, S_k\}$, main network architecture $\mathcal{N}$, meta network
       $\mathcal{M}$, $\mathrm{BN}^{(S_i)}$ for each training resolution $S_i, i = 1, 2, \cdots, k$ and FC.
Output: Category prediction of the test image.
1 Search the nearest resolution $S(T) \in \mathbb{S}$ for $T$.
2 Derive scale encoding $\varepsilon^{(S(T))}$[a].
3 Generate convolutional layers $\mathcal{M}(\varepsilon^{(S(T))})$ for main network $\mathcal{N}^{(S(T))}$[a].
4 Parameterize $\mathcal{N}^{(T)} = \mathcal{N}^{(S(T))}$ with $\mathcal{M}(\varepsilon^{(S(T))})$, $\mathrm{BN}^{(S(T))}$ and FC.
5 Feed the image $\mathcal{I}$ to $\mathcal{N}^{(T)}$ to get its category prediction $\mathcal{N}^{(T)}(\mathcal{I})$.
6 return $\mathcal{N}^{(T)}(\mathcal{I})$.
```

[a] The steps 2 and 3 can be omitted if the pre-trained model $\mathcal{N}^{(S(T))}$ is on hand.

Algorithm 4: Data-Free Ideal Inference of Scale Adaptive Network

```
Input: Test image $\mathcal{I}$ with the resolution $T$, training resolution range
       $\mathbb{S} = \{S_1, S_2, \cdots, S_k\}$, main network architecture $\mathcal{N}$, meta network
       $\mathcal{M}$, $\mathrm{BN}^{(S_i)}$ for each training resolution $S_i, i = 1, 2, \cdots, k$ and FC.
Output: Category prediction of the test image.
Search the two nearest resolutions $S_{floor}(T), S_{ceil}(T) \in \mathbb{S}$ for $T$, s.t.
  $S_{floor}(T) < T < S_{ceil}(T)$.
Derive scale encoding $\varepsilon^{(T)}$.
Generate convolutional layers $\mathcal{M}(\varepsilon^{(T)})$ for main network $\mathcal{N}^{(T)}$.
Calculate $\mathrm{BN}^{(T)}$ by linearly interpolating between $\mathrm{BN}^{(S_{floor}(T))}$ and
  $\mathrm{BN}^{(S_{ceil}(T))}$.
Parameterize $\mathcal{N}^{(T)}$ with $\mathcal{M}(\varepsilon^{(T)})$, $\mathrm{BN}^{(T)}$ and FC.
Feed the image $\mathcal{I}$ to $\mathcal{N}^{(T)}$ to get its category prediction $\mathcal{N}^{(T)}(\mathcal{I})$.
return $\mathcal{N}^{(T)}(\mathcal{I})$.
```

Table 1. Ratios of weights to biases in the meta networks for each individual layer in the ResNet-18 (*left*), ResNet-50 (*middle*) and MobileNetV2 (*right*) models.

Block	Layer	Ratio
conv2_0	layer1	0.4750
	layer2	0.4894
conv2_1	layer1	0.4821
	layer2	0.4963
conv3_0	layer1	0.4853
	layer2	0.5026
	skip connection	0.4993
conv3_1	layer1	0.5002
	layer2	0.5081
conv4_0	layer1	0.4937
	layer2	0.5137
	skip connection	0.4766
conv4_1	layer1	0.5114
	layer2	0.5179
conv5_0	layer1	0.5012
	layer2	0.5222
	skip connection	0.4953
conv5_1	layer1	0.5263
	layer2	0.5316

Block	Layer	Ratio
conv2_0	layer1	0.5117
	layer2	0.4500
	layer3	0.4386
	skip connection	0.4652
conv2_1	layer1	0.4330
	layer2	0.4400
	layer3	0.4521
conv2_2	layer1	0.4485
	layer2	0.4418
	layer3	0.4501
conv3_0	layer1	0.4414
	layer2	0.4340
	layer3	0.4367
	skip connection	0.4414
conv3_1	layer1	0.4525
	layer2	0.4600
	layer3	0.4468
conv3_2	layer1	0.4557
	layer2	0.4574
	layer3	0.4549
conv3_3	layer1	0.4497
	layer2	0.4602
	layer3	0.4557
conv4_0	layer1	0.4549
	layer2	0.4401
	layer3	0.4490
	skip connection	0.4424
conv4_1	layer1	0.4731
	layer2	0.4797
	layer3	0.4754
conv4_2	layer1	0.4676
	layer2	0.4797
	layer3	0.4746
conv4_3	layer1	0.4705
	layer2	0.4775
	layer3	0.4753
conv4_4	layer1	0.4699
	layer2	0.4756
	layer3	0.4766
conv4_5	layer1	0.4705
	layer2	0.4706
	layer3	0.4731
conv5_0	layer1	0.4691
	layer2	0.4576
	layer3	0.4635
	skip connection	0.4609
conv5_1	layer1	0.4844
	layer2	0.4919
	layer3	0.4940
conv5_2	layer1	0.4930
	layer2	0.5079
	layer3	0.5108

Block	Layer	Ratio
block0	layer1	0.4561
block1	layer1	0.5730
	layer2	0.4905
block2	layer1	0.4874
	layer2	0.4946
	layer3	0.4703
block3	layer1	0.5081
	layer2	0.5487
	layer3	0.5362
block4	layer1	0.4949
	layer2	0.4617
	layer3	0.4730
block5	layer1	0.4931
	layer2	0.5195
	layer3	0.5067
block6	layer1	0.5051
	layer2	0.5229
	layer3	0.5054
block7	layer1	0.4720
	layer2	0.5268
	layer3	0.4647
block8	layer1	0.5064
	layer2	0.6210
	layer3	0.5130
block9	layer1	0.5080
	layer2	0.6138
	layer3	0.5095
block10	layer1	0.5103
	layer2	0.6087
	layer3	0.5059
block11	layer1	0.4898
	layer2	0.8277
	layer3	0.4806
block12	layer1	0.5007
	layer2	0.6022
	layer3	0.5089
block13	layer1	0.4992
	layer2	0.6108
	layer3	0.5006
block14	layer1	0.4941
	layer2	0.6202
	layer3	0.4818
block15	layer1	0.5345
	layer2	0.8139
	layer3	0.5334
block16	layer1	0.5284
	layer2	0.7795
	layer3	0.5212
block17	layer1	0.5011
	layer2	1.6666
	layer3	0.4897

B Ratios of Weights to Biases in the Meta Networks

As a sanity check, we investigate the trained meta network for the l^{th} convolutional layer and evaluate the ratio of its weight $\mathbf{W}_l$ to bias $\mathbf{b}_l$. Specifically, we take the average of their absolute values and compute the corresponding ratio. We report these derived ratios for each layer of three backbone network architectures in Table 1. We identify that these ratios are mostly around 0.5, showing a similar level of magnitude between weights and biases, thus the equivalent importance in contributing to the learning dynamics of the meta networks.

C Complementary Information across Input Resolutions

Through the statistical experiments in this section, we demonstrate why diverse input resolutions could contain complementary information and whether our proposed SAN framework takes advantage of such information or not. Regarding the first question, we maintain a suspicion of whether the correctly classified images at a small resolution will fall into the subset of those correctly classified at another large resolution. We conduct relevant experiments using ResNet-18 on ImageNet with the selected training resolution range $\mathbb{S} = \{S_1, S_2, S_3, S_4, S_5\} = \{224, 192, 160, 128, 96\}$ and summarize the statistics in the left panel of Table 2. The numeric value located in the i^{th} row and j^{th} column refers to the proportion of validation images missed in the top-1 prediction of $\mathcal{N}^{(S_i)}$ but hit in that of $\mathcal{N}^{(S_j)}$. It is consistent with our intuition that the hit rate reduces with the decreasing resolution inside each row. However, it is intriguing that there remain around 5% hit rates in the upper triangular part of this matrix chart, which implies a negative response to our suspicion above. The inspiration derived from the results is that images resized to one resolution could contain information complementary to their versions of other resolutions regarding the recognition process. Hence it would be promising to enhance the model performance on one test resolution through learning informative features from other resolutions during the mixed-scale training process. Hopefully, the model optimized under this collaborative regime could also improve its tolerance to other auxiliary resolutions in turn during evaluation. In order to justify these conjectures, we also show the improved results of ResNet18-based SAN in the right panel of Table 2. Since the mixed-scale training enforces one image instance to be correctly recognized at a spectrum of resolutions simultaneously, the model predictions achieve better consistency (reduced hit-miss ratios) across different resolutions under our SAN framework, validated by the comparison between the two sub-tables in Table 2. Therefore, the scale-adaptive behavior and improved performance of SAN could be attributed to better utilization of the complementary information across different input resolutions.

D Switching Input Resolution vs. Width Multipliers

Tuning the width multipliers and input resolutions are two ubiquitous ways to adapt CNNs to the on-device computational budget [2][3][1]. We compare

Table 2. Percentage of ImageNet validation images which are mistakenly classified at one input resolution but accurately recognized at another resolution according to the top-1 prediction. The ResNet-18-based SAN (*right*) achieves consistently reduced hit-miss ratios compared to the ResNet-18 baseline models (*left*).

Miss \ Hit	224 × 224	192 × 192	160 × 160	128 × 128	96 × 96
224 × 224	0	5.60%	5.50%	5.17%	4.73%
192 × 192	6.82%	0	5.88%	5.47%	5.06%
160 × 160	8.00%	7.15%	0	5.92%	5.27%
128 × 128	9.79%	8.86%	8.04%	0	5.81%
96 × 96	13.15%	12.26%	11.19%	9.61%	0

Miss \ Hit	224 × 224	192 × 192	160 × 160	128 × 128	96 × 96
224 × 224	0	2.88%	3.33%	3.67%	3.76%
192 × 192	3.81%	0	3.18%	3.57%	3.79%
160 × 160	5.88%	4.79%	0	3.45%	3.75%
128 × 128	9.02%	7.99%	6.25%	0	3.98%
96 × 96	12.38%	11.74%	10.36%	8.54%	0

Table 3. Top-1 accuracy (%) of MobileNetV2 baseline models with different width multipliers, S-MobileNetV2 and US-MobileNetV2 (*left*), as well as baseline models operating on different input resolutions and MobileNetV2-based-SAN (*right*) on the ImageNet validation set. The results of S-Net and US-Net are extracted from the official publications.

Width Multiplier	MFLOPs	MobileNetV2	S-Net [6]	US-Net [5]
1.0×	301	72.2	70.5	71.5
0.8×	222	69.8	-	70.0
0.65×	161	68.0	-	68.3
0.5×	97	64.6	64.4	65.0
0.35×	59	60.1	59.7	62.2

Input Resolution	MFLOPs	MobileNetV2	SAN (**ours**)
224 × 224	300	72.2	**72.8**
192 × 192	221	71.1	**72.2**
160 × 160	154	69.5	**71.2**
128 × 128	99	66.7	**69.1**
96 × 96	56	62.7	**65.1**

the effect on performance via scaling the width multiplier or input resolution to achieve a similar level of computational complexities, taking MobileNetV2 as the backbone network. From the comparison between baseline MobileNetV2 models in the two sub-tables of Table 3, we reach the preliminary conclusion that adjusting input resolutions is more effective at achieving higher accuracy with the similar computational cost. Recently, a series of methods, S-MobileNetV2 [6] and US-MobileNetV2 [5], were proposed to flexibly switch the network width to achieve an accuracy-efficiency trade-off at runtime, but they merely obtain marginal gains or even slight drop regarding the recognition accuracy as shown in the left panel of Table 3. In contrast, our SAN realizes dynamic and adaptive inference from the perspective of input resolutions, achieving further performance enhancement even compared to those strong baseline models operating on their training resolutions as shown in the right panel of Table 3.

E Generalization to Other Large Resolutions

Similar to an ablation study in the Section 4.2 of the main paper, we also shift the range of training resolutions not so aggressively, with a selection of $\mathbb{S} = \{320, 256, 224, 192, 160\}$. The accuracy of baseline models trained individually are summarized in the top panel of Table 4 while the accuracy of SAN are shown in the bottom panel of Table 4. The tendency of accuracy gains with respect to test resolutions is similar to the results based on ranges of both smaller and larger training resolutions in the main paper.

Table 4. Comparison of ResNet-18 baseline models (*top*) and ResNet-18-based SAN (*bottom*) with a range of medium training resolutions. The shaded numerical values have the same meaning as those in Table 1 of the main paper.

Train \ Test	384	320	256	224	208	192	176	160	144	128	112	96	64
320	73.42	72.56	70.03	67.44	64.92	63.23	59.47	56.07	51.17	45.89	38.64	30.27	11.74
256	72.75	72.68	71.71	70.24	68.00	67.27	64.05	62.12	57.20	53.27	45.45	38.00	17.00
224	71.87	72.36	71.90	70.97	69.14	68.72	66.20	64.68	60.36	56.85	50.17	42.55	20.40
192	69.95	71.07	71.71	71.25	69.64	69.75	67.69	67.04	63.23	60.47	54.21	47.73	25.10
160	66.95	69.05	70.72	70.70	69.50	70.14	68.47	68.48	65.61	63.80	58.54	53.78	31.03

Train \ Test	384	320	256	224	208	192	176	160	144	128	112	96	64
320	74.23	74.48	73.30	71.92	70.57	69.66	67.31	65.66	61.68	58.36	52.10	45.22	23.85
256	73.25	74.19	73.96	73.10	72.01	71.55	69.53	68.26	65.22	62.18	56.65	50.34	28.15
224	72.03	73.44	73.88	73.31	72.33	71.12	70.35	69.31	66.64	63.99	58.81	52.77	30.52
192	70.35	72.11	73.34	73.23	72.33	72.32	70.95	70.10	67.70	65.49	60.77	55.25	33.13
160	67.57	70.12	72.14	72.43	71.67	71.99	70.96	70.52	68.25	66.70	62.36	57.67	36.18

F More Dense Sampled Resolutions for Inference

To further prove the applicability of our method to universally scalable input images, we sample the test resolutions in a more dense style. Suppose the training resolution range is $\mathbb{S} = \{S_1, S_2, S_3, S_4, S_5\} = \{224, 192, 160, 128, 96\}$ with the interval of 32, the test resolution range is set as $\mathbb{T} = \{S \pm \frac{1}{2} \times 32 | S \in \mathbb{S}\} = \{208, 176, 144, 112\}$ in the main paper. In addition, we conduct experiments on the test resolution range in a more dense arrangement as $\mathbb{T}' = \{S \pm \frac{1}{4} \times 32 | S \in \mathbb{S}\} = \{216, 200, 184, 168, 152, 136, 120, 104\}$. We show the evaluation results of SAN on $\mathbb{T}'$ with the proxy inference method in the bottom panel of Table 5. We also provide baseline models individually trained on more dense input resolutions in the top panel of Table 5. It is observed that in the corresponding column, the shaded value (in the bottom sub-table) from our SAN achieves higher accuracy on these unseen resolutions than all evaluation results from the baseline models (in the top sub-table), demonstrating its versatility to a wide range of test resolutions. In other words, SAN is competent in processing input images well with arbitrary resolutions during inference.

G Visualization of Privatized BN Layers

Regarding privatized BN layers of ResNet-18-based SAN for each training resolution, we visualize their trainable parameters (scale and shift parameter γ and β) and statistics (running mean μ and variance σ) in Fig. 1. Specifically, we compute the mean values of these parameters across channels. There exist eight residual blocks in ResNet-18 and each residual block contains two BN layers. We plot the same type of BN parameters from all the sixteen layers in one subfigure and make the observations that the mean values of BN parameters in each layer varies monotonically with respect to the training resolutions, *i.e.*, following an either ascending or descending distribution with the training resolutions increasing. This phenomenon partially interprets the proper approximation of the proxy inference method and inspires the following data-free inference method.

Table 5. Comparison of ResNet-18 baseline models (*top*) and ResNet-18-based-SAN (*bottom*), with a more dense setting of test resolutions. The shaded results have the same meaning as those in Table 1 of the main paper.

Train \ Test	216	200	184	168	152	136	120	104
224	70.40	68.28	67.69	64.55	63.18	57.60	54.20	45.26
208	70.11	69.77	68.12	66.81	63.77	61.44	55.67	50.27
192	70.88	69.05	69.04	66.52	65.75	60.83	58.41	49.98
176	70.56	70.50	69.35	68.44	66.25	64.14	59.55	54.95
160	70.43	68.94	69.70	67.13	67.50	63.49	61.89	54.46
144	69.89	69.84	69.47	68.84	67.61	66.16	62.71	59.08
128	68.55	66.97	68.78	66.66	67.91	64.49	64.68	57.83
112	66.37	66.63	67.37	67.44	67.16	66.49	64.46	62.26
96	59.24	57.66	62.56	60.64	64.58	61.92	64.63	60.18
Train \ Test	216	200	184	168	152	136	120	104
224	72.62	71.10	71.34	69.30	68.75	65.28	63.23	56.55
192	72.58	71.33	71.94	70.07	70.09	66.86	65.49	59.31
160	71.90	70.78	71.81	70.14	70.76	67.76	67.16	61.74
128	70.27	69.05	70.78	69.30	70.58	67.83	68.13	63.30
96	65.60	64.26	67.63	65.80	68.63	66.03	67.60	63.34

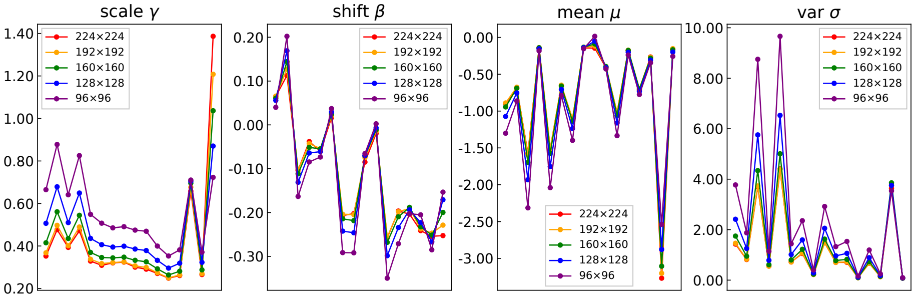

Fig. 1. Visualization of parameters and statistics of privatized BN layers in ResNet-18-based SAN. The x-axis represents the residual block index while the y-axis represents the mean value.

Table 6. Accuracy of ResNet-18-based-SAN with data-free ideal inference and proxy inference respectively. The results of proxy inference are identical to those in Table 1 of the main paper and Table 5 of the supplementary materials.

Inference \ Resolution	216	208	200	184	176	168	152	144	136	120	112	104
Data-Free Ideal	72.66	72.08	71.32	72.00	71.15	70.20	70.78	69.35	67.91	68.11	65.95	63.38
Proxy	72.62	72.10	71.33	71.94	71.04	70.14	70.76	69.35	67.83	68.13	65.78	63.34

H Data-Free Ideal Inference via BN Interpolation

In consideration of the monotonicity and continuity of BN parameters as visualized in Section G, we come up with another solution to derive BN parameters for the ideal inference method. For an arbitrary test resolution T, we first search the nearest training resolutions located in its two sides, represented as $S_{floor}(T)$ and $S_{ceil}(T)$ where $S_{floor}(T) < S_{ceil}(T)$. Then we compute the BN parameters via a linear interpolation between these two groups of BN parameters, $\mathrm{BN}^{(S_{floor}(T))}$ and $\mathrm{BN}^{(S_{ceil}(T))}$. Specifically, the interpolation of BN is defined as

$$\mathrm{BN}^{(T)} = \frac{T - S_{floor}(T)}{S_{ceil}(T) - S_{floor}(T)} \mathrm{BN}^{(S_{ceil}(T))} + \frac{S_{ceil}(T) - T}{S_{ceil}(T) - S_{floor}(T)} \mathrm{BN}^{(S_{floor}(T))}, \tag{1}$$

where the parameters (γ, β, μ and σ) of BN are interpolated separately. The main network $\mathcal{N}^{(T)}$ could be parameterized with $\mathcal{M}(\varepsilon^{(T)})$, $\mathrm{BN}^{(T)}$ and FC. It is noteworthy that this method does not depend on recalculating statistics of the training set anymore, so it is called *data-free ideal inference*. The detailed inference process with this data-free BN computation method is depicted in Algorithm 4. We report the performance of this method on the set of interpolated test resolutions $\mathbb{T} \cup \mathbb{T}'$ and compare it to the proxy inference method (mainly used in our experiments) in Table 6. We could see that data-free ideal inference is a more independent method compared with the original ideal inference while retaining the performance.

I Border and Round-off Effects

It has drawn our attention that the curve of performance in Fig. 1 of the main paper tends to fluctuate at certain test resolutions that are not divisible by 32 (modern neural networks usually have a down-sampling rate of 1/32, hence these resolutions lead to discarded pixels of the intermediate feature maps), resembling the discovery of border and round-off effects discussed in Appendix of [4]. It is also noted that the aforementioned resolutions non-divisible by 32 are exactly the test resolutions in $\mathbb{T} \cup \mathbb{T}'$ sandwiched by the training ones in $\mathbb{S} = \{S_1, S_2, S_3, S_4, S_5\} = \{224, 192, 160, 128, 96\}$. Our SAN model could achieve strong performance even at these interpolation points of training resolutions as demonstrated in Table 6, which allows the evaluated images to be universally scalable. We further conduct experiments to prove by contradiction that the degradation phenomenon at these test resolutions does not arise from interpolation of training resolutions but essentially owing to the discrete nature of convolutional kernel sizes and strides. For the ResNet-18-based-SAN, we select three resolutions $\mathbb{S} = \{S_1, S_2, S_3\} = \{224, 160, 96\}$ for training and interpolate test resolutions which *are* multiples of 32 during inference, shaping convex accuracy curves *without* fluctuation in Fig. 2.

To alleviate the round-off effect for baseline models, we further individually train them on these test resolutions from scratch, establishing a stronger baseline

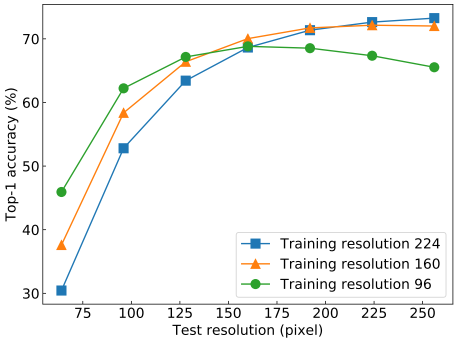

Fig. 2. Validation accuracy curves of ResNet-18-based SAN evaluated on a spectrum of test resolutions which *are* divisible by 32. The training resolutions are set to 224, 160 and 96, so test resolutions 256, 192, 128 and 64 are still interpolated or extrapolated operating points yet divisible by 32.

Table 7. Comparison of ResNet-18 baseline models individually trained on the test resolutions and ResNet-18-based-SAN with proxy inference.

Method \ Resolution	216	208	200	184	176	168	152	144	136	120	112	104
baseline	70.99	70.59	70.50	70.09	69.49	69.25	67.61	68.23	67.19	65.45	64.69	62.82
SAN	72.62	72.10	71.33	71.94	71.04	70.14	70.76	69.35	67.83	68.13	65.78	63.34

for comparison. The results are compared with our SAN (with proxy inference) again in Table 7. Even compared to these better-performing baseline models that are pre-trained on the test resolutions, our SAN models still show superiority regarding the top-1 validation accuracy on ImageNet.

J Training Baseline Longer

One may contradict that the training budget of SAN is much larger than a *single* baseline model. However, this contradiction might overlook that although SAN training is based on more than one main network, it competes against *as many baseline models as possible* after a single run of training. Therefore, we would like to argue that our comparison in the main paper follows a fair setting and a series of baseline models may even consume more computational resources than SAN in total. For instance, we simultaneously train on 5 resolutions for 150 epochs, then the training cost is roughly the same as separately training 5 baseline models on these resolutions, each for 150 epochs. In Fig.1 and Table 1 of the main paper, our single SAN is compared to 5 such models respectively at their theoretically best operating point (when test resolution meets the training one). Furthermore, if another unseen-sized image comes, SAN could directly perform the inference. But we may need to retrain a model on the new test resolution to obtain a near-optimal result for baseline, just like what the 5 previous baseline models do. Now, the total training cost of baseline models has surpassed SAN.

Table 8. Accuracy of ResNet-18 baseline models on a spectrum of test resolutions when trained for 450 epochs, *i.e.*, 3 × 150 epochs.

Train \ Test	256	224	208	192	176	160	144	128	112	96	64
224	73.55	72.63	71.18	70.63	68.59	66.64	63.00	59.13	52.40	44.41	21.25
192	73.04	72.70	71.84	71.56	69.98	68.55	66.13	62.48	57.25	49.40	25.04
160	71.34	71.92	71.39	71.46	70.37	70.16	67.68	65.46	61.00	54.62	30.50
128	65.13	67.41	67.31	68.73	68.06	69.02	67.72	67.37	63.55	59.16	37.06
96	47.44	53.72	53.41	59.18	58.99	63.64	63.00	65.73	63.82	63.35	46.82

On the other hand, if one consists on considering *only a single baseline model* for comparison, we train each baseline model for 3× epochs (roughly the same wall-clock time as SAN training, sufficient for convergence), but only obtain marginal gains compared to the original baselines (cf. Table 1 top-left in the main paper), as illustrated in Table 8. More importantly, they still perform poorly on other test resolutions with the enlarged training budget, due to possibly over-fitting to the training resolution.

References

1. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q.V., Adam, H.: Searching for mobilenetv3. In: ICCV (2019)
2. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv e-prints arXiv:1704.04861 (Apr 2017)
3. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: CVPR (2018)
4. Touvron, H., Vedaldi, A., Douze, M., Jégou, H.: Fixing the train-test resolution discrepancy. In: NeurIPS (2019)
5. Yu, J., Huang, T.S.: Universally slimmable networks and improved training techniques. In: ICCV (2019)
6. Yu, J., Yang, L., Xu, N., Yang, J., Huang, T.: Slimmable neural networks. In: ICLR (2019)