# CHIP: CHannel Independence-based Pruning for Compact Neural Networks

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

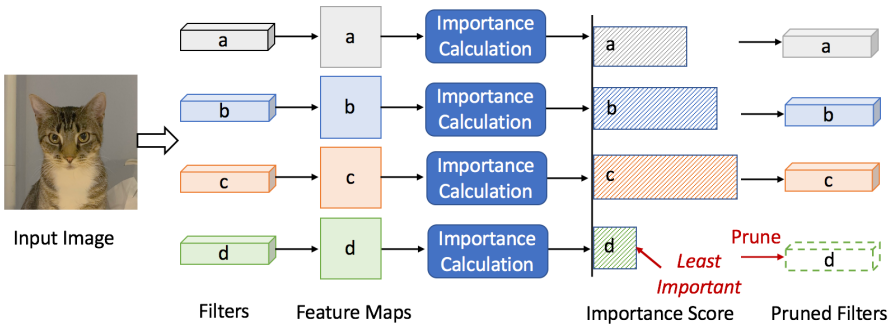
Filter pruning has been widely used for neural network compression because of its enabled practical acceleration. To date, most of the existing filter pruning works explore the importance of filters via using intra-channel information. In this paper, starting from an inter-channel perspective, we propose to perform efficient filter pruning using *channel independence*, a metric measures the correlations among different feature maps. The less independent feature map is interpreted as containing less useful information/knowledge, and hence its corresponding filter can be pruned without affecting model capacity. We systematically investigate the quantification metric, measuring scheme and sensitiveness/reliability of channel independence in the context of filter pruning. Our evaluation results for different models on various datasets show the superior performance of our approach. Notably, on CIFAR-10 dataset our solution can bring $0.75\%$ and $0.94\%$ accuracy increase over baseline ResNet-56 and ResNet-110 models, respectively, and meanwhile the model size and FLOPs are reduced by $42.8\%$ and $47.4\%$ (for ResNet-56) and $48.3\%$ and $52.1\%$ (for ResNet-110), respectively. On ImageNet dataset, our approach can achieve $40.8\%$ and $44.8\%$ storage and computation reductions, respectively, with $0.15\%$ accuracy increase over baseline ResNet-50 model.

## 1 Introduction

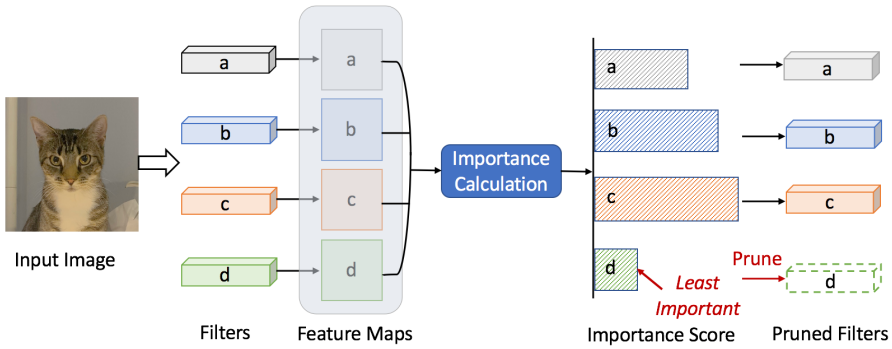
Convolutional neural networks (CNNs) have obtained widespread adoptions in numerous important AI applications [8, 29, 28, 4, 3, 27, 20]. However, CNNs are inherently computation intensive and storage intensive, thereby posing severe challenges for their efficient deployment on resource-constrained embedded platforms. To address these challenges, *model compression* is widely used to accelerate and compress CNN models on the edge devices. To date, various types of compression strategies, such as network pruning [6, 7, 22, 35, 16, 5], bit-precision reduction [7, 32, 26], low-rank approximation [33, 24] and knowledge distillation [12, 25], have been proposed and explored. Among them, *network pruning* is the most popular and extensively studied model compression technique in both academia and industry.

Based on their differences on pruning granularity, the state-of-the-art network pruning approaches can be roughly categorized to *weight pruning* [7, 6] and *filter pruning* [31, 15, 11]. Weight pruning focuses on the proper selection of the to-be-pruned weights within the filters. Although enabling high compression ratio, this strategy meanwhile causes unstructured sparsity patterns, which are not well supported by the general-purpose hardware in practice. On the other hand, filter pruning emphasizes the removal of the entire selected filters. The resulting structured sparsity patterns can be then properly leveraged by the off-the-shelf CPUs/GPUs to achieve acceleration in the real-world scenario.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.




(a) Feature information-based filter pruning from intra-channel perspective.




(b) Feature information-based filter pruning from inter-channel perspective.

Figure 1: Intra-channel vs Inter-channel perspectives for filter pruning.

**Existing Filter Pruning Methods.** Motivated by the potential practical speedup offered by filter pruning, to date numerous research efforts have been conducted to study *how to determine the important filters* – the key component of efficient filter pruning. A well-known strategy is to utilize the norms of different filters to evaluate their importance. Such "smaller-norm-less-important" hypothesis is adopted in several pioneering filter pruning works [15, 9]. Later, considering the limitations of norm-based criterion in real scenarios, [10] proposes to utilize geometric median-based criterion. More recently, first determining those important feature maps and then preserving the corresponding filters, instead of directly selecting the filters, become a popular strategy for filter pruning. As indicated in [18, 30], the features, by their natures, reflect and capture rich and important information and characteristics of both input data and filters, and hence measuring the importance of features can provide better guideline to determine the important filters. Built on this new pruning philosophy, several feature-guided filter pruning approaches [18, 30] have been proposed and developed, and the evaluation results show their superior performance over the state-of-the-art filter-guided counterparts with respect to both task performance (e.g., accuracy) and compression performance (e.g., model size and floating-point operations per second (FLOPS) reductions).

**Determining Importance: Intra-channel & Inter-channel Perspectives.** These recent advancements on filter pruning indeed show the huge benefits of leveraging feature information to determine the importance of filters. To date, most of the feature-guided approaches measure the importance from the *intra-channel* perspective. In other words, no matter which importance metric is used, the importance of one feature map (and its corresponding filter), is measured only upon the information of this feature map in its own channel. On the other aspect, the *inter-channel perspective*, which essentially determines the importance via using cross-channel information, is little explored or studied.

**Benefits of Inter-channel Perspective.** In principle, the feature information across multiple channels, if leveraged properly, can potentially provide richer knowledge for filter pruning than the intra-channel information. Specifically, this is because: 1) the importance of one filter, if being solely determined by its corresponding feature map, may be sensitive to input data; while the cross-channel feature information can bring more stable and reliable measurement; and 2) consider the essential mission of

pruning is to remove the unnecessary redundancy, the inter-channel strategy can inherently better identify and capture the potential unnecessary correlations among different feature maps (and the corresponding filters), and thereby unlocking the new opportunity of achieving better task and compression performance.

**Technical Preview and Contributions.** Motivated by these promising potential benefits, in this paper we propose to explore and leverage the cross-channel feature information for efficient filter pruning. To be specific, we propose ***Channel Independence***, a cross-channel correlation-based metric to measure the importance of filters. Channel independence can be intuitively understood as the measurement of "replaceability": when the feature map of one filter is measured as exhibiting lower independence, it means this feature map tends to be more linearly dependent on other feature maps of other channels. In such scenario, the contained information of this low-independence feature map is believed to have already been implicitly encoded in other feature maps – in other words it does not contain useful information or knowledge. Therefore the corresponding filter, which outputs this low-independence feature map, is viewed as unimportant and can be safely removed without affecting the model capacity. Overall, the contributions of this paper are summarized as:

- We propose channel independence, a metric that measures the correlation of multiple feature maps, to determine the importance of filters. Built from an inter-channel perspective, channel independence can identify and capture the filter importance in a more global and precise way, thereby providing better guideline for filter pruning.
- We systematically investigate and analyze the suitable quantification metric, the complexity of measuring scheme and the sensitiveness & reliability of channel independence, and then we develop a low-cost fine-grained high-robustness channel independence calculation scheme for efficient filter pruning.
- We empirically apply the channel independence-based importance determination in different filter pruning tasks. The evaluation results show that our proposed approach brings very high pruning performance with preserving high accuracy. Notably, on CIFAR-10 dataset our solution can bring $0.75\%$ and $0.94\%$ accuracy increase over baseline ResNet-56 and ResNet-110 models, respectively, and meanwhile the model size and FLOPs are reduced by $42.8\%$ and $47.4\%$ (for ResNet-56) and $48.3\%$ and $52.1\%$ (for ResNet-110), respectively. On ImageNet dataset, our approach can achieve $40.8\%$ and $44.8\%$ storage and computation reductions, respectively, with $0.15\%$ accuracy increase over baseline ResNet-50 model.

## 2 Preliminaries

**Filter Pruning.** For a CNN model with $L$ layers, its $l$-th convolutional layer $\boldsymbol{\mathcal{W}}^l = \{\boldsymbol{\mathcal{F}}_1^l, \boldsymbol{\mathcal{F}}_2^l, \cdots, \boldsymbol{\mathcal{F}}_{c^l}^l\}$ contains $c^l$ filters $\mathcal{F}_i^l \in \mathbb{R}^{c^{l-1}\times k^l \times k^l}$, where $c^l$, $c^{l-1}$ and $k^l$ denote the number of output channels, the number of input channels and the kernel size, respectively. In general, network pruning can be formulated as the following optimization problem:

$$\min_{\{\boldsymbol{\mathcal{W}}^l\}_{l=1}^L} \quad \mathcal{L}(\boldsymbol{\mathcal{Y}}, f(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{W}}^l)), \text{s.t.} \quad \|\boldsymbol{\mathcal{W}}^l\|_0 \leq \kappa^l, \tag{1}$$

where $\mathcal{L}(\cdot,\cdot)$ is the loss function, $\boldsymbol{\mathcal{Y}}$ is the ground-truth labels, $\boldsymbol{\mathcal{X}}$ is the input data, and $f(\cdot,\cdot)$ is the output function of CNN model $\{\boldsymbol{\mathcal{W}}^l\}_{l=1}^L$. Besides, $\|\cdot\|_0$ is the $\ell_0$-norm that measures the number of non-zero elements in the set, and $\kappa^l$ is the number of filters to be preserved in the $l$-th layer.

**Feature-guided Filter Pruning.** Consider the feature maps, in principle, contain rich and important information of both filters and input data, approaches using feature information have become popular and achieved the state-of-the-art performance for filter pruning. To be specific, unlike the filter-guided methods that directly minimize the loss function involved with filters (as Eq. 1), the objective of feature-guided filter pruning is to minimize the following loss function:

$$\min_{\{\boldsymbol{\mathcal{A}}^l\}_{l=1}^L} \quad \mathcal{L}(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{A}}^l), \text{s.t.} \quad \|\boldsymbol{\mathcal{A}}^l\|_0 \leq \kappa^l, \tag{2}$$

where $\boldsymbol{\mathcal{A}}^l = \{\boldsymbol{A}_1^l, \boldsymbol{A}_2^l, \cdots, \boldsymbol{A}_{c^l}^l\} \in \mathbb{R}^{c^l \times h \times w}$ is a set of feature maps output from the $l$-th layer, and $\boldsymbol{A}_i^l \in \mathbb{R}^{h\times w}$ is the feature map corresponds to the $i$-th channel. In general, after the $\kappa^l$ important feature maps are identified and selected, their corresponding $\kappa^l$ filters are preserved after pruning.

# 3 The Proposed Method

## 3.1 Motivation

As formulated in Eq. 2, feature-guided filter pruning leverages the generated feature maps in each layer to identify the important filters. To achieve that, various types of feature information, such as the high ranks [18] and the scaling factors [30], have been proposed and utilized to select the proper feature maps and the corresponding filters. A common point for these state-of-the-art approaches is that all of them focus on measuring the importance via using the information contained in each feature map. On the other hand, the correlation among different feature maps, as another type of rich information provided by the neural networks, is little exploited in the existing filter pruning works.

**Why Inter-channel Perspective?** We argue that the feature information across multiple channels is of significant importance and richness, and it can be leveraged towards efficient filter pruning. Such inter-channel perspective is motivated by two promising benefits. First, filter pruning is essentially a data-driven strategy. When the importance of one filter solely depends on the information represented by its own generated feature map, the measurement of the importance may be unstable and sensitive to the slight change of input data. On the other hand, determining the importance built upon information contained in the multiple feature maps, if performed properly, can reduce the potential disturbance incurred by the change of input data, and thereby making the importance ranking more reliable and stable. Second, the inter-channel strategy, by its nature, can better model and capture the cross-channel correlation. In the context of model compression, these identified correlations can be interpreted as a type of architecture-level redundancy, which is exactly what filter pruning aims to remove. Therefore, inter-channel strategy can enable more aggressive pruning while still preserving high accuracy.

## 3.2 Channel Independence: A New Lens for Filter Importance

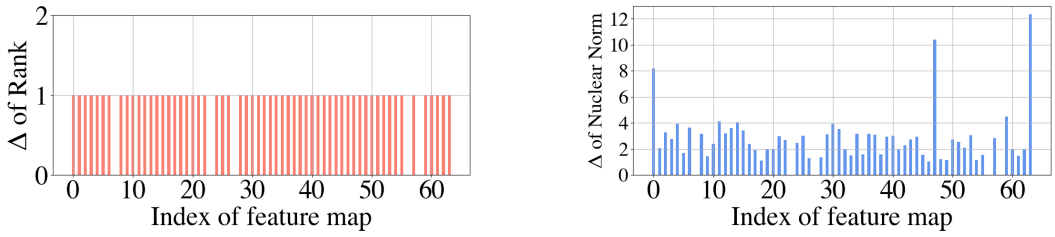
**Key Idea.** Motivated by these promising benefits, we propose to explore the filter importance from the inter-channel perspective. Our key idea is to use *channel independence* to represent the importance of each feature map (and its corresponding filter). To be specific, when one feature map of one channel is highly *linearly dependent* on other feature maps of other channels, it implies that its contained information has already been largely encoded in other feature maps. Consequently, even we remove the corresponding filter, the represented information & knowledge of its generated low-independence feature map can still be largely preserved and approximately reconstructed by other feature maps of other filters after fine-tuning procedure. In other words, the filters that generate low-independence feature maps tend to exhibit more "replaceability", which can be interpreted as lower importance. Therefore, removing those filters with low channel-independence feature maps will be safe while still preserving high model capacity.

**How to Measure Channel Independence?** Next we discuss how to properly measure the independence of one feature map from others. To that end, four important questions need to be answered.

**Question #1:** *Which mathematical metric should be adopted to quantify the independence of one feature map from other feature maps?*

Analysis. Considering the entire set of feature maps generated from one layer is a 3-D tensor, we propose to extract the linear dependence information of each feature map within the framework of linear algebra. To be specific, given output feature map set of the $l$-th layer $\boldsymbol{\mathcal{A}}^l$, we first matricize $\boldsymbol{\mathcal{A}}^l$ to $\boldsymbol{A}^l = [\boldsymbol{a}_1^{l\,T}, \boldsymbol{a}_2^{l\,T}, \cdots, \boldsymbol{a}_{c^l}^{l\,T}]^T \in \mathbb{R}^{c^l \times hw}$, where $\boldsymbol{a}_i^l \in \mathbb{R}^{hw}$ is the vectorized $\boldsymbol{A}_i^l$. In such scenario, the linear independence of each vectorized feature map $\boldsymbol{a}_i^l$, as a row of the matricized entire set of feature maps $\boldsymbol{A}^l$, can be measured via the existing matrix analysis tool. The most straightforward solution is to use *rank* to determine the independence of $\boldsymbol{a}_i^l$, since rank mathematically represents the maximum number of linearly independent rows/columns of the matrix. For instance, we can remove one row from the matrix, and calculate the rank change of the matrix, and then identify the impact and the importance of the deleted row – the less rank change, the less independence (and the importance) of the removed row.

Our Proposal. However, in the context of filter pruning, we believe, the ***change of nuclear norm of the entire set of feature maps***, is a better metric to quantify the independence of each feature map. This is because, as the $\ell_1$-norm of singular values of the matrix, the nuclear norm can reveal richer "soft" information on the impact of the deleted row on the matrix; while the rank, as the $\ell_0$-norm




(a) The change of rank.

(b) The change of nuclear norm.

Figure 2: The **change** of (a) rank and (b) nuclear norm of the entire set of feature maps ($\boldsymbol{A}^l$) when one feature map ($\boldsymbol{a}_i^l$) is removed. The x-axis represents the index of feature map that is removed. The y-axis represents the the corresponding rank/nuclear norm change of the entire set of feature maps. The feature maps are output from one layer of ResNet-50 model with input as ImageNet image. **It is seen that change of nuclear norm can better reveal the impact of the deleted feature map on the entire set of feature maps.**

of the singular values, is too "hard" to reflect such change. For instance, as shown in Fig. 2, when we select $\boldsymbol{a}_i^l$, as one row of $\boldsymbol{A}^l$, to be removed, the rank change of $\boldsymbol{A}^l$ is almost the same regardless of our selection of $\boldsymbol{a}_i^l$; while the corresponding changes of nuclear norm vary significantly when different $\boldsymbol{a}_i^l$ are deleted. Therefore, the change of nuclear norm can be viewed as a more precise metric to measure the linear independence of one feature map in a more fine-grained way. In general, the channel independence of one feature map is defined and calculated as below:

**Definition 1 (Channel independence of single feature map)** *For the $i$-th layer with output feature maps $\boldsymbol{\mathcal{A}}^l = \{\boldsymbol{A}_1^l, \boldsymbol{A}_2^l, \cdots, \boldsymbol{A}_{c^l}^l\} \in \mathbb{R}^{c^l \times h \times w}$ , the Channel Independence (CI) of one feature map $\boldsymbol{A}_i^l \in \mathbb{R}^{h \times w}$ in the $i$-th channel is defined and calculated as:*

$$CI(\boldsymbol{A}_i^l) \triangleq \|\boldsymbol{A}^l\|_* - \|\boldsymbol{M}_i^l \odot \boldsymbol{A}^l\|_*, \tag{3}$$

where $\boldsymbol{A}^l \in \mathbb{R}^{c^l \times hw}$ is the matricized $\boldsymbol{\mathcal{A}}^l$, $\|\cdot\|_*$ is the nuclear norm, $\odot$ is the Hardmard product, and $\boldsymbol{M}_i^l \in \mathbb{R}^{c^l \times hw}$ is the *row mask matrix* whose $i$-th row entries are zeros and other entries are ones.

**Question #2:** *What is the proper scheme to quantify the independence of multiple feature maps?*

Analysis. Eq. 3 describes the measurement of channel independence for single feature map. However, in practice filter pruning typically aims to remove multiple filters, which means the independence of the combination of multiple feature maps needs to be calculated. In the case of pruning $m$ filters, such scenario corresponds to checking the changes of nuclear norm of the original $c^l$-row $\boldsymbol{A}^l$ after removing $m$ rows ($\boldsymbol{a}_i^l$). A straightforward solution is to just calculate $C_m^{c^l}$ changes of nuclear norms for all the possible $m$-row removal choices, and then select the one which corresponds to the smallest change. However, this strategy is very computationally expensive, and sometimes even intractable when $c^l$ is large. For instance, in order to identify the smallest nuclear norm change for pruning 50% filters of a 256-output channel ResNet-50 layer, such brutal-force measurement requires more than $5 \times 10^{75}$ times of nuclear norm calculation.

Our Proposal. To address this computational challenge, we propose to leverage the independence of individual feature map to *approximate* the independence of their combination. To be specific, in order to determine $m$ least independent rows in the $\boldsymbol{A}^l$, we first iteratively remove one row ($\boldsymbol{a}_i^l$) from $\boldsymbol{A}^l$ and calculate the corresponding nuclear norm change between the remaining $(c^l - 1)$-row matrix and the original $c^l$-row $\boldsymbol{A}^l$. Then, among the $c^l$ calculated changes, we identify the $m$ smallest ones and the corresponding removed $\boldsymbol{a}_i^l$. Those selected $m$ vectorized feature maps $\boldsymbol{a}_i^l$ are interpreted as the less independent from other feature maps, and hence their corresponding filters $\boldsymbol{\mathcal{F}}_i^l$ are less important ones that should be pruned. In general, this individual independence-based measurement can closely approximate the combined independence of multiple feature maps (see **Definition 2**). Such approximation requires much less computational complexity (reduction from $\mathcal{O}(C(N, \kappa))$ to $\mathcal{O}(N)$); while still achieving superior filter pruning performance (see Section 4 for evaluation results).

**Definition 2 (Approximated channel independence of combined multiple feature maps)** *For the $i$-th layer with output feature maps $\boldsymbol{\mathcal{A}}^l \in \mathbb{R}^{c^l \times h \times w}$ , the channel independence of combined $m$*

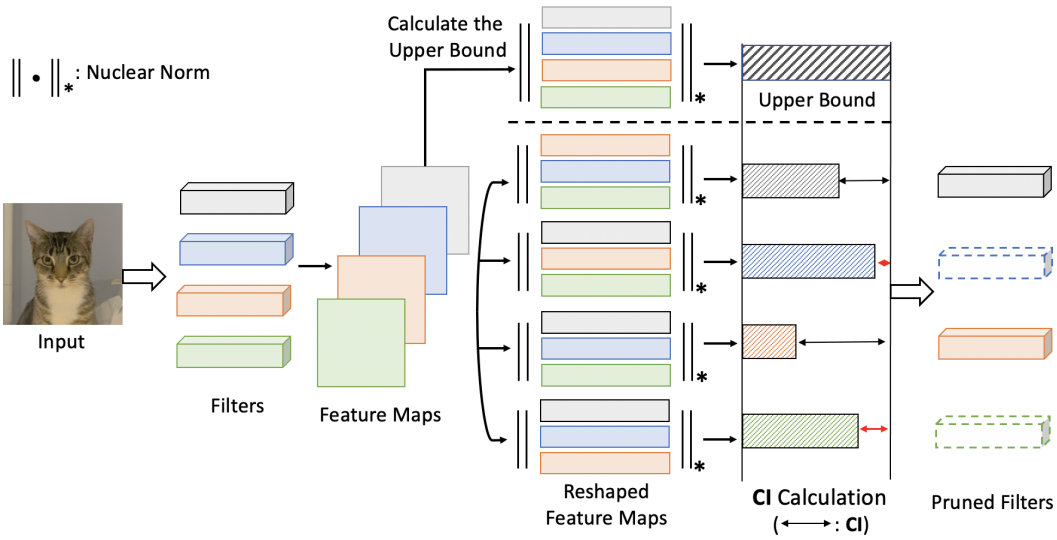



Figure 3: Example of filter pruning process using the change of nuclear norm-based channel independence (CI) criterion.

*feature maps* $\{\boldsymbol{A}_{b_i}^l\}_{i=1}^m$*, where* $\boldsymbol{A}_{b_i}^l \in \mathbb{R}^{h\times w}$ *is in the* $b_i$*-th channel, is defined and approximated as:*

$$CI(\{\boldsymbol{A}_{b_i}^l\}_{i=1}^m) \triangleq \|\boldsymbol{A}^l\|_* - \|\boldsymbol{M}_{b_1,\cdots,b_m}^l \odot \boldsymbol{A}^l\|_* \approx \sum_{i=1}^m CI(\boldsymbol{A}_{b_i}^l), \tag{4}$$

where $\boldsymbol{M}_{b_1,\cdots,b_m}^l$ is the multi-row mask matrix, in which the $b_1,\cdots,b_m$-th row entries are zeros and all the other entries are ones.

**Question #3:** *How is the sensitiveness of channel independence related to the distribution of input data?*

Our Observation. Consider our proposed channel independence-based filter pruning is a data-driven approach, its reliability with different distributions of input data should be carefully ensured and examined. To that end, we perform empirical evaluations on the channel independence with respect to multiple input images. We observe that the *average* channel independence of each feature map is very stable *at the batch level*. In other words, we can simply input a small batches of image samples, and calculate the average channel independence, and then such averaged channel independence with a small number of input data can be used to estimate the channel independence with all the input data. As illustrated in Fig. 4, for the same feature map, the average channel independence in different batches remains very similar, thereby indicating that our channel independence-based approach is robust against different input data.

**Question #4:** *Is this one-shot static importance determination scheme good enough? Do we need to further dynamically learn and adjust the pruning mask from the data?*

Our Observation. As described above, our proposed scheme calculates the channel independence to identify the filter importance. Considering our approach is built on one-shot static calculation, a natural extension is to further dynamically adjust the importance ranking via additional learning. To be specific, if we interpret the filter pruning is a *channel-wise masking operation* over the entire weight tensor, the selection of channel mask can be learned from data, and such learning process can use the pruning mask determined by our approach as the initialization. Though in principle this learning-based strategy is expected to enable additional performance improvement, our empirical evaluations show that the dynamic learning procedure does not bring further accuracy increase (with the target compression ratio) or compression ratio increase (with the target accuracy) – more experimental details are reported in Supplementary Materials. We hypothesize the reason of such phenomenon is that, our proposed nuclear norm change-based channel independence, though only requires one-time calculation, already identifies and captures the importance of feature maps (and its corresponding filters) very accurately, and hence further learning-based adjustment of pruning mask does not provide additional improvement.

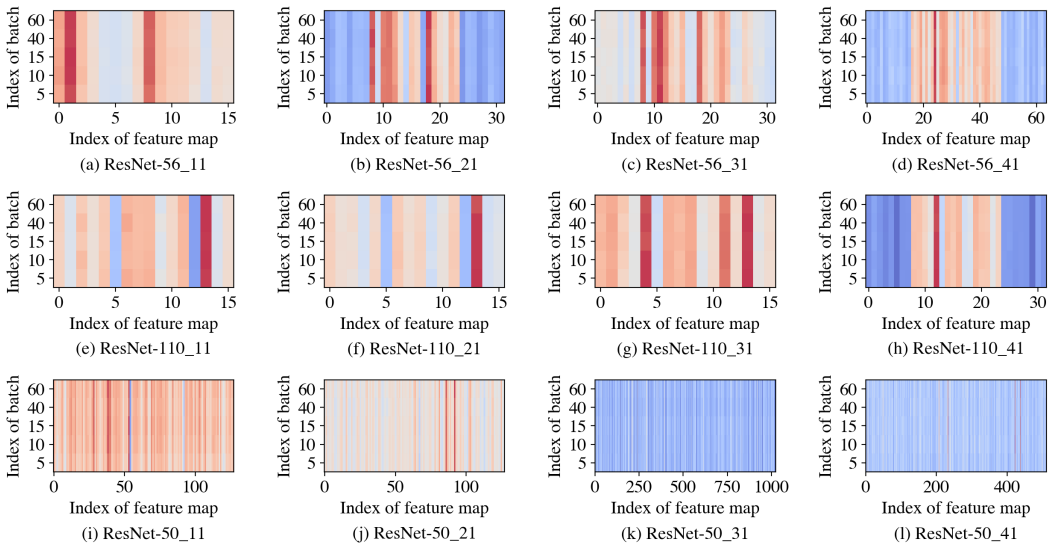



Figure 4: The channel independence of feature maps for one layer in ResNet. Here the channel independence is averaged for one batch of input images. The x-axis is the index of feature map. The y-axis is the index of batches of input images. Here the batch size is 128. **Different colors denote the different values of channel independence. It is seen that the average channel independence is very stable regardless of different input data batches.**

**Algorithm 1** CHannel Independence-based Pruning (CHIP) procedure for the $l$-th layer

**Input:** Pre-trained weight tensor $\boldsymbol{\mathcal{W}}^l$, $N$ sets of feature maps $\boldsymbol{\mathcal{A}}^l = \{\boldsymbol{A}_1^l, \boldsymbol{A}_2^l, \cdots, \boldsymbol{A}_{c^l}^l\} \in \mathbb{R}^{c^l \times h \times w}$ from $N$ input samples, and the desired number of filters to be preserved $\kappa^l$.

**Output:** Pruned weight tensor $\boldsymbol{\mathcal{W}}_{prune}^l$.

**for** each input sample **do**
  **Flatten feature maps:** $\boldsymbol{A}^l := \texttt{reshape}(\boldsymbol{\mathcal{A}}^l, [c^l, hw])$;
  **for** $i = 1$ to $c^l$ **do**
    **CI calculation:** Calculate $CI(\boldsymbol{A}_i^l)$ via Equation 3;
  **end for**
**end for**
**Averaging:** Average $CI(\boldsymbol{A}_i^l)$ under all $N$ input samples;
**Sorting:** Sort $\{CI(\boldsymbol{A}_i^l)\}_{i=1}^{c^l}$ in ascending order;
**Pruning**: Prune $c^l - \kappa^l$ filters in $\boldsymbol{\mathcal{W}}^l$ corresponding to the $c^l - \kappa^l$ smallest $CI(\boldsymbol{A}_i^l)$;
**Fine-tuning:** Obtain final $\boldsymbol{\mathcal{W}}_{prune}^l$ via fine-tuning $\boldsymbol{\mathcal{W}}^l$ with removing the pruned filter channels.

**The Overall Algorithm.** After addressing the above four problems, we can then integrate our proposals and observations to develop the entire filter pruning procedure from the inter-channel perspective. Algorithm 1 describes and summarizes the overall scheme for our proposed CHannel Independence-based filter Pruning (CHIP) algorithm.

# 4 Experiments

## 4.1 Experimental Settings

**Baselines Models and Datasets.** To demonstrate the effectiveness and generality of our proposed channel independence-based approach, we evaluate its pruning performance for various baseline models on different image classification datasets. To be specific, we conduct experiments for three CNN models (ResNet-56, ResNet-110 and VGG-16) on CIFAR-10 dataset [14]. Also, we further evaluate our approach and compare its performance with other state-of-the-art pruning methods for ResNet-50 model on large-scale ImageNet dataset [1].

**Pruning and Fine-tuning Configurations.** We conduct our empirical evaluations on Nvidia Tesla V100 GPUs with PyTorch 1.7 framework. To determine the importance of each filter, we randomly sample 5 batches (640 input images) to calculate the average channel independence of each feature map in all the experiments. After performing the channel independence-based filter pruning, we then perform fine-tuning on the pruned models with Stochastic Gradient Descent (SGD) as the optimizer. To be specific, we perform the fine-tuning for 300 epochs on CIFAR-10 datasets with the batch size, momentum, weight decay and initial learning rate as 128, 0.9, 0.05 and 0.01, respectively. On the ImageNet dataset, fine-tuning is performed for 180 epochs with the batch size, momentum, weight decay and initial learning rate as 256, 0.99, 0.0001 and 0.1, respectively.

Table 1: Experimental results on CIFAR-10 dataset.

| Method | Top-1 Accuracy (%) | | | # Params. (↓%) | FLOPs (↓%) |
|---|---|---|---|---|---|
| | Baseline | Pruned | Δ | | |
| ResNet-56 | | | | | |
| $\ell_1$-norm (2016) [15] | 93.04 | 93.06 | +0.02 | 0.73M (14.1) | 90.90M (27.6) |
| NISP (2018) [34] | 93.26 | 93.01 | -0.25 | 0.49M (42.4) | 81.00M (35.5) |
| GAL (2019) [19] | 93.26 | 92.98 | -0.28 | 0.75M (11.8) | 78.30M (37.6) |
| HRank (2020) [18] | 93.26 | 93.52 | +0.26 | 0.71M (16.8) | 88.72M (29.3) |
| **CHIP (Ours)** | 93.26 | **94.16** | **+0.75** | **0.48M(42.8)** | **65.94M(47.4)** |
| GAL (2019) [19] | 93.26 | 90.36 | -2.90 | 0.29M (65.9) | 49.99M (60.2) |
| LASSO (2017) [11] | 92.80 | 90.80 | -2.00 | N/A | 62.00M (50.6) |
| HRank (2020) [18] | 93.26 | 90.72 | -2.54 | 0.27M(68.1) | 32.52M(74.1) |
| **CHIP (Ours)** | 93.26 | **92.05** | **-1.21** | **0.24M(71.8)** | 34.79M(72.3) |
| ResNet-110 | | | | | |
| $\ell_1$-norm (2016) [15] | 93.50 | 93.30 | -0.20 | 1.16M (32.6) | 155.00M (38.7) |
| HRank (2020) [18] | 93.50 | 94.23 | +0.73 | 1.04M(39.4) | 148.70M(41.2) |
| **CHIP (Ours)** | 93.50 | **94.44** | **+0.94** | **0.89M(48.3)** | **121.09M(52.1)** |
| GAL (2019) [19] | 93.50 | 92.55 | -0.95 | 0.95M (44.8) | 130.20M (48.5) |
| HRank (2020) [18] | 93.50 | 92.65 | -0.85 | 0.53M(68.7) | 79.30M(68.6) |
| **CHIP (Ours)** | 93.50 | **93.63** | **+0.13** | 0.54M(68.3) | **71.69M(71.6)** |
| VGG-16 | | | | | |
| SSS (2018) [13] | 93.96 | 93.02 | -0.94 | 3.93M (73.8) | 183.13M (41.6) |
| GAL (2019) [19] | 93.96 | 92.03 | -1.93 | 3.36M (77.6) | 189.49M (39.6) |
| HRank (2020) [18] | 93.96 | 93.43 | -0.53 | 2.51M(82.9) | 145.61M(53.5) |
| **CHIP (Ours)** | 93.96 | **93.86** | **-0.10** | 2.76M(81.6) | **131.17M(58.1)** |
| GAL (2019) [19] | 93.96 | 90.73 | -3.23 | 2.67M (82.2) | 171.89M (45.2) |
| HRank (2020) [18] | 93.96 | 92.34 | -1.62 | 2.64M(82.1) | 108.61M(65.3) |
| **CHIP (Ours)** | 93.96 | **93.72** | **-0.24** | **2.50M(83.3)** | **104.78M(66.6)** |
| HRank (2020) [18] | 93.96 | 91.23 | -2.73 | 1.78M(92.0) | 73.70M(76.5) |
| **CHIP (Ours)** | 93.96 | **93.18** | **-0.78** | 1.90M(87.3) | **66.95M(78.6)** |

### 4.2 Evaluation and Comparison on CIFAR-10 Dataset

Table 1 shows the evaluation results of the pruned ResNet-56, ResNet-110 and VGG-16 models on CIFAR-10 dataset. To be consistent with prior works, we evaluate the performance for two scenarios: targeting high accuracy and targeting high model size and FLOPs reductions.

**ResNet-56.** For ResNet-56 model, our channel independence-based approach can bring $0.75\%$ accuracy increase over baseline model with $42.8\%$ and $47.4\%$ model size and FLOPs reductions, respectively. When we adopt aggressive compression with $71.8\%$ and $72.3\%$ model size and FLOPs reductions, we can still achieve high performance – our solution enables $1.33\%$ higher accuracy than the state-of-the-art pruning method HRank [18] with the similar model size and computational costs.

**ResNet-110.** For ResNet-110 model, our approach can bring $0.94\%$ accuracy increase over baseline model with $48.3\%$ and $52.1\%$ model size and FLOPs reductions, respectively. When we perform aggressive pruning with $68.3\%$ and $71.6\%$ model size and FLOPs reductions, our pruned model can

still achieve $0.13\%$ higher accuracy over the baseline model; while the other pruning methods suffer nearly $1\%$ accuracy loss under the similar compression setting.

**VGG-16.** For VGG-16 model, our approach can bring $81.6\%$ and $58.1\%$ model size and FLOPs reductions, respectively, with only $0.1\%$ accuracy drop; while the state-of-the-art methods experience at least $0.5\%$ accuracy drop with the similar pruning effect. Moreover, our approach can still provide high accuracy with the further increased compression ratio. Our pruned model, with $83.3\%$ and $66.6\%$ storage and computational cost reductions, achieves $1.38\%$ higher accuracy than the model using other pruning approaches under the similar compression ratio. For even higher FLOPS reduction ($78.6\%$), our method can bring nearly $2\%$ accuracy increase over the prior works.

### 4.3 Evaluation and Comparison on ImageNet Dataset

Table 2 summarizes the pruning performance of our approach for ResNet-50 on ImageNet dataset. It is seen that when targeting moderate compression ratio, our approach can achieve $40.8\%$ and $44.8\%$ storage and computation reductions, respectively, with $0.15\%$ accuracy increase over baseline model. When we further increase the compression ratio, our approach still achieves superior performance than the state-of-the-art works. For instance, compared with the state-of-the-art SCOP [30], our approach shows higher accuracy ($0.12\%$) in the moderate compression region and the same accuracy in the high compress region; while meanwhile enjoying much smaller model size and fewer FLOPs.

Table 2: Experimental results on ImageNet dataset.

| Method | Top-1 Accuracy (%) | | | Top-5 Accuracy (%) | | | Params. | FLOPs |
|---|---|---|---|---|---|---|---|---|
| | Baseline | Pruned | Δ | Baseline | Pruned | Δ | ↓(%) | ↓(%) |
| ResNet-50 | | | | | | | | |
| ThiNet (2017) [22] | 72.98 | 72.04 | -0.94 | 91.14 | 90.67 | -0.47 | 33.72 | 36.8 |
| SFP (2018)[9] | 76.15 | 74.61 | -1.54 | 92.87 | 92.06 | -0.81 | N/A | 41.8 |
| Autopruner (2020) [21] | 76.15 | 74.76 | -1.39 | 92.87 | 92.15 | -0.72 | N/A | 48.7 |
| FPGM (2019) [10] | 76.15 | 75.59 | -0.56 | 92.87 | 92.27 | -0.60 | 37.5 | 42.2 |
| Taylor (2019) [23] | 76.18 | 74.50 | -1.68 | N/A | N/A | N/A | 44.5 | 44.9 |
| C-SGD (2019) [2] | 75.33 | 74.93 | -0.40 | 92.56 | 92.27 | -0.29 | N/A | 46.2 |
| GAL (2019) [19] | 76.15 | 71.95 | -4.20 | 92.87 | 90.94 | -1.93 | 16.9 | 43 |
| RRBP (2019) [36] | 76.10 | 73.00 | -3.10 | 92.9 | 91.00 | -1.90 | N/A | 54.5 |
| PFP (2019) [17] | 76.13 | 75.91 | -0.22 | 92.87 | 92.81 | -0.06 | 18.1 | 10.8 |
| HRank (2020) [18] | 76.15 | 74.98 | -1.17 | 92.87 | 92.33 | -0.54 | 36.6 | 43.7 |
| SCOP (2020) [30] | 76.15 | 75.95 | -0.20 | 92.87 | 92.79 | -0.08 | 42.8 | 45.3 |
| **CHIP (Ours)** | 76.15 | **76.30** | **+0.15** | 92.87 | **93.02** | **+0.15** | 40.8 | 44.8 |
| **CHIP (Ours)** | 76.15 | 76.15 | 0.00 | 92.87 | 92.91 | +0.04 | **44.2** | **48.7** |
| PFP (2019) [17] | 76.13 | 75.21 | -0.92 | 92.87 | 92.43 | -0.44 | 30.1 | 44 |
| SCOP (2020) [30] | 76.15 | 75.26 | -0.89 | 92.87 | 92.53 | -0.34 | 51.8 | 54.6 |
| **CHIP (Ours)** | 76.15 | **75.26** | **-0.89** | 92.87 | **92.53** | **-0.34** | **56.7** | **62.8** |
| HRank (2020) [18] | 76.15 | 71.98 | -4.17 | 92.87 | 91.01 | -1.86 | 46.0 | 62.1 |
| HRank (2020) [18] | 76.15 | 69.10 | -7.05 | 92.87 | 89.58 | -3.29 | 67.5 | 76.0 |
| **CHIP (Ours)** | 76.15 | **73.30** | **-2.85** | 92.87 | **91.48** | **-1.39** | **68.6** | **76.7** |

## 5 Conclusion

In this paper, we propose to use channel independence, an inter-channel perspective-motivated metric, to evaluate the importance of filters for network pruning. By systematically exploring the quantification metric, measuring scheme, and sensitiveness and reliability of channel independence, we develop CHIP, a CHannel Independence-based filter pruning for neural network compression. Extensive evaluation results on different datasets show our proposed approach brings significant storage and computational cost reductions while still preserving high model accuracy.

# References


[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[2] Xiaohan Ding, Guiguang Ding, Yuchen Guo, and Jungong Han. Centripetal sgd for pruning very deep convolutional networks with complicated structure. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4943–4953, 2019.

[3] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[4] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[5] Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic network surgery for efficient dnns. *arXiv preprint arXiv:1608.04493*, 2016.

[6] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[7] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, pages 1135–1143, 2015.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[9] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. Soft filter pruning for accelerating deep convolutional neural networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2234–2240, 2018.

[10] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2019.

[11] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1389–1397, 2017.

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[13] Zehao Huang and Naiyan Wang. Data-driven sparse structure selection for deep neural networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 304–320, 2018.

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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

[16] Yawei Li, Shuhang Gu, Christoph Mayer, Luc Van Gool, and Radu Timofte. Group sparsity: The hinge between filter pruning and decomposition for network compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8018–8027, 2020.

[17] Lucas Liebenwein, Cenk Baykal, Harry Lang, Dan Feldman, and Daniela Rus. Provable filter pruning for efficient neural networks. *arXiv preprint arXiv:1911.07412*, 2019.

[18] 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.

[19] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 2790–2799, 2019.

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[21] Jian-Hao Luo and Jianxin Wu. Autopruner: An end-to-end trainable filter pruning method for efficient deep model inference. *Pattern Recognition*, 107:107461, 2020.

[22] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5058–5066, 2017.

[23] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11264–11272, 2019.

[24] Yu Pan, Jing Xu, Maolin Wang, Jinmian Ye, Fei Wang, Kun Bai, and Zenglin Xu. Compressing recurrent neural networks with tensor ring for action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4683–4690, 2019.

[25] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[26] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pages 525–542. Springer, 2016.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[28] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[29] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[30] Yehui Tang, Yunhe Wang, Yixing Xu, Dacheng Tao, Chunjing XU, Chao Xu, and Chang Xu. Scop: Scientific control for reliable neural network pruning. In *Advances in Neural Information Processing Systems*, pages 10936–10947, 2020.

[31] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2074–2082, 2016.

[32] Yuhui Xu, Yongzhuang Wang, Aojun Zhou, Weiyao Lin, and Hongkai Xiong. Deep neural network compression with single and multiple level quantization. *arXiv preprint arXiv:1803.03289*, 2018.

[33] Yinchong Yang, Denis Krompass, and Volker Tresp. Tensor-train recurrent neural networks for video classification. In *International Conference on Machine Learning*, pages 3891–3900, 2017.

[34] Ruichi Yu, Ang Li, Chun-Fu Chen, Jui-Hsin Lai, Vlad I Morariu, Xintong Han, Mingfei Gao, Ching-Yung Lin, and Larry S Davis. Nisp: Pruning networks using neuron importance score propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9194–9203, 2018.

[35] Tianyun Zhang, Shaokai Ye, Kaiqi Zhang, Jian Tang, Wujie Wen, Makan Fardad, and Yanzhi Wang. A systematic dnn weight pruning framework using alternating direction method of multipliers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 184–199, 2018.

[36] Yuefu Zhou, Ya Zhang, Yanfeng Wang, and Qi Tian. Accelerate cnn via recursive bayesian pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3306–3315, 2019.

# Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Please see "Technical Preview and Contribution" in Section 1
   (b) Did you describe the limitations of your work? [Yes] In Section 3 we indicate that the calculation of channel independence of multiple feature maps is an approximated method to save computational cost.
   (c) Did you discuss any potential negative societal impacts of your work? [No] Model compression can bring more energy-efficient deployment of deep learning, hence there are not potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We have read the ethics review guidelines and follow them when preparing the paper.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] This paper is not a theoretical paper.
   (b) Did you include complete proofs of all theoretical results? [N/A] This paper does not contain theoretical proof.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The code is proprietary.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please see "Experimental Setting" in Section 4 and Supplemental materials
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We indeed have measured the error bars after multiple runs. However, because 1) our results are very stable with respect to different random seeds; and 2) The compared state-of-the-art works in Section 4 do not report error bars, we do not list ours to be consistent with their reporting.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please see "Experimental Setting" in Section 4
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cite the creators of ImageNet and CIFAR-10 datasets in the Reference.
   (b) Did you mention the license of the assets? [N/A] ImageNet and CIFAR-10 are public datasets.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] We do not curate or release new assets.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] ImageNet and CIFAR-10 are public datasets and they are free to download.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] No personal identifiable or offensive content is included in ImageNet or CIFAR-10 datasets.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] This paper is not related to research with human subjects.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] This paper is not related to research with human subjects.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] This paper is not related to research with human subjects.