A. Datasets and Experiment Settings

Dataset CIFAR-10 has 50 thousand training images and 10 thousand testing images in 10 classes with resolution 32x32. CIFAR-100 has the same number of training/testing images but in 100 classes. ImageNet-1k has over 1.2 million training images and 50 thousand validation images in 1000 classes. We use the official training/validation split in our experiments.

Augmentation We use the following augmentations as in [38]: mix-up [64], label-smoothing [48], random erasing [68], random crop/resize/flip/lighting and AutoAugment [10].

Optimizer For all experiments, we use SGD optimizer with momentum 0.9; weight decay 5e-4 for CIFAR-10/100, 4e-5 for ImageNet; initial learning rate 0.1 with batch size 256; cosine learning rate decay [27]. We train models up to 1440 epochs in CIFAR-10/100, 480 epochs in ImageNet. Following previous works [2, 21, 5], we use EfficientNet-B3 as teacher networks when training ZenNets.

B. Implementation

Our code is implemented in PyTorch. The synflow implementation is available from `https://github.com/mohsaied/zero-cost-nas/blob/main/foresight/pruners/measures/synflow.py`. The official TE-NAS score implementation is available from `https://github.com/VITA-Group/TENAS/blob/main/lib/procedures`. The official NASWOT implementation is available from `https://github.com/BayesWatch/nas-without-training`. Our searching and training code are released on `https://github.com/idstcv/ZenNAS`.

C. Additional Figures

We test the performance of ZenNets on devices other than NVIDIA V100 GPU. The two hardware platforms are considered. NVIDIA T4 is an industrial level GPU optimized for INT8 inference. All networks are exported to TensorRT engine at precision INT8 to benchmark their inference speed on T4. Google Pixel2 is a modern cell phone with moderate powerful mobile GPU. In Figure 5 and Figure 6, we report the inference speed of ZenNets on T4 and Pixel2 as well as several SOTA models. The best ZenNet-1.2ms is 10.9x times faster than EfficientNet on NVIDIA T4, 1.6x times faster on Pixel2.

The evolutionary processes of optimizing zero-shot proxies are plotted in Figure 7, 8, 9, 10, 11.

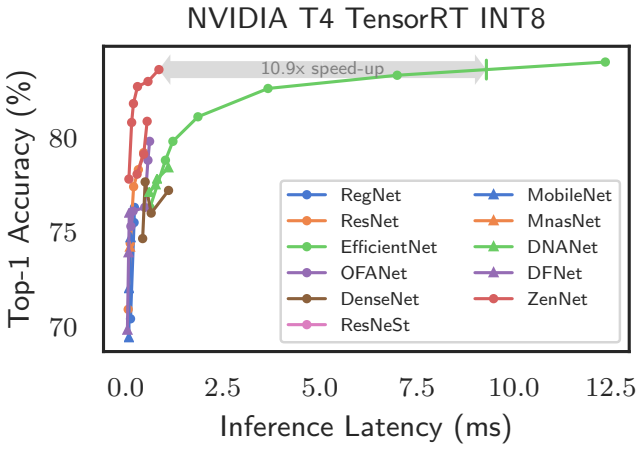


Figure 5. ZenNets top-1 accuracy on ImageNet-1k v.s. inference latency (milliseconds per image) on NVIDIA T4, TensorRT INT8, batch size 64. ZenNet-0.8ms∼1.2ms and ZenNet-400M-SE∼900M-SE are plotted as two separated curves.

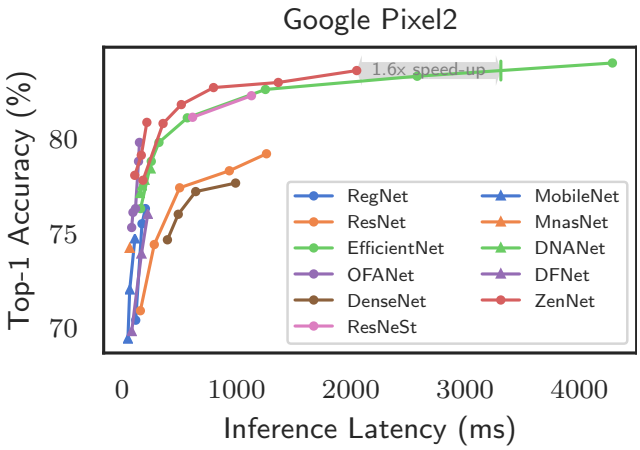


Figure 6. ZenNets top-1 accuracy on ImageNet-1k v.s. inference latency (milliseconds per image) on Google Pixel2, single image. ZenNet-0.8ms∼1.2ms and ZenNet-400M-SE∼900M-SE are plotted as two separated curves.

D. Zen-NAS on CIFAR

Following previous works, we use Zen-NAS to optimize model size on CIFAR-10 and CIFAR-100 datasets. We use Search Space I in this experiment. We constrain the number of network parameters within $\{1.0\,\text{M}, 2.0\,\text{M}\}$. The resultant networks are labeled as ZenNet-1.0M/2.0M. Table 4 summarized our results. We compare several popular NAS-designed models for CIFAR-10/CIFAR-100 in Figure 13, including AmoebaNet [41], DARTS [26], P-DARTS [8], SNAS [59], NASNet-A [70], ENAS[38], PNAS [25], ProxylessNAS [6]. ZenNets outperform baseline methods by $30\% \sim 50\%$ parameter reduction while achieving the same accuracies.

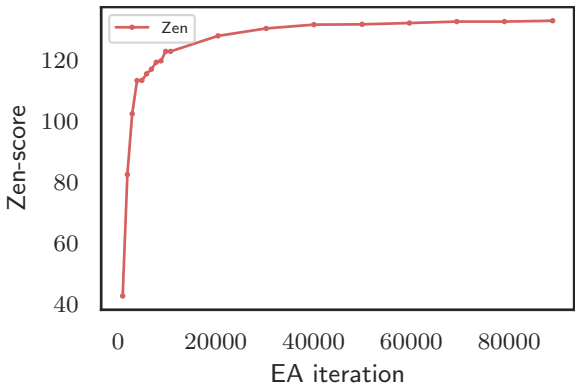


Figure 7. NAS process for maximizing Zen-Score. x-axis: number of evolutionary iterations. y-axis: Largest Zen-Score in the current population.

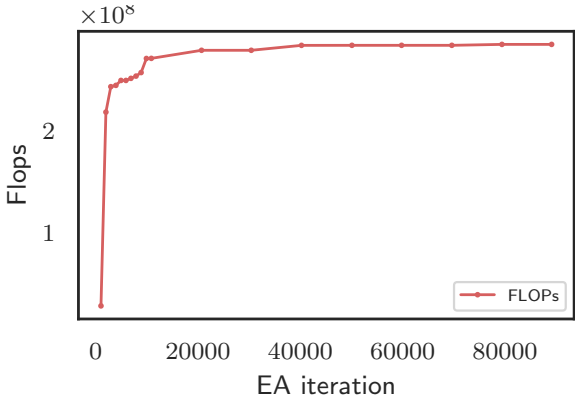


Figure 8. NAS process for maximizing FLOPs. x-axis: number of evolutionary iterations. y-axis: Largest FLOPs in the current population.

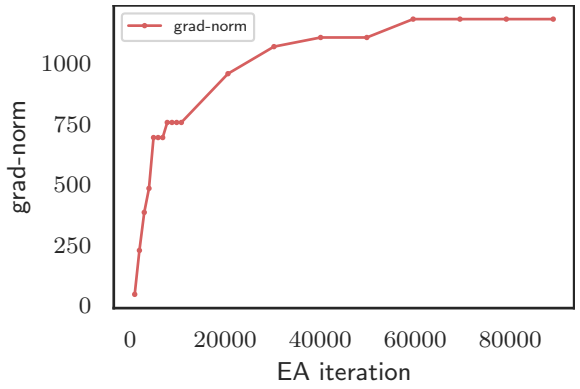


Figure 9. NAS process for maximizing grad-norm. x-axis: number of evolutionary iterations. y-axis: Largest grad-norm in the current population.

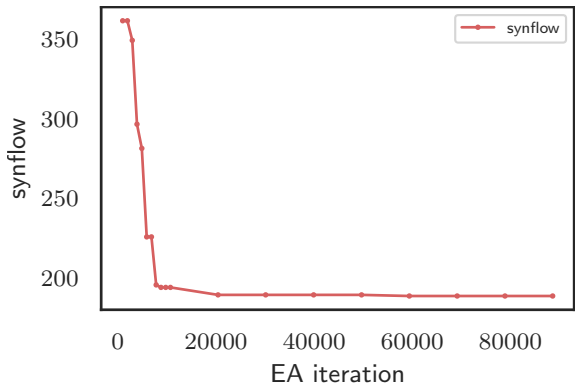


Figure 10. NAS process for maximizing synflow. x-axis: number of evolutionary iterations. y-axis: Smallest synflow in the current population.

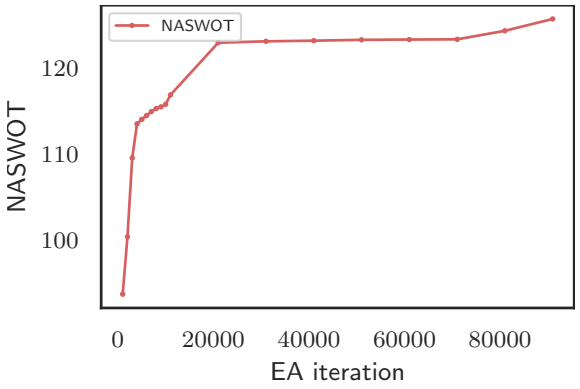


Figure 11. NAS process for maximizing NASWOT. x-axis: number of evolutionary iterations. y-axis: Largest NASWOT score in the current population.

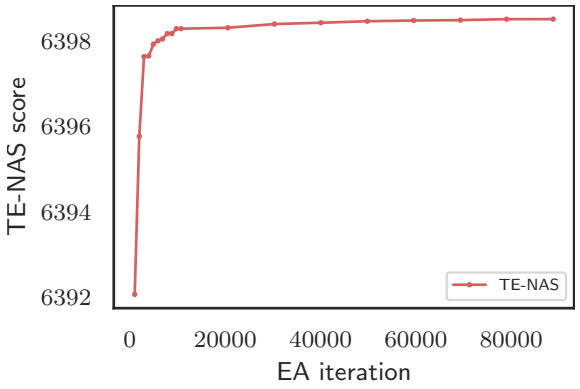


Figure 12. NAS process for maximizing TE-NAS score. x-axis: number of evolutionary iterations. y-axis: Largest TE-NAS score in the current population. The NTK score in TE-NAS is the smaller the better. Therefore we use R_N − NTK as TE-score in EA. This is slightly different from [7] where the rank of NTK is used as score.

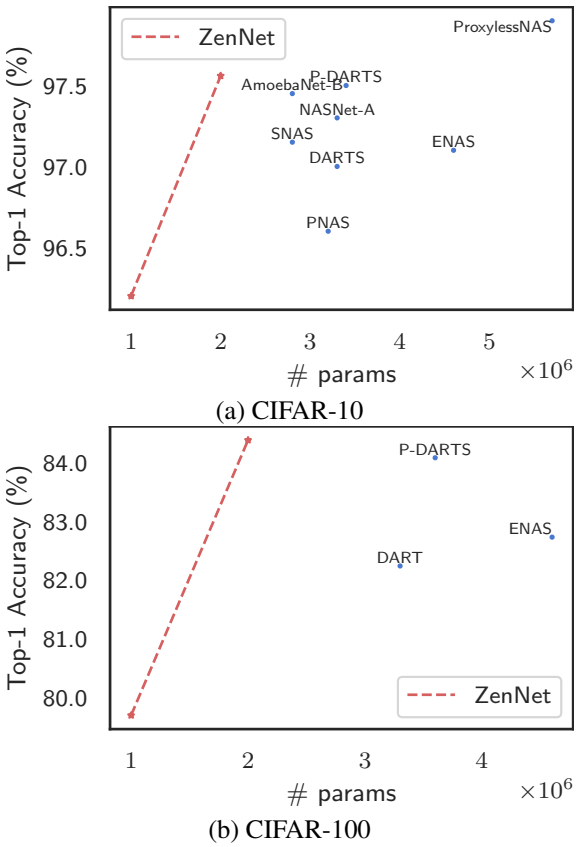


Figure 13. ZenNet accuracy v.s. model size (# params) on CIFAR-10 and CIFAR-100.

model	# params	FLOPs	C10	C100
ZenNet-1.0M	1.0 M	162 M	96.5%	80.1%
ZenNet-2.0M	2.0 M	487 M	97.5%	84.4%

Table 4. ZenNet-1.0M/2.0M on CIFAR-10 (C10) and CIFAR-100 (C100).

E. Zen-Scores and Accuracies of ResNets under Fair Training Setting

Model	FLOPs	# Params	Zen-Score
ResNet-18	1.82G	11.7M	59.53
ResNet-34	3.67G	21.8M	112.32
ResNet-50	4.12G	25.5M	140.3
ResNet-101	7.85G	44.5M	287.87
ResNet-152	11.9G	60.2M	433.57

Table 5. Zen-Scores of ResNets.

ResNets are widely used in computer vision. It is interesting to understand the ResNets via Zen-Score analysis. We report the Zen-Scores of ResNets in Table 5. In Figure 14, we plot the Zen-Score against top-1 accuracy of ResNet and ZenNet on ImageNet. From the figure, it is clearly that even for the same model, the training method matters a lot. There is considerable performance gain of ResNets after using our enhanced training methods. The Zen-Scores positively correlate to the top-1 accuracies for both ResNet and ZenNets.

Next we show that the Zen-Scores is well-aligned with top-1 accuracies across different models. We consider two baselines in Table 6. The 2nd column reports the top-1 accuracies obtained in the ResNet original paper [15]. We found that these models are under-trained. We use enhanced training methods to train ResNets in the same way as we trained ZenNets. The corresponding top-1 accuracies are reported in the 3rd column.

Model	Top-1 [15]	Top-1 (ours)
ResNet-18	70.9%	72.1%
ResNet-34	74.4%	76.3%
ResNet-50	77.4%	79.0%
ResNet-101	78.3%	81.0%
ResNet-152	79.2%	82.3%

Table 6. Top-1 accuracies of ResNets. Reported by [15] and using enhanced training methods we used in this paper.

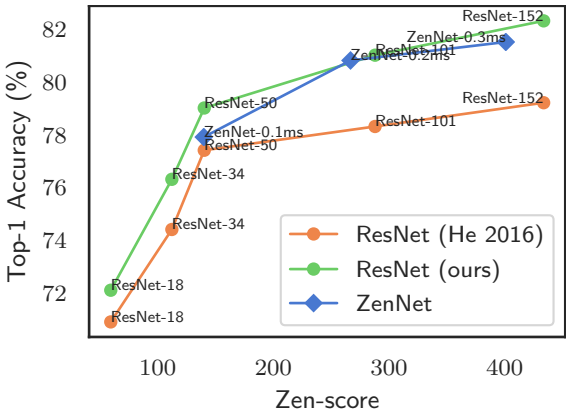
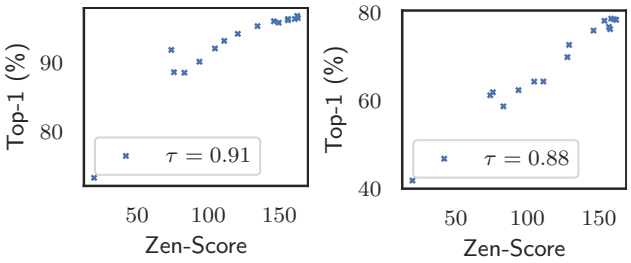


Figure 14. ResNet/ZenNet Zen-Score v.s. top-1 accuracy on ImageNet.

F. Effectiveness of Zen-Score

We show that Zen-Score effectively indicates the model accuracy during the evolutionary search. In the searching process of ZenNet-1.0M, we uniformly sample 16 structures from the evolutionary population. These structures



(a) CIFAR-10 (b) CIFAR-100

Figure 15. Zen-Score v.s. top-1 accuracy, 16 randomly sampled structures generated from ResNet-50, with Kendall's τ-score between accuracy and Zen-Score.

have different number of channels and layers. Then the sampled structures are trained on CIFAR-10/CIFAR-100. The top-1 accuracy v.s. Zen-Score are plotted in Figure 15. The Zen-Scores effectively indicates the network accuracies, especially in high-precision regimes.

G. FLOPs/Params/Latency of ZenNets in Table 1

proxy	params	FLOPs	latency
Zen-Score	1.0M	170M	0.15ms
FLOPs	1.0M	285M	0.07ms
grad	0.2M	41M	0.14ms
synflow	1.0M	104M	0.11ms
TE-Score	1.0M	118M	0.08ms
NASWOT	1.0M	304M	0.25ms
Random	1.0M	110 M	0.09ms

Latency is measured on NVIDIA V100 FP16 batch size 64. 'grad' cannot find a model near params≈1M.

H. Zen-NAS for Object Detection

We apply Zen-NAS in designing ResNet-like networks for MS-COCO Object Detection dataset, aligned with ResNet-50 for the same mAP and/or the same speed. Following common practice, the resolution is 480x480, pre-trained on ImageNet-1k for 12 epochs, using FCOS framework. ZenNets achieve better mAP and/or faster inference speed.

backbone	params	FLOPs	latency	mAP
ResNet-50	25.5M	18.8G	0.80ms	0.387
ZenNet-same-mAP	20.1M	**9.25G**	**0.47ms**	0.385
ZenNet-same-speed	61.5M	19.8G	0.76ms	**0.403**

I. Proof of Theorem 1

We introduce a few more notations for our proof. Suppose the network has L convolutional layers. The t-th layer has m_{t-1} input channels and m_t output channels. The convolutional kernel is $\boldsymbol{\theta}_t \in \mathbb{R}^{m_t \times m_{-1t} \times k \times k}$. The image resolution is $H \times W$. The mini-batch size is B. The output feature map of the t-th layer is $\boldsymbol{x}_t$. We use $\boldsymbol{x}_t^{(i,b,h,w)}$ to denote the pixel of $\boldsymbol{x}_t$ in the i-th channel, b-th image at cooridinate (h, w). $\mathcal{N}(\mu, \sigma)$ denotes Gaussian distribution with mean μ and variance σ^2. For random variables z, a and a constant b, the notation $z = a \pm b$ means $|z - a| \leq b$. To avoid notation clutter, we use $C_{\log}^{1/\delta}(\cdot)$ to denote some logarithmic polynomial in $1/\delta$ and some other variables. Since the order of these variables in $C_{\log}^{1/\delta}(\cdot)$ is logarithmic, they do not alternate the polynomial order of our bounds.

The input image $\boldsymbol{x}_0$ are sampled from $\mathcal{N}(0, 1)$. In a vanilla network without BN layer, the feature map $\bar{\boldsymbol{x}}_t$ is generated by the following forward inference process:

$$\begin{aligned}\bar{\boldsymbol{x}}_0 =& \boldsymbol{x}_0\\ \bar{\boldsymbol{x}}_t =& [\boldsymbol{\theta}_t * \bar{\boldsymbol{x}}_{t-1}]_+\end{aligned}$$

where $*$ is the convolutional operator, $[z]_+ = \max(z, 0)$.

In Zen-Score computation, BN layer is inserted after every convolutional operator. The forward inference now becomes:

$$\boldsymbol{g}_t = \boldsymbol{\theta}_t * \boldsymbol{x}_{t-1} \tag{5}$$

$$[\boldsymbol{\sigma}_t^{(i)}]^2 = \frac{1}{BHW} \sum_{b,h,w} [\boldsymbol{g}_t^{(i,b,h,w)}]^2 \tag{6}$$

$$\bar{\sigma}_t^2 = \frac{1}{m_t} \sum_{i=1}^{m_t} [\boldsymbol{\sigma}_t^{(i)}]^2 \tag{7}$$

$$\boldsymbol{x}_t^{(i)} = \left[\frac{\boldsymbol{g}_t^{(i)}}{\boldsymbol{\sigma}_t^{(i)}}\right]_+ = \frac{1}{\boldsymbol{\sigma}_t^{(i)}} [\boldsymbol{g}_t^{(i)}]_+ \,. \tag{8}$$

Please note that in Eq. (8), we use a modified BN layer instead of the standard BN, where we do not subtract mean value in the normalization step. This will greatly simply the proof. If the reader is concerned about this, it is straightforward to replace all BN layers with our modified BN layers so that the computational process exactly follows our proof. In practice, we did not observe noticable difference by switching between two BN layers because the mean value of mini-batch is very close to zero.

To show that the Zen-Score computed on BN-enabled network $f(\boldsymbol{x}_0) = \boldsymbol{x}_L$ approximates the Φ-score computed on BN-free network $\bar{f}(\boldsymbol{x}_0) = \bar{\boldsymbol{x}}_L$, we only need to prove

$$(\prod_{t=1}^{L} \bar{\sigma}_t)^2 \mathbb{E}_\theta \|\boldsymbol{x}_L\|^2 \approx \mathbb{E}_\theta \|\bar{\boldsymbol{x}}_L\|^2 \,. \tag{9}$$

In deed, when Eq. (9) holds true, by taking gradient w.r.t. $\boldsymbol{x}$ on both side, the proof is then completed. To prove Eq. (9), we need the following theorems and lemmas.

I.1. Useful Theorems and Lemmas

The first theorem is Bernstein's inequality. It can be found in many statistical textbooks, such as [53, Theorem 2.8.1].

Theorem 2 (Bernstein's inequality)**.** *Let $x_1, x_2, \cdots, x_N$ be independent bounded random variables of mean zero, variance σ. $|x_i| \leq K$ for all $i \in \{1, 2, \cdots, N\}$. $\boldsymbol{a} = [a_1, a_2, \cdots, a_N]$ is a fixed N-dimensional vector. Then $\forall t \geq 0$,*

$$\mathbb{P}(\left|\sum_{i=1}^{N} a_i x_i\right| \geq t) \leq 2 \exp\left[-c \min\left(\frac{t^2}{\sigma^2 \|\boldsymbol{a}\|_2^2}, \frac{t}{K\|\boldsymbol{a}\|_\infty}\right)\right] \,.$$

A direct corollary gives the upper bound of sum of random variables.

Corollary 1. *Under the same setting of Theorem 2, with probability at least* $1-\delta$*,*

$$\left|\sum_{i=1}^{N} a_i x_i\right| \leq C_{\log}^{1/\delta}(\cdot)\sigma\|\boldsymbol{a}\|_2 \ .$$

Proof. Let

$$\begin{aligned}\delta &\triangleq 2\exp\left[-c\min\left(\frac{t^2}{\sigma^2\|\boldsymbol{a}\|_2^2}, \frac{t}{K\|\boldsymbol{a}\|_\infty}\right)\right] \\ &= \max\left\{2\exp\left[-c\frac{t^2}{\sigma^2\|\boldsymbol{a}\|_2^2}\right], 2\exp\left[-c\frac{t}{K\|\boldsymbol{a}\|_\infty}\right]\right\} \ .\end{aligned}$$

That is,

$$\begin{aligned}&\delta \geq 2\exp\left[-c\frac{t^2}{\sigma^2\|\boldsymbol{a}\|_2^2}\right] \\ \Leftrightarrow &t \leq \sqrt{\frac{1}{c}\log(2/\delta)}\sigma\|\boldsymbol{a}\|_2 = C_{\log}^{1/\delta}(\cdot)\sigma\|\boldsymbol{a}\|_2 \ ,\end{aligned}$$

and

$$\begin{aligned}&\delta \geq 2\exp\left[-c\frac{t}{K\|\boldsymbol{a}\|_\infty}\right] \\ \Leftrightarrow &t \leq \frac{1}{c}\log(2/\delta)K\|\boldsymbol{a}\|_\infty = C_{\log}^{1/\delta}(\cdot)K\|\boldsymbol{a}\|_\infty \ .\end{aligned}$$

Therefore, with probability at least $1-\delta$,

$$\begin{aligned}\left|\sum_{i=1}^{N} a_i x_i\right| \leq &\min\{C_{\log}^{1/\delta}(\cdot)\sigma\|\boldsymbol{a}\|_2, C_{\log}^{1/\delta}(\cdot)K\|\boldsymbol{a}\|_\infty\} \\ \leq &C_{\log}^{1/\delta}(\cdot)\min\{\sigma\|\boldsymbol{a}\|_2, K\|\boldsymbol{a}\|_\infty\} \ .\end{aligned}$$

That is,

$$\left|\sum_{i=1}^{N} a_i x_i\right| \leq C_{\log}^{1/\delta}(\cdot)\sigma\|\boldsymbol{a}\|_2 \ .$$

□

When the random variables are sampled from Gaussian distribution, it is more convenient to use the following tighter bound.

Theorem 3. *Let* $x_1, x_2, \cdots, x_N$ *be sampled from* $\mathcal{N}(0,\sigma)$*,* $\boldsymbol{a} \in \mathbb{R}^N$ *be a fixed a vector. Then* $\forall t \geq 0$*,*

$$\mathbb{P}(\left|\sum_{i=1}^{N} a_i x_i\right| > t) \leq \exp\left[-\frac{t^2}{2\sigma^2\|a\|_2^2}\right] \ .$$

Corollary 2. *With probability at least* $1-\delta$*,*

$$\left|\sum_{i=1}^{N} a_i x_i\right| \leq \sqrt{2\log(1/\delta)}\sigma\|\boldsymbol{a}\|_2 = C_{\log}^{1/\delta}(\cdot)\sigma\|\boldsymbol{a}\|_2 \ .$$

The proof is simple since the sum of Gaussian random variables is still Gaussian random variables.

The following two lemmas are critical in our lower bound analysis. The proof is straightforward once using the symmetric property of random variable distribution.

Lemma 3. *Suppose* $x \in \mathbb{R}$ *is a mean zero, variance* σ^2 *random variable with symmetric distribution. Then* $\mathbb{E}[x]_+^2 = 4\sigma^2/4$*.*

Lemma 4. *Suppose* $\theta_i \sim \mathcal{N}(0,1)$*.* $\|\boldsymbol{x}\| = \|\boldsymbol{y}\|$ *are two fixed vectors. Then*

$$\mathbb{E}_\theta[\sum_i \theta_i x_i]_+^2 = \frac{1}{2}\mathbb{E}_\theta[\sum_i \theta_i x_i]^2 = \mathbb{E}_\theta[\sum_i \theta_i y_i]_+^2 \ .$$

I.2. Proof of Eq. (9)

Since $\boldsymbol{x}_0 \sim \mathcal{N}(0,1)$, with probability at least $1-\delta$, $\|\boldsymbol{x}_0\|_\infty \leq C_{\log}^{1/\delta}(\cdot) \triangleq K_0$ for some constant K_0. Now suppose at layer t, $\|\boldsymbol{x}_{t-1}\|_\infty \leq K_{t-1}$. The following lemma shows that after convolution, $\|\boldsymbol{g}_t\|_\infty$ is also bounded with high probability.

Lemma 5. *Let $\boldsymbol{\theta}^{(i,b,h,w)} \sim \mathcal{N}(0,1)$, $\boldsymbol{\theta}_t \in \mathbb{R}^{m_t \times m_{t-1} \times k \times k}$. For fixed $\boldsymbol{x}_{t-1} \in \mathbb{R}^{m_{t-1} \times B \times H \times W}$, $\boldsymbol{g}_t \triangleq \boldsymbol{\theta}_t * \boldsymbol{x}_{t-1}$. Then with probability at least $1-\delta$,*

$$\|\boldsymbol{g}_t\|_\infty \leq C_{\log}^{1/\delta}(\cdot)^2 k\sqrt{m_{t-1}} K_{t-1} \, .$$

Proof. Let us consider $\boldsymbol{g}_t^{(j,b,\alpha,\beta)}$, that is, the j-th channel, b-th image, at pixel (α,β). For any $1 \leq j \leq m_t$, $1 \leq \alpha \leq H$, $1 \leq \beta \leq W$,

$$\boldsymbol{g}_t^{(j,b,\alpha,\beta)} = \sum_{i=1}^{m_{-1t}} \sum_{p=-\frac{k-1}{2}}^{\frac{k-1}{2}} \sum_{q=-\frac{k-1}{2}}^{\frac{k-1}{2}} \boldsymbol{\theta}_t^{(j,i,p,q)} \boldsymbol{x}_{t-1}^{(i,b,\alpha+p,\beta+p)}$$

Clearly,

$$\mathbb{E}_\theta \boldsymbol{g}_t^{(j,b,\alpha,\beta)} = 0 \, .$$

According to Corollary 2,

$$\begin{aligned} |\boldsymbol{g}_t^{(j,b,\alpha,\beta)}| &\leq C_{\log}^{1/\delta}(\cdot) C_{\log}^{1/\delta}(\cdot) K_{t-1}\sqrt{m_{t-1}} k \\ &\leq C_{\log}^{1/\delta}(\cdot)^2 k\sqrt{m_{t-1}} K_{t-1} \, . \end{aligned}$$

□

The variance of $\boldsymbol{g}_t$ is bounded with high probability too.

Lemma 6. *With probability at least $1-\delta$,*

$$\begin{aligned} \mathbb{E}[\boldsymbol{g}_t^{(j,b,\alpha,\beta)}]^2 =& \sigma_t^* \pm C_{\log}^{1/\delta}(\cdot) k\sqrt{m_{t-1}} K_{t-1} \\ \sigma_t^{*2} \triangleq& \frac{1}{4} m_{t-1} k^2 \, . \end{aligned}$$

Proof. By definition,

$$\boldsymbol{g}_t^{(j,b,\alpha,\beta)} = \sum_{i=1}^{m_{-1t}} \sum_{p=-\frac{k-1}{2}}^{\frac{k-1}{2}} \sum_{q=-\frac{k-1}{2}}^{\frac{k-1}{2}} \boldsymbol{\theta}_t^{(j,i,p,q)} \boldsymbol{x}_{t-1}^{(i,b,\alpha+p,\beta+p)}$$

Clearly, $\boldsymbol{g}_t^{(j,b,\alpha,\beta)}$ is Gaussian random variable with zero-mean.

$$\mathbb{E}[\boldsymbol{g}_t^{(j,b,\alpha,\beta)}]^2 = \sum_{i=1}^{m_{t-1}} \sum_{p=-\frac{k-1}{2}}^{\frac{k-1}{2}} \sum_{q=-\frac{k-1}{2}}^{\frac{k-1}{2}} [\boldsymbol{x}_{t-1}^{(i,b,\alpha+p,\beta+p)}]^2 \, .$$

By Lemma 3,

$$\mathbb{E}[\boldsymbol{x}_{t-1}^{(i,b,\alpha+p,\beta+p)}]^2 = \frac{1}{4} \, .$$

Therefore,

$$\begin{aligned} &|\mathbb{E}[\boldsymbol{g}_t^{(j,b,\alpha,\beta)}]^2 - \frac{1}{4} m_{t-1} k^2| \\ \leq& C_{\log}^{1/\delta}(\cdot) k\sqrt{m_{t-1}} K_{t-1} \, . \end{aligned}$$

Define $\sigma_t^{*2} \triangleq \frac{1}{4} m_{t-1} k^2$, the proof is completed. □

Next we show that both $\sigma_t^{(i)}$ and $\bar{\sigma}_t$ concentrate around σ^*.

Lemma 7. *With probability* $1-\delta$*,*

$$\begin{aligned}[\boldsymbol{\sigma}_t^{(i)}]^2 =&(1\pm\epsilon_t)[\sigma_t^*]^2\\ \bar{\sigma}_t =&(1\pm\frac{\epsilon_t}{\sqrt{m_t}})[\sigma_t^*]^2\end{aligned}$$

where

$$\epsilon_t \triangleq 4C_{\log}^{1/\delta}(\cdot)^5\frac{1}{\sqrt{BHW}}K_{t-1}^2$$

Proof. Directly apply Corollary 1,

$$\begin{aligned}[\boldsymbol{\sigma}_t^{(i)}]^2 =&\mathbb{E}[\boldsymbol{g}_t^{(j,b,\alpha,\beta)}]^2 \pm C_{\log}^{1/\delta}(\cdot)\frac{1}{\sqrt{BHW}}\max\{[\boldsymbol{g}_t^{(j,b,\alpha,\beta)}]^2\}\\ =&\mathbb{E}[\boldsymbol{g}_t^{(j,b,\alpha,\beta)}]^2 \pm C_{\log}^{1/\delta}(\cdot)\frac{1}{\sqrt{BHW}}C_{\log}^{1/\delta}(\cdot)^4 m_{t-1}k^2K_{t-1}^2\\ =&[\sigma_t^*]^2 \pm \frac{1}{\sqrt{BHW}}C_{\log}^{1/\delta}(\cdot)^5 m_{t-1}k^2K_{t-1}^2\ .\end{aligned}$$

Similary,

$$\bar{\sigma}_t^2 =[\sigma_t^*]^2 \pm \frac{1}{\sqrt{m_tBHW}}C_{\log}^{1/\delta}(\cdot)^5 m_{t-1}k^2K_{t-1}^2\ .$$

Define

$$\begin{aligned}\epsilon_t \triangleq&\frac{1}{[\sigma_t^*]^2}\frac{1}{\sqrt{BHW}}C_{\log}^{1/\delta}(\cdot)^5 m_{t-1}k^2K_{t-1}^2\\ =&\frac{4}{m_{t-1}k^2}C_{\log}^{1/\delta}(\cdot)^5\frac{1}{\sqrt{BHW}}m_{t-1}k^2K_{t-1}^2\\ =&4C_{\log}^{1/\delta}(\cdot)^5\frac{1}{\sqrt{BHW}}K_{t-1}^2\end{aligned}$$

Then we have

$$\begin{aligned}[\boldsymbol{\sigma}_t^{(i)}]^2 =&(1\pm\epsilon_t)[\sigma_t^*]^2\\ \bar{\sigma}_t =&(1\pm\frac{\epsilon_t}{\sqrt{m_t}})[\sigma_t^*]^2\end{aligned}$$

□

Next is our main lemma.

Lemma 8. *Under the same setting of Lemma 7, with probability* $1-\delta$*,*

$$(\sigma_t^*)^2\|\boldsymbol{x}_t\|^2 = \frac{1}{1\pm\epsilon_t}\|\,[\boldsymbol{g}_t]_+\,\|^2\ .$$

Proof. By definition,

$$\begin{aligned}\|\boldsymbol{x}_t\|^2 =&\sum_i\left[\frac{1}{\boldsymbol{\sigma}_t^{(i)}}\right]^2\left[\boldsymbol{g}_t^{(i)}\right]_+^2\\ =&\sum_i\left[\frac{1}{(1\pm\epsilon_t)\sigma_t^*}\right]^2\left[\boldsymbol{g}_t^{(i)}\right]_+^2\end{aligned}$$

Then

$$\begin{aligned}&\frac{(\sigma_t^*)^2\|\boldsymbol{x}_t\|^2}{\sum_i\left[\boldsymbol{g}_t^{(i)}\right]_+^2}\\=&\frac{1}{\sum_i\left[\boldsymbol{g}_t^{(i)}\right]_+^2}\sum_i\left[\frac{\sigma_t^*}{\boldsymbol{\sigma}_t^{(i)}}\right]^2\left[\boldsymbol{g}_t^{(i)}\right]_+^2\end{aligned}$$

By Lemma 7, we have

$$\frac{1}{1+\epsilon_t}\leq\frac{\sigma_t^*}{\boldsymbol{\sigma}_t^{(i)}}\leq\frac{1}{1-\epsilon_t}$$

□

Finally, we inductively bound $|\boldsymbol{x}_t^{(i,b,h,w)}|$.

Lemma 9. *With probability at least* $1-\delta$*,*

$$\begin{aligned}|\boldsymbol{x}_t^{(i,b,h,w)}|&\leq\frac{C_{\log}^{1/\delta}(\cdot)^2}{\sqrt{(1-\epsilon_t)}}K_{t-1}\\K_t&\leq C_{\log}^{1/\delta}(\cdot)^{2t}\prod_{j=1}^{t}(1-\epsilon_j)^{-1/2}K_0\,.\end{aligned}$$

Proof. By definition,

$$\boldsymbol{x}_t^{(i,b,h,w)}=\frac{1}{\boldsymbol{\sigma}_t^{(i)}}[\boldsymbol{g}_t^{(i,b,h,w)}]_+$$

From Lemma 5,

$$[\boldsymbol{g}_t^{(i,b,h,w)}]_+\leq C_{\log}^{1/\delta}(\cdot)^2K_{t-1}\sqrt{m_{t-1}}k$$

From Lemma 7,

$$\begin{aligned}[\boldsymbol{\sigma}_t^{(i)}]^2&=(1\pm\epsilon_t)[\sigma_t^*]^2\\&=\frac{1}{4}(1\pm\epsilon_t)m_{t-1}k^2\end{aligned}$$

Then

$$\begin{aligned}|\boldsymbol{x}_t^{(i,b,h,w)}|\leq&\frac{C_{\log}^{1/\delta}(\cdot)^2K_{t-1}\sqrt{m_{t-1}}k}{\sqrt{\frac{1}{4}(1\pm\epsilon_t)m_{t-1}k^2}}\\\leq&\frac{C_{\log}^{1/\delta}(\cdot)^2K_{t-1}\sqrt{m_{t-1}}k}{\sqrt{\frac{1}{4}(1-\epsilon_t)m_{t-1}k^2}}\\\leq&2\frac{C_{\log}^{1/\delta}(\cdot)^2K_{t-1}}{\sqrt{(1-\epsilon_t)}}\\\to&\frac{C_{\log}^{1/\delta}(\cdot)^2K_{t-1}}{\sqrt{(1-\epsilon_t)}}\quad\text{absorb 2 into }C_{\log}^{1/\delta}(\cdot)\end{aligned}$$

Therefore,

$$K_t \triangleq \frac{C_{\log}^{1/\delta}(\cdot)^2 K_{t-1}}{\sqrt{(1-\epsilon_t)}}$$
$$\Rightarrow K_t = C_{\log}^{1/\delta}(\cdot)^{2t} \prod_{j=1}^{t} (1-\epsilon_j)^{-1/2} K_0$$

□

Combining all above together, we are now ready to prove Eq. (9).

Denote $z_0 = 1$. It is trivial to see that $z_0\|\boldsymbol{x}_0\|^2 = z_0\|\bar{\boldsymbol{x}}_t\|^2$. By induction, suppose at layer t, we already have $z_{t-1}\|\boldsymbol{x}_{t-1}\|^2 = \|\bar{\boldsymbol{x}}_{t-1}\|^2$. Using Lemma 4,

$$\begin{aligned}
\mathbb{E}_\theta\|\bar{\boldsymbol{x}}_t\|^2 =& \mathbb{E}_\theta\| \left[\boldsymbol{\theta}_t * \bar{\boldsymbol{x}}_{t-1}\right]_+ \|^2 \\
=& \mathbb{E}_\theta\| \left[\boldsymbol{\theta}_t * z_{t-1}\boldsymbol{x}_{t-1}\right]_+ \|^2 \\
=& z_{t-1}\mathbb{E}_\theta\| \left[\boldsymbol{\theta}_t * \boldsymbol{x}_{t-1}\right]_+ \|^2 \\
=& z_{t-1}\mathbb{E}_\theta\| \left[\boldsymbol{g}_t\right]_+ \|^2
\end{aligned}$$

On the other hand, from Lemma 8,

$$\begin{aligned}
\bar{\sigma}_t^2 z_{t-1}\|\boldsymbol{x}_t\|^2 =& z_{t-1}\frac{\bar{\sigma}_t^2}{(\sigma_t^*)^2}(\sigma_t^*)^2\|\boldsymbol{x}_t\|^2 \\
=& z_{t-1}(1 \pm \frac{\epsilon_t}{\sqrt{m_t}})(\sigma_t^*)^2\|\boldsymbol{x}_t\|^2 \quad \text{Lemma [lem:sigma-i-concentration]} \\
=& z_{t-1}(1 \pm \frac{\epsilon_t}{\sqrt{m_t}})\frac{1}{1\pm\epsilon_t}\| \left[\boldsymbol{g}_t\right]_+ \|^2 .
\end{aligned}$$

Therefore,

$$\mathbb{E}_\theta\{\bar{\sigma}_t^2 z_{t-1}\|\boldsymbol{x}_t\|^2\} = (1 \pm \frac{\epsilon_t}{\sqrt{m_t}})\frac{1}{1\pm\epsilon_t}\mathbb{E}_\theta\|\bar{\boldsymbol{x}}_t\|^2$$

By taking

$$z_t \triangleq \bar{\sigma}_t^2 z_{t-1} / [(1 \pm \frac{\epsilon_t}{\sqrt{m_t}})\frac{1}{1\pm\epsilon_t}] ,$$

we complete the induction of $z_t\|\boldsymbol{x}_t\|^2 = \|\bar{\boldsymbol{x}}_t\|^2$ for all t.

Chaining $t = \{1, 2, \cdots, L\}$, we get

$$\mathbb{E}_\theta\{(\prod_{t=1}^{L}\bar{\sigma}_t^2)\|\boldsymbol{x}_L\|^2\} = \prod_{t=1}^{L}\left[(1 \pm \frac{\epsilon_t}{\sqrt{m_t}})\frac{1}{1\pm\epsilon_t}\right]\mathbb{E}_\theta\|\bar{\boldsymbol{x}}_L\|^2 ,$$

where

$$\begin{aligned}
\epsilon_t \triangleq& 4C_{\log}^{1/\delta}(\cdot)^5\frac{1}{\sqrt{BHW}}K_{t-1}^2 \\
K_t \triangleq& C_{\log}^{1/\delta}(\cdot)^{2t}\prod_{j=1}^{t}(1-\epsilon_j)^{-1/2}K_0 .
\end{aligned}$$

Finally, integrate everything together, we have proved that, with probability at least $1-\delta$,

$$(\prod_{t=1}^{L}\bar{\sigma}_t^2)\mathbb{E}_\theta\{\|\boldsymbol{x}_L\|^2\} = \prod_{t=1}^{L}\left[(1 \pm \frac{\epsilon_t}{\sqrt{m_t}})\frac{1}{1\pm\epsilon_t}\right]\mathbb{E}_\theta\|\bar{\boldsymbol{x}}_L\|^2 .$$

That is,

$$\prod_{t=1}^{L}\left[(1-\frac{\epsilon_t}{\sqrt{m_t}})\frac{1}{1+\epsilon}\right] \leq \frac{(\prod_{t=1}^{L}\bar{\sigma}_t^2)\mathbb{E}_\theta\{\|\boldsymbol{x}_L\|^2\}}{\mathbb{E}_\theta\|\bar{\boldsymbol{x}}_L\|^2} \leq \prod_{t=1}^{L}\left[(1+\frac{\epsilon_t}{\sqrt{m_t}})\frac{1}{1-\epsilon_t}\right] .$$

To further simply the above results, we consider the asymptotic case where BHW is large enough. Then ϵ_t will be a small number. By first order approximation of binomial expansion, $(1+\epsilon)^L \approx 1+L\epsilon+\mathcal{O}(\epsilon^2)$. To see that ϵ_t is bounded by a small constant, we denote $\gamma_t \triangleq \max_{j\in[1,t]}\epsilon_j$. Then

$$K_t \leq \mathcal{O}[(1+\frac{t-1}{2}\gamma_{t-1})K_0]$$

$$\gamma_t \leq \mathcal{O}\{\frac{K_0}{\sqrt{BHW}}[(1+\frac{(t-1)}{2}\gamma_{t-1}]\} . \tag{10}$$

By the recursive equation Eq. (10), when $\gamma_{t-1} \leq \frac{2}{L-1}$,

$$\gamma_t \leq \mathcal{O}\{\frac{K_0}{\sqrt{BHW}}[(1+\frac{(t-1)}{2}\gamma_{t-1}]\} \leq \mathcal{O}\{\frac{2K_0}{\sqrt{BHW}}\} .$$

Therefore, by taking $\frac{2K_0}{\sqrt{BHW}} \leq \frac{2}{L}$, that is $BHW \geq \mathcal{O}\{L^2K_0^2\}$, we have

$$\epsilon = \max\epsilon_t \leq \mathcal{O}\{\frac{2K_0}{\sqrt{BHW}}\}$$

to be a small number.

When ϵ is a small number,

$$\begin{aligned}\frac{(\prod_{t=1}^{L}\bar{\sigma}_t^2)\mathbb{E}_\theta\{\|\boldsymbol{x}_L\|^2\}}{\mathbb{E}_\theta\|\bar{\boldsymbol{x}}_L\|^2} &\leq \prod_{t=1}^{L}\left[(1+\frac{\epsilon_t}{\sqrt{m_t}})\frac{1}{1-\epsilon_t}\right] \\ &\leq (1+\epsilon)^L(1-\epsilon_t)^{-L} \\ &\approx (1+L\epsilon)^2 .\end{aligned}$$

Similarly,

$$\frac{(\prod_{t=1}^{L}\bar{\sigma}_t^2)\mathbb{E}_\theta\{\|\boldsymbol{x}_L\|^2\}}{\mathbb{E}_\theta\|\bar{\boldsymbol{x}}_L\|^2} \geq (1-L\epsilon)^2 .$$

J. One Big Table of Networks on ImageNet

model	resolution	# params	FLOPs	Top-1 Acc	latency(ms)		
					V100	T4	Pixel2
RegNetY-200MF	224	3.2 M	200 M	70.4%	0.22	0.12	118.17
RegNetY-400MF	224	4.3 M	400 M	74.1%	0.44	0.17	181.09
RegNetY-600MF	224	6.1 M	600 M	75.5%	0.25	0.21	173.19
RegNetY-800MF	224	6.3 M	800 M	76.3%	0.31	0.22	202.66
ResNet-18	224	11.7 M	1.8 G	70.9%	0.13	0.06	158.70
ResNet-34	224	21.8 M	3.6 G	74.4%	0.22	0.11	280.44
ResNet-50	224	25.6 M	4.1 G	77.4%	0.40	0.20	502.43
ResNet-101	224	44.5 M	7.8 G	78.3%	0.66	0.32	937.11
ResNet-152	224	60.2 M	11.5 G	79.2%	0.94	0.46	1261.97
EfficientNet-B0	224	5.3 M	390 M	76.3%	0.35	0.62	160.72
EfficientNet-B1	240	7.8 M	700 M	78.8%	0.55	1.02	254.26
EfficientNet-B2	260	9.2 M	1.0 G	79.8%	0.64	1.21	321.45
EfficientNet-B3	300	12.0 M	1.8 G	81.1%	1.12	1.86	569.30
EfficientNet-B4	380	19.0 M	4.2 G	82.6%	2.33	3.66	1252.79
EfficientNet-B5	456	30.0 M	9.9 G	83.3%	4.49	6.99	2580.25
EfficientNet-B6	528	43.0 M	19.0 G	84.0%	7.64	12.36	4287.81
EfficientNet-B7	600	66.0 M	37.0 G	84.4%	13.73	†	8615.92
MobileNetV2-0.25	224	1.5 M	44 M	51.8%	0.08	0.04	16.71
MobileNetV2-0.5	224	2.0 M	108 M	64.4%	0.10	0.05	26.99
MobileNetV2-0.75	224	2.6 M	226 M	69.4%	0.15	0.08	49.78
MobileNetV2-1.0	224	3.5 M	320 M	72.0%	0.17	0.08	65.59
MobileNetV2-1.4	224	6.1 M	610 M	74.7%	0.24	0.12	110.70
MnasNet-1.0	224	4.4 M	330 M	74.2%	0.17	0.11	65.50
DNANet-a	224	4.2 M	348 M	77.1%	0.29	0.60	157.94
DNANet-b	224	4.9 M	406 M	77.5%	0.37	0.77	173.66
DNANet-c	224	5.3 M	466 M	77.8%	0.37	0.81	194.27
DNANet-d	224	6.4 M	611 M	78.4%	0.54	1.10	248.08
DFNet-1	224	8.5 M	746 M	69.8%	0.07	0.04	82.87
DFNet-2	224	18.0 M	1.8 G	73.9%	0.12	0.07	168.04
DFNet-2a	224	18.1 M	2.0 G	76.0%	0.19	0.09	223.20
OFANet-9ms	118	5.2 M	313 M	75.3%	0.14	0.13	82.69
OFANet-11ms	192	6.2 M	352 M	76.1%	0.17	0.19	94.17
OFANet-389M(+)	224	8.4 M	389 M	79.1%	0.26	0.49	116.34
OFANet-482M(+)	224	9.1 M	482 M	79.6%	0.33	0.57	142.76

OFANet-595M(+)	236	9.1 M	595 M	80.0%	0.41	0.61	150.83
OFANet-389M*	224	8.4 M	389 M	76.3%	0.26	0.49	116.34
OFANet-482M*	224	9.1 M	482 M	78.8%	0.33	0.57	142.76
OFANet-595M*	236	9.1 M	595 M	79.8%	0.41	0.61	150.83
DenseNet-121	224	8.0 M	2.9 G	74.7%	0.53	0.43	395.51
DenseNet-161	224	28.7 M	7.8 G	77.7%	1.06	0.50	991.61
DenseNet-169	224	14.1 M	3.4 G	76.0%	0.69	0.65	490.24
DenseNet-201	224	20.0 M	4.3 G	77.2%	0.89	1.10	642.98
ResNeSt-50	224	27.5 M	5.4 G	81.1%	0.76	‡	615.77
ResNeSt-101	224	48.3 M	10.2 G	82.3%	1.40	‡	1130.59
ZenNet-0.1ms	224	30.1 M	1.7 G	77.8%	0.10	0.08	181.7
ZenNet-0.2ms	224	49.7 M	3.4 G	80.8%	0.20	0.16	357.4
ZenNet-0.3ms	224	85.4 M	4.9 G	81.5%	0.30	0.26	517.0
ZenNet-0.5ms	224	118 M	8.3 G	82.7%	0.50	0.41	798.7
ZenNet-0.8ms	224	183 M	13.9 G	83.0%	0.80	0.57	1365.0
ZenNet-1.2ms	224	180 M	22.0 G	83.6%	1.20	0.85	2051.1
ZenNet-400M-SE	224	5.7 M	410 M	78.0%	0.248	0.39	87.9
ZenNet-600M-SE	224	7.1 M	611 M	79.1%	0.358	0.52	128.6
ZenNet-900M-SE	224	13.3 M	926 M	80.8%	0.55	0.55	215.68

Table 7: One big table of all networks referred in this work.
+: OFANet trained using supernet parameters as initialization.
∗: OFANet trained from scratch. We adopt this setting for fair comparison.
†: fail to run due to out of memory.
‡: official model implementation not supported by TensorRT.

K. Detail Structure of ZenNets

We list detail structure in Table 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18.

The 'block' column is the block type. 'Conv' is the standard convolution layer followed by BN and RELU. 'Res' is the residual block used in ResNet-18. 'Btn' is the residual bottleneck block used in ResNet-50. 'MB' is the MobileBlock used in MobileNet and EfficientNet. To be consistent with 'Btn' block, each 'MB' block is stacked by two MobileBlocks. That is, the kxk full convolutional layer in 'Btn' block is replaced by depth-wise convolution in 'MB' block. 'kernel' is the kernel size of kxk convolution layer in each block. 'in', 'out' and 'bottleneck' are numbers of input channels, output channels and bottleneck channels respectively. 'stride' is the stride of current block. '# layers' is the number of duplication of current block type.

block	kernel	in	out	stride	bottleneck	# layers
Conv	3	3	24	2	-	1
Res	3	24	32	2	64	1
Res	5	32	64	2	32	1
Res	5	64	168	2	96	1
Btn	5	168	320	1	120	1
Btn	5	320	640	2	304	3
Btn	5	640	512	1	384	1
Conv	1	512	2384	1	-	1

Table 8. ZenNet-0.1ms

block	kernel	in	out	stride	bottleneck	# layers
Conv	3	3	24	2	-	1
Btn	5	24	32	2	32	1
Btn	7	32	104	2	64	1
Btn	5	104	512	2	160	1
Btn	5	512	344	1	192	1
Btn	5	344	688	2	320	4
Btn	5	688	680	1	304	3
Conv	1	680	2552	1	-	1

Table 9. ZenNet-0.2ms

block	kernel	in	out	stride	bottleneck	# layers
Conv	3	3	24	2	-	1
Btn	5	24	64	2	32	1
Btn	3	64	128	2	128	1
Btn	7	128	432	2	128	1
Btn	5	432	272	1	160	1
Btn	5	272	848	2	384	4
Btn	5	848	848	1	320	3
Btn	5	848	456	1	320	3
Conv	1	456	6704	1	-	1

Table 10. ZenNet-0.3ms

block	kernel	in	out	stride	bottleneck	# layers
Conv	3	3	8	2	-	1
Btn	7	8	64	2	32	1
Btn	3	64	152	2	128	1
Btn	5	152	640	2	192	4
Btn	5	640	640	1	192	2
Btn	5	640	1536	2	384	4
Btn	5	1536	816	1	384	3
Btn	5	816	816	1	384	3
Conv	1	816	5304	1	-	1

Table 11. ZenNet-0.5ms

block	kernel	in	out	stride	bottleneck	# layers
Conv	3	3	16	2	-	1
Btn	5	16	64	2	64	1
Btn	3	64	240	2	128	2
Btn	7	240	640	2	160	3
Btn	7	640	768	1	192	4
Btn	5	768	1536	2	384	5
Btn	5	1536	1536	1	384	3
Btn	5	1536	2304	1	384	5
Conv	1	2304	4912	1	-	1

Table 12. ZenNet-0.8ms

block	kernel	in	out	stride	bottleneck	# layers
Conv	3	3	32	2	-	1
Btn	5	32	80	2	32	1
Btn	7	80	432	2	128	5
Btn	7	432	640	2	192	3
Btn	7	640	1008	1	160	5
Btn	7	1008	976	1	160	4
Btn	5	976	2304	2	384	5
Btn	5	2304	2496	1	384	5
Conv	1	2496	3072	1	-	1

Table 13. ZenNet-1.2ms

block	kernel	in	out	stride	bottleneck	expansion	# layers
Conv	3	3	16	2	-	-	1
MB	7	16	40	2	40	1	1
MB	7	40	64	2	64	1	1
MB	7	64	96	2	96	4	5
MB	7	96	224	2	224	2	5
Conv	1	224	2048	1	-	-	1

Table 14. ZenNet-400M-SE

block	kernel	in	out	stride	bottleneck	expansion	# layers
Conv	3	3	24	2	-	-	1
MB	7	24	48	2	48	1	1
MB	7	48	72	2	72	2	1
MB	7	72	96	2	88	6	5
MB	7	96	192	2	168	4	5
Conv	1	192	2048	1	-	-	1

Table 15. ZenNet-600M-SE

block	kernel	in	out	stride	bottleneck	expansion	# layers
Conv	3	3	16	2	-	-	1
MB	7	16	48	2	72	1	1
MB	7	48	72	2	64	2	3
MB	7	72	152	2	144	2	3
MB	7	152	360	2	352	2	4
MB	7	360	288	1	264	4	3
Conv	1	288	2048	1	-	-	1

Table 16. ZenNet-900M-SE

block	kernel	in	out	stride	bottleneck	# layers
Conv	3	3	88	1	-	1
Btn	7	88	120	1	16	1
Btn	7	120	192	2	16	3
Btn	5	192	224	1	24	4
Btn	5	224	96	2	24	2
Btn	3	96	168	2	40	3
Btn	3	168	112	1	48	3
Conv	1	112	512	1	-	1

Table 17. ZenNet-1.0M for CIFAR-10/CIFAR-100

block	kernel	in	out	stride	bottleneck	# layers
Conv	3	3	32	1	-	1
Btn	5	32	120	1	40	1
Btn	5	120	176	2	32	3
Btn	7	176	272	1	24	3
Btn	3	272	176	1	56	3
Btn	3	176	176	1	64	4
Btn	5	176	216	2	40	2
Btn	3	216	72	2	56	2
Conv	1	72	512	1	-	1

Table 18. ZenNet-2.0M for CIFAR-10/CIFAR-100