This is the supplementary material for paper "EZCrop: Energy-Zoned Channels for Robust Output Pruning". It covers the following contents:

- The detailed proof of Theroem 1. (Appendix 1)
- Additional 3D plots to visualize the convolution process and the energy distribution of the feature maps in the frequency domain. (Appendix 2.1)
- The details of how to define and draw the circle energy zone in the frequency domain. (Appendix 2.2)
- Discussion on the effectiveness of rank-based and energy-based metric. (Appendix 3)
- Simple examples to illustrate how to find the square energy zone when the given feature map is not of a regular shape. (Appendix 4.1)
- The pseudocode of EZCrop. (Appendix 4.2)
- Training details of the experiments. (Appendix 5)
- Additional experimental results. (Appendix 6)
- Run the codes. (Appendix 7)

# 1. Convolution in the Frequency Domain

**Theorem 1.** *For a single 3-D input $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{S\times H\times W}$ and a CONV layer with a 4-D kernel tensor $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{D\times D\times S\times T}$, their convolution result $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{T\times H\times W}$ can be formalized as:*

$$\boldsymbol{\mathcal{Y}}[j,:,:] = \sum_{i=1}^{S} \mathcal{F}^{-1}(\mathcal{F}(\hat{\boldsymbol{\mathcal{K}}}[:,:,i,j]) \odot \mathcal{F}(\boldsymbol{\mathcal{X}}[i,:,:])), \quad (1)$$

*where $\odot$ stands for the point-wise multiplication (also called Hadamard product), and $\hat{\boldsymbol{\mathcal{K}}} \in \mathbb{R}^{H\times W\times S\times T}$ is the expanded filter, whose slices $\hat{\boldsymbol{\mathcal{K}}}[:,:,i,j]$ are all doubly circulant matrices generated by the torus form of $\boldsymbol{\mathcal{K}}[:,:,i,j]$.*

*Proof.* Figure 1 demonstrates the relations among equations, which helps understand the proof process.

**Circulant Matirx and its Decomposition** We define $F \in \mathbb{C}^{n\times n}$ to be the Discrete Fourier Transform (DFT) matrix with $F_{ij} = \omega^{ij}$ ($w = e^{-\frac{2\pi i}{n}}$ with $i^2 = -1$, this i appears here only once and is not to be confused with the index $i$ in other places),

$$F = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)(n-1)} \end{pmatrix}, \quad (2)$$

and we use $[n]$ to dente the set $0, 1, \ldots, n-1$.

To ease illustration, we normalize the columns of $F$ to unit length by defining the unitary matrix $Q = \frac{1}{\sqrt{n}}F$ such that $QQ^* = Q^*Q = I$ with $I \in \mathbb{R}^{n\times n}$ being the identity matrix. A circulant matrix $A \in \mathbb{R}^{n\times n}$ is fully characterized by its first column (row), followed by successive bottom-up (right-left) cyclic shifts. Specifically, let

$$A = \begin{pmatrix} a_0 & a_{n-1} & \cdots & a_1 \\ a_1 & a_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1} \\ a_{n-1} & \cdots & a_1 & a_0 \end{pmatrix}, \quad (3)$$

$$C = \begin{pmatrix} 0 & 0 & \cdots & 1 \\ 1 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 1 & 0 \end{pmatrix}, \quad (4)$$

then with $A[:,j]$ denoting the $j$th column of $A$, and noting we start counting from 0, we have

$$A[:,j] = C^j A[:,0] \text{ for } j \in [n]. \quad (5)$$

It is well known that $A$ has a fixed set of eigenvectors (up to scaling) solely determined by its circulant structure, namely,

$$AQ^* = Q^*\Lambda \text{ where } \Lambda = \mathrm{diag}(FA[:,0]). \quad (6)$$

By scaling both sides with $\sqrt{n}$ we get $AF^* = F^*\Lambda$ though we will stick to Eq. 6 for consistency. This implies

$$A = Q^*\Lambda Q = \bar{Q}\Lambda Q \text{ and } \Lambda = QAQ^* = QA\bar{Q}. \quad (7)$$

Using tensor mode product notation [2], Eq. 7 can be expressed as $A = \Lambda \times_1 Q^* \times_2 Q^T = \Lambda \times_1 Q^* \times_2 Q$ since $Q$ is symmetric, and similarly, $\Lambda = A \times_1 Q \times_2 Q^*$. Such eigen-decomposition of a circulant $A$ can be used to speed up the matrix-vector product $y = Ax$, where $x, y \in \mathbb{R}^n$, via

$$y = Ax = Q^*\Lambda Qx = \frac{1}{n}F\mathrm{diag}(FA[:,0])Fx. \quad (8)$$

Using the definitions $\mathcal{F}_1(v) := Fv$ and $\mathcal{F}_1^{-1}(v) := \frac{1}{n}\bar{F}v$, and that $\mathrm{diag}(u)v = u \odot v$, Eq. 8 can be computed by

$$y = \mathcal{F}_1^{-1}(\mathcal{F}_1(A[:,0]) \odot \mathcal{F}_1(x)). \quad (9)$$

**Single-Channel Convolution** Circulant matrices arise naturally in the analytical derivations of convolutional neural network (CNN) operations. We use a toy example with one input channel and one output channel for easy illustration. Here $X, Y \in \mathbb{R}^{4\times 4}$, $W \in \mathbb{R}^{3\times 3}$ are the input channel,

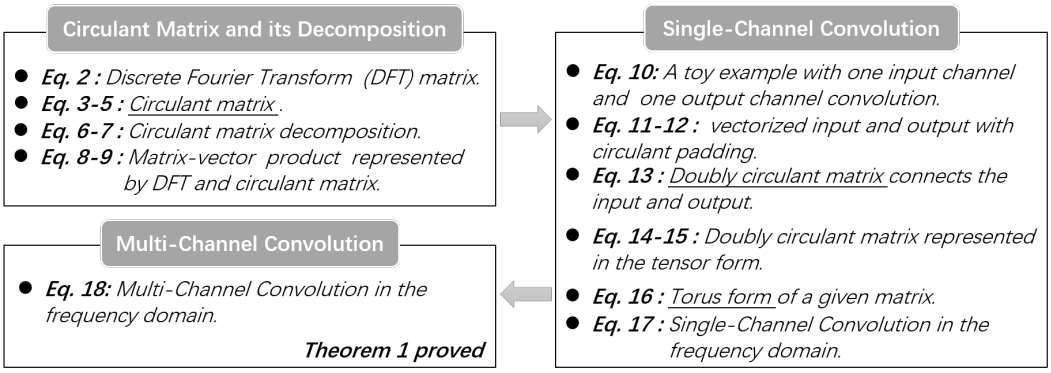



Figure 1. The road map about how we prove Theorem 1.

output channel and kernel filter, respectively.

$$\begin{pmatrix} y_{00} & y_{01} & y_{02} & y_{03} \\ y_{10} & y_{11} & y_{12} & y_{13} \\ y_{20} & y_{21} & y_{22} & y_{23} \\ y_{30} & y_{31} & y_{32} & y_{33} \end{pmatrix} = \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{pmatrix} * \begin{pmatrix} w_{00} & w_{01} & w_{02} \\ w_{10} & w_{11} & w_{12} \\ w_{20} & w_{21} & w_{22} \end{pmatrix}. \tag{10}$$

We assume circulant padding, namely, surrounding the input $X$ with eight $X$'s around it. It is obvious that one-pixel expansion (padding) around the four sides of $X$ is implied to equate the input-output dimensions. Now, if we vectorize $X$ and $Y$ and write out the the linear mapping relating them, we get

$$\mathrm{vec}(Y) = (y_{00}, y_{10}, y_{20}, y_{30} | \dots | y_{03}, y_{13}, y_{23}, y_{33}), \tag{11}$$

$$\mathrm{vec}(X) = (x_{00}, x_{10}, x_{20}, x_{30} | \dots | x_{03}, x_{13}, x_{23}, x_{33}), \tag{12}$$

$$\mathrm{vec}(Y) = B\mathrm{vec}(X), \tag{13}$$

where $B$ is a doubly circulant matrix as shown in Figure 2, and represents the flattened filtering matrix arising from W.

Next, we tensorize $B$ into a 4-way tensor $\boldsymbol{\mathcal{B}}$ whose modes are marked by the axes $i_1$ to $i_4$ as in Figure 2. It can be seen that the dimension of each axis is 4 and $i_1, i_2, i_3, i_4 \in [4]$. The doubly circulant property refers to $1, 2$-circulant and $3, 4$-circulant. Similar to Eq. 5, it can be easily verified that the cyclic shift holds true for each circulant mode pair

$$\boldsymbol{\mathcal{B}}[:, i_2, :, i_4] = \boldsymbol{\mathcal{B}}[:, 0, :, 0] \times_1 C^{i_2} \times_3 C^{i_4} \text{ for } i_2, i_4 \in [4], \tag{14}$$

where the shifting matrix $C$ is assumed appropriately sized. Moreover, using results from [6], a totally diagonal 4-way tensor $\Omega$ can be obtained from

$$\Omega = \boldsymbol{\mathcal{B}} \times_1 Q \times_2 Q^* \times_3 Q \times_4 Q^*, \tag{15}$$

which in turn suggests $\boldsymbol{\mathcal{B}} = \Omega \times_1 Q^* \times_2 Q \times_3 Q^* \times_4 Q$.

We now focus on $\boldsymbol{\mathcal{B}}[:, 0, :, 0]$ which consists of entries in the first column in B and forms a 2-way tensor (viz. a matrix) along $i_1$ and $i_3$. We denote this matrix by $\hat{W}$ that spans the $i_1$ and $i_3$ axes of $\boldsymbol{\mathcal{B}}$, namely,

$$\hat{W} = \boldsymbol{\mathcal{B}}[:, 0, :, 0] = \begin{pmatrix} w_{11} & w_{10} & 0 & w_{12} \\ w_{01} & w_{00} & 0 & w_{02} \\ 0 & 0 & 0 & 0 \\ w_{21} & w_{20} & 0 & w_{22} \end{pmatrix}, \tag{16}$$

and $\hat{W}$ is called torus form of $W$.

Analogous to Eq. 6, the diagonal entries in $\Omega$ can be found from $\boldsymbol{\mathcal{B}}[:, 0, :, 0] \times_1 F \times_3 F = F\hat{W}F^T$ which is exactly the 2D-FFT of $\hat{W}$. In particular, the diagonal entries in $\Omega$ can also be regarded as the diagonal entries after diagonalizing the matrix $B$ in Figure 2, which are the 16 entries in $\mathrm{vec}(F\hat{W}F)$. By unvectorizing $\mathrm{vec}(Y)$ and $\mathrm{vec}(X)$ in Eq. 11 and 12 back to matrices, we have the 2D counterpart to Eq. 9

$$Y = \mathcal{F}^{-1}(\mathcal{F}(\hat{W}) \odot \mathcal{F}(X)). \tag{17}$$

An intuitive view of Eq. 17 is that a multiplicative mask $\mathcal{F}(\hat{W})$ is applied to the 2D FFT of the input X.

**Multi-Channel Convolution** When given a single 3-D input $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{S \times H \times W}$ and a CONV layer with a 4-D kernel tensor $\boldsymbol{\mathcal{K}} \in \mathbb{R}^{D \times D \times S \times T}$, each slice of $\boldsymbol{\mathcal{X}}$, namely, $\boldsymbol{\mathcal{X}}[i, :, :]$ is equivalent to the $X$ in Eq. 17. Besides, the torus form of each slice $\boldsymbol{\mathcal{K}}[:, :, i, j]$ in $\boldsymbol{\mathcal{K}}$ is equal to the $\hat{W}$ in Eq. 17, denoted as $\hat{\boldsymbol{\mathcal{K}}}[:, :, i, j]$. Figure 3 shows how $\boldsymbol{\mathcal{X}}$ does convolution with a single filter $\boldsymbol{\mathcal{K}}[:, :, :, j]$ in the frequency domain. The process can be formalized as

$$\boldsymbol{\mathcal{Y}}[j, :, :] = \sum_{i=1}^{S} \mathcal{F}^{-1}(\mathcal{F}(\hat{\boldsymbol{\mathcal{K}}}[:, :, i, j]) \odot \mathcal{F}(\boldsymbol{\mathcal{X}}[i, :, :])). \tag{18}$$

Theorem 1 is proved. □

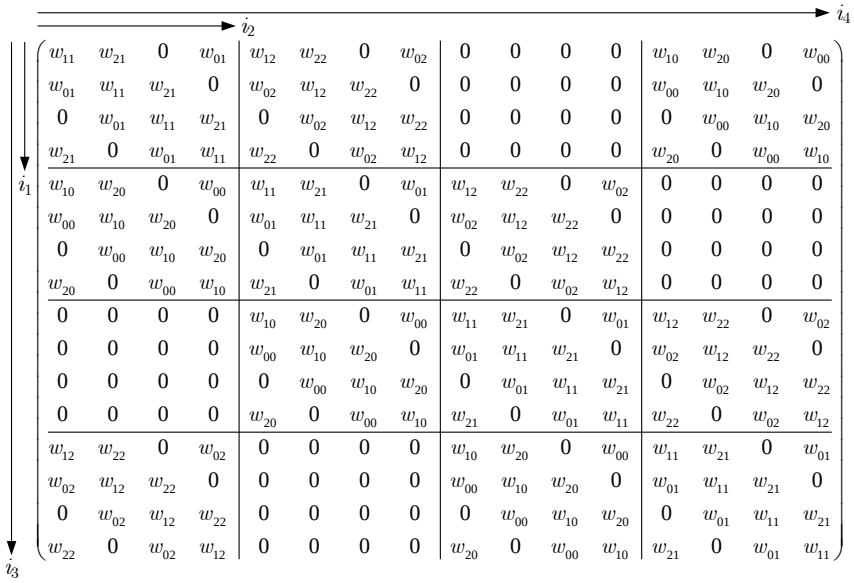



Figure 2. Matrix $B$ and its tensorization into the 4-way tensor $\mathcal{B}$. The index pair $\{i_1, i_2\}$ addresses entries within each block, whereas $\{i_3, i_4\}$ indexes the blocks.

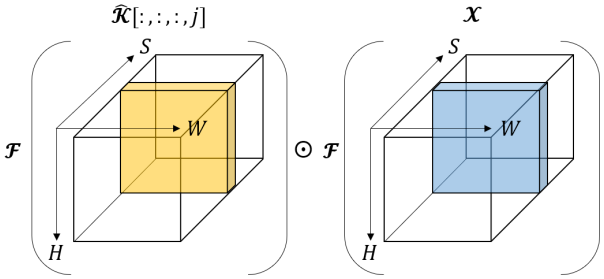



Figure 3. Multi-Channel Convolution in the frequency domain. $\hat{\mathcal{K}}$ is the expanded version of $\mathcal{K}$. By adding up all the convolution results between the two corresponding matrices along the S-axis, we get one slices $\mathcal{Y}[j,:,:]$ of the output.

# 2. Constant Ranks in Feature Map: A Frequency-Domain Perspective

## 2.1. Visualize Convolution in the Frequency Domian

In the main paper, we visualize the convolution in the frequency domain for VGGNet, a neural network with a plain structure. In this section, we show more visualization results for neural networks with special structures: 1) ResNet-56 with residual blocks (Figure 4), and 2) DenseNet-40 with inception modules (Figure 5). It can be observed that the relations between rank and energy distribution are similar in the three kinds of neural networks. The filter with a large corresponding rank is expected to generate feature maps with dispersed energy distribution in the frequency domain. The consistency of the observations for VGGNet, ResNet-56, and DenseNet-40 show the ability of EZCrop to be applied to different network structures.

## 2.2. Energy in the Circle

In this section, we give the details of how we draw the circles. Figure 6 shows an example using feature map of size $8 \times 8$, and displays three circles with radii $1, 2$ and $3$, respectively. In short, we start from the DC center and expand the energy-zone outward in a diamond shape. The distance from the center of the DC center to the diamond's vertex is defined as the radius. The ratio of the sum of the elements in the diamond area to the sum of all the feature map elements is fixed, and we set the ratio in the main paper as $70\%$. In fact, we use an energy-zone of diamond shape, and the circles are employed for better visualization. When fixing the ratio, the process to find the radii of the circles can be formalized as below:

$$\min_{R_i^j[b]} \frac{S(R_i^j[b])}{S(\boldsymbol{E}_i^j[b,:,:])} \geq \alpha, R_i^j = \frac{1}{B}\sum_{b=1}^{B} R_i^j[b], \tag{19a}$$

$$\boldsymbol{E}_i^j[b,:,:] = \text{abs}(\text{fftshift}(\mathcal{F}(\boldsymbol{Y}_i^j[b,:,:]))), \tag{19b}$$

where $R_i^j[b]$ is the radius of the circle for the $b$-th sample in a batch, $R_i^j$ is the average radius of the whole batch, $S(R_i^j[b])$ is the sum of the elements in the area covered by DC as the center and $R_i^j[b]$ as the radius in $\boldsymbol{E}_i^j[b,:,:]$, $S(\boldsymbol{E}_i^j[b,:,:])$ is the sum of all elements in $\boldsymbol{E}_i^j[b,:,:]$, and $\alpha$ is a hyper-parameter between 0 and 1.

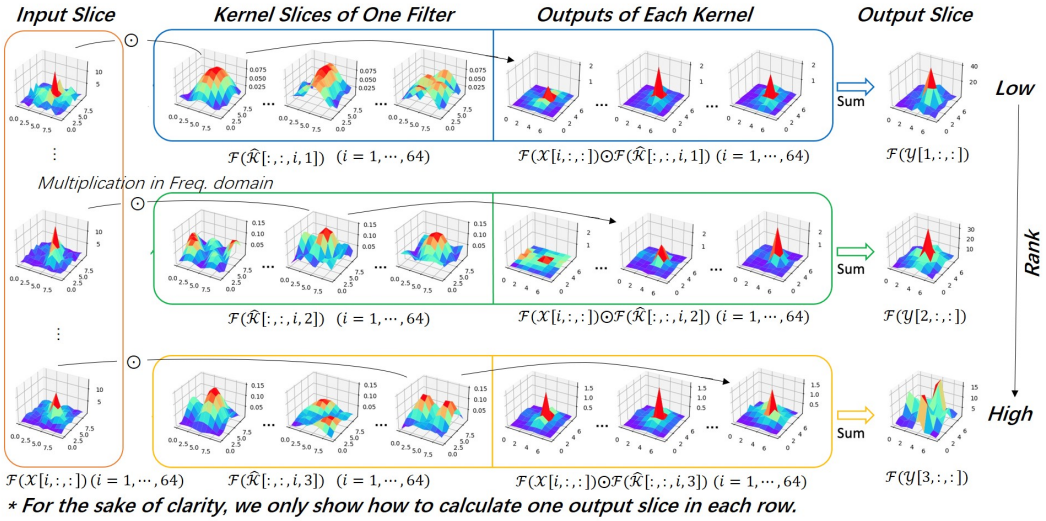



Figure 4. The visualization results of convolution in the frequency domain for ResNet-56. The three filters are selected from layer3.1.conv1, the number of input and output channels are both 64.

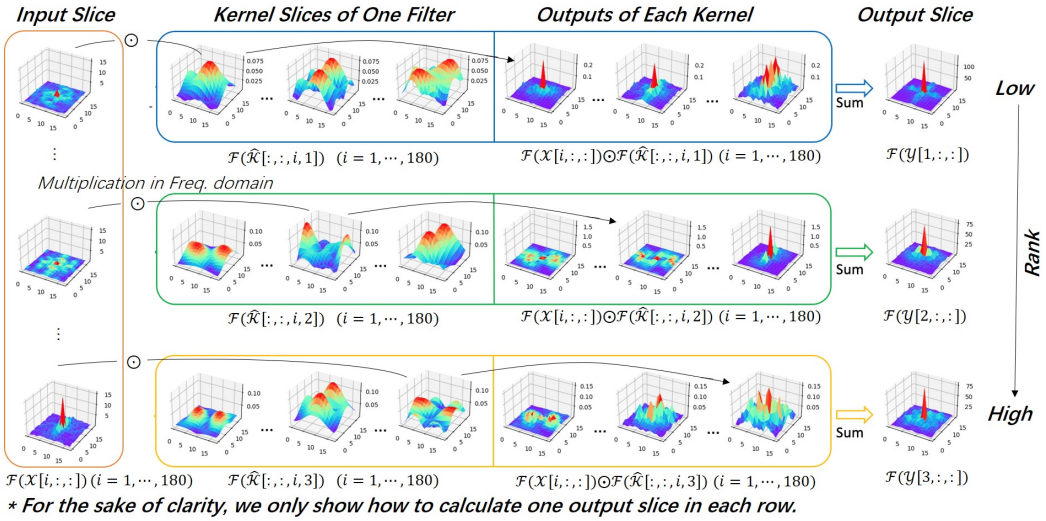


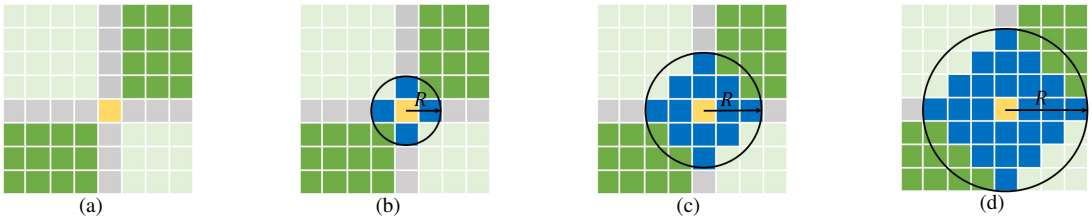

Figure 5. The visualization results of convolution in the frequency domain for DenseNet-40. The three filters are selected from dense2.1.conv1, the number of input and output channels are 180 and 12, respectively. For the last column, we can observe the concentration of energy through the coverage of the blue area near the red cone. From top to bottom, the distribution of the blue area changes from concentrated to scattered, reflecting the concentration of energy from high to low.

## 3. Effectiveness of Rank-based and Energy-based Metrics

In this section, we compare effectiveness of rank-based and energy-based metrics to show why EZCrop can give more accurate filter importance evaluation and prune the given model with higher robustness.

**Resolution.** Compared with the rank-based metric (HRank), the energy-based metric (EZCrop) has a higher resolution. More specifically, HRank evaluates the importance of the filters by computing the average rank of several batches of slices. However, the rank of each feature map slice is only an integer in $[0, \min(H_i, W_i)]$. While in EZCrop, the information in the whole slice $H_i \times W_i$ has been considered. A higher precision decimal can represent the energy-ratio of each slice.

**Feature Map Extraction.** In HRank, the feature maps must be extracted after the operations like ReLU, batch normalization, max-pooling, etc. These operations increase the sparsity of the feature map slices. Otherwise, the feature map's rank will always tend to be full, which makes it difficult to distinguish the importance of filters. However,




(a) (b) (c) (d)

Figure 6. We take a feature map of size $8 \times 8$ as an example to illustrate how to decide the radius of the circle. In the first subplot, The dark and light green show the conjugate symmetry property after $\mathcal{F}(\cdot)$ and $\mathrm{fftshift}(\cdot)$ operations, and the yellow square represents the DC component. Subplots (b)-(c) illustrate how to define the radius, and the blue squares are the elements will be summed up in the circle. The radii in (b)-(c) are $1, 2$ and $3$, respectively.

EZCrop has broader applicability, which does not rely on the feature map slices' sparsity. In other words, EZCrop is a more general filter evaluation method that can be applied on feature maps extracted from any position of the neural network.

**Richness of Information.** We use $Y$ to denote a single feature map slice, and $\hat{Y}$ to denote the feature map mapped to the frequency domain, and $F$ is the DFT matrix defined in Eq. 2, then $Y$ can be represented as below:

$$Y = F(\hat{Y} \odot E)F + F(\hat{Y} \odot \bar{E})F, \tag{20}$$

where $E$ is a mask having the same size as the feature map, and the elements in the selected area are all 1, while other elements are all 0. We denote the opposite mask of $E$ as $\bar{E}$. The first term of Eq. 20 can be regarded as a set of low-frequency components. The smaller the energy-zone ratio, the smaller the frequency diversity in the set of low-frequency components. In other words, the larger the energy-zone ratio, the first term contains richer the information of various frequency components.

# 4. EZCrop

## 4.1. Feature Maps of Different Sizes

There are some scenarios the feature maps are not square $(H_i \neq W_i)$, Figure 7 illustrates how EZCrop finds the DC center and the energy zone under the exceptional cases. Obviously, EZCrop can be applied to feature maps with any shape without loss of generality.

## 4.2. Algorithm

Algorithm 1 describes the workflow how EZCrop evaluates the importance of the filters in a given CONV layer. It is worth noting that we use the average energy-zone ratio of several batches instead of the ratio of a single feature map to evaluate the filters' importance.

# 5. Training Details

For all networks in the main paper and Appendix, we use 5 batches of samples to compute the pruning metrics. There are 128 samples in each batch. Excluding the ablation study to explore the relations between the pruning performance and $\beta$, we use $\beta = 0.25$ in all our experiments. To make all comparative experiments between EZCrop and HRank fair, we follow the training settings of HRank, which can be found in the official GitHub repository of [4].

| | HRrank [4] | **EZCrop** |
|---|---|---|
| ResNet-110 | 1863.45s | **774.25**s (58.45%) |
| GoogLeNet | 3294.31s | **670.12**s (79.66%) |

Table 1. Runtimes for metric computation in HRank [4] and EZCrop on CIFAR-10/ImageNet(bottom row) nets.

# 6. Additional Experimental Results

Due to space limitations, we put the experimental results of ResNet-110 / GoogleNet on CIFAR-10 here. The setting for pruning metric generation are the same in the main paper, namely, $\beta = 0.25$. Besides, we set the training settings for each comparative experiments between HRank and EZCrop under different compression rate the same too.

**Time Comparison.** Table 1 shows the actual runtimes of EZCrop and HRank to generate the required pruning metrics for ResNet-110 / GoogleNet on one GTX1080Ti GPU. For ResNet-110, EZCrop reduces the required time by more than half, nearly $60\%$. The time reduction for GoogleNet is more impressive, reaching about $80\%$.

**ResNet-110.** Table 2 displays the results of ResNet-110 on CIFAR-10. Compared with L1 and GAL-0.5, EZCrop has advantages in all aspects. There are three sets of comparative experiments between HRank and EZCrop. However, we find the difference between EZCrop and HRank's Top-1 accuracy is not as obvious as in ResNet-56. When the parameter reduction is $39.1\%$ and $68.3\%$, their accuracies are exactly the same. When the parameters are reduced by $58.1\%$, EZCrop shows a $0.05\%$ advantage, which is insignificant. According to the results of ResNet-56/110, EZCrop has more significant advantages for shallower networks containing residual blocks, and offers a more precise evaluation of filters' importance.

**GoogLeNet.** Table 3 shows the pruning results of

(a) (b) (c) (d)

Figure 7. (a) Square feature map slice with even height and width. (b) Square feature map slice with odd height and width. (c) Rectangular feature map slice with odd height and even width. (d) Rectangular feature map slice with even height and odd width.

**Algorithm 1** Using EZCrop to evaluate the importance of the filters in the $i$-th CONV layer.

**Input:** $M$ different batches of the output of the $i$-th CONV layer $\boldsymbol{\mathcal{Y}}_i^m \in \mathbb{R}^{B\times T_i\times H_i\times W_i}$.

**Output:** A set of filter indexes $I_i = \{I_i^1, \cdots, I_i^{T_i}\}$, whose elements are sorted according to the importance of the filters from significant to trivial.

**Step 1: Find the Square Center.**
**if** $H_i$ is even **then**
  $x_i = \frac{H_i}{2} + 1$
**else**
  $x_i = \frac{H_i+1}{2}$
**end if**
**if** $W_i$ is even **then**
  $y_i = \frac{W_i}{2} + 1$
**else**
  $y_i = \frac{W_i+1}{2}$
**end if**
**Step 2: Decide the Expanding Distance.**
$l_{ih} = H_i - x_i$ , $l_{iw} = W_i - y_i$
**if** $x_i - 1 = 0$ **or** $y_i - 1 = 0$ **then**
  $d_i = 0$
**else**
  $d_i = \text{ceil}(\beta \cdot \min(l_{ih}, l_{iw}))$
**end if**
**Step 3: Calculate the Energy Zone Rate.**
**for** $m = 1, 2, \cdots, M$ **do**
  **for** $j = 1, 2, \cdots, T_i$ **do**
    $\eta_i^j[m] = \frac{1}{B} \cdot \left(1 - \sum_{b=1}^{B} \frac{S_m(d_i[b])}{S_m(\boldsymbol{E}_i^j[b,:,:])}\right)$
  **end for**
**end for**
$\eta_i^j = \frac{1}{m} \cdot \sum_{m=1}^{M} \eta_i^j[m]$
**Step 4: Sort the Filters.**
$I_i = \{I_i^1, \cdots, I_i^{T_i}\}$ where $\eta_i^{I_i^m} \leq \eta_i^{I_i^n}$ for $m \geq n$.

GoogLeNet on CIFAR-10. Compared with the upper part approaches, EZCrop outperforms all of them in accuracy, FLOPs, and Params reduction. There are two sets of comparative experiments between EZCrop and HRank. For the first set, the percentage of parameter reduction is $53.5\%$, and the difference in Top-1 accuracy between the two approaches is $0.03\%$. For the second set, the gap between their Top-1 accuracy is $0.02\%$ when the parameters are reduced by $65.8\%$. The performances of EZCrop and HRank are very close here, showing that both schemes have a highly consistent choice of filters for pruning networks with inception modules.

| Model | Top-1% | FLOPs | Params |
|---|---|---|---|
| ResNet-110 | 93.50 | 252.89M(0.0%) | 1.72M(0.0%) |
| L1 [3] | 93.30 | 155.00M(38.7%) | 1.16M(32.6%) |
| GAL-0.5 [5] | 92.55 | 130.20M(48.5%) | 0.95M(44.8%) |
| HRank [4] | 94.20 | 140.54M(44.4%) | 1.04M(39.1%) |
| **EZCrop** | **94.20** | 140.54M(44.4%) | 1.04M(39.1%) |
| HRank [4] | 93.81 | 101.97M(59.6%) | 0.72M(58.1%) |
| **EZCrop** | **93.86** | 101.97M(59.6%) | 0.72M(58.1%) |
| HRank [4] | 93.23 | 71.69M(71.6%) | 0.54M(68.3%) |
| **EZCrop** | **93.23** | 71.69M(71.6%) | 0.54M(68.3%) |

Table 2. Pruning results of ResNet-110 on CIFAR-10.

| Model | Top-1% | FLOPs | Params |
|---|---|---|---|
| GoogLeNet | 95.05 | 1.52B(0.0%) | 6.15M(0.0%) |
| Random | 94.54 | 0.96B(36.8%) | 3.58M(41.8%) |
| L1 [3] | 94.54 | 1.02B(32.9%) | 3.51M(42.9%) |
| GAL-0.05 [5] | 93.93 | 0.94B(38.2%) | 3.12M(49.3%) |
| GAL-ApoZ [1] | 92.11 | 0.76B(50%) | 2.85M(53.7%) |
| HRank [4] | 95.04 | 0.65B(57.2%) | 2.85M(53.5%) |
| **EZCrop** | **95.07** | 0.65B(57.2%) | 2.85M(53.5%) |
| HRank [4] | 94.82 | 0.40B(73.9%) | 2.09M(65.8%) |
| **EZCrop** | **94.84** | 0.40B(73.9%) | 2.09M(65.8%) |

Table 3. Pruning results of GoogLeNet on CIFAR-10.

**Repetitive Pruning for VGGNet on CIFAR-10** For a fixed compression rate, we compare the accuracy of 1-pass and multi-pass pruning when reaching the required model size. For fairness, all settings of HRank and EZCrop for every single-pass are the same. Table 4 indicates that for a fixed compression rate, the robustness of EZCrop is obvious. However, due to the accumulation of small inaccuracies, the 1-pass accuracy of HRank is better than that of multi-pass.

## 7. Run the Codes

We have submitted the codes in the supplementary material as well. To use EZCrop, the users are expected to generate the ratios first, then do the pruning and retrain the pruned model based on the evaluation results. There

| #Passes (#epochs) | FLOPs | Params | HRank [4] | **EZCrop** |
|---|---|---|---|---|
| 1 (150) | $207.60M$ | $9.15M$ | 93.76% | **94.12%** |
| 2 (150) | $125.68M$ | $4.81M$ | 93.47% | **93.78%** |
| 3 (150) | $66.95M$ | $1.90M$ | 93.02% | **93.18%** |
| 1 (450) | $66.95M$ | $1.90M$ | 93.10% | 93.11% |

Table 4. Repetitive pruning of VGGNet on CIFAR-10.

is a folder called *script*, the users can modify the related arguments and run EZCrop easily by the *.sh* files. In the following, we give an example to illustrate how to run the codes. To generate the ratio:

```
python ratio_generation.py \
–alpha 0.25 \
–data_dir ./data \
–dataset cifar10 \
–arch vgg_16_bn \
–pretrain_dir ./vgg_16_bn.pth \
–gpu 0,1
```

According to the importance evaluation results, we can discard the trivial filters and retrain the pruned model:

```
python ratio_generation.py \
–data_dir ./data \
–arch vgg_16_bn \
–pretrain_dir ./vgg_16_bn.pth \
–ratio_conv_prefix ./ratio_conv/vgg_16_bn_limit5 \
–compress_rate [0.45]*7+[0.78]*5 \
–gpu 0,1
```

After pruning and training, there will be a *log* file, which records the whole process.

## References


[1] Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016.

[2] T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[3] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[4] Mingbao Lin, Rongrong Ji, Yan Wang, Yichen Zhang, Baochang Zhang, Yonghong Tian, and Ling Shao. Hrank: Filter pruning using high-rank feature map. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1529–1538, 2020.

[5] Shaohui Lin, Rongrong Ji, Chenqian Yan, Baochang Zhang, Liujuan Cao, Qixiang Ye, Feiyue Huang, and David Doermann. Towards optimal structured cnn pruning via generative adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2799, 2019.

[6] Mansoor Rezgi and Lars Eldén. Diagonalization of tensors with circulant structure. *Linear Algebra and its Applications*, 303:422–447, 08 2011.