BMIPRUNE: CLASS- AND LAYER-AWARE ONE-SHOT AVERAGE ACTIVATION-BASED PRUNING FOR CONVOLUTIONAL NEURAL NETWORKS VIA MUTUAL INFORMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Activation-based pruning of convolutional networks has been pursued from two directions: class-agnostic criteria rank filters by the magnitude of their responses, while class-aware criteria rank filters by how strongly a response discriminates between classes. Both are applied across the network without regard to the depth of a filter's layer, even though a filter's role changes with depth. Early convolutions respond to generic low-level structure that is present regardless of the class, while later convolutions compose that structure into class-specific parts. A criterion that asks only what a filter reveals about the label therefore has little to measure where filters are not class-selective, and a purely magnitude-based criterion may discard class information carried by a less active filter. This work proposes BMIPrune, a hybrid, one-shot criterion in which each layer's own profile sets the balance between the two signals. Spatially averaged filter responses are modeled as class-conditional Gaussians, from which a Bayes mutual information (BMI) is computed for each filter. Weighting the posterior entropy by activation magnitude decomposes the resulting score exactly into a magnitude component and a class-information component, and their ratio defines each layer's profile. On VGG16, the magnitude component accounts for about 98% of the profile in the first convolutions and as little as 68% in the last block, so the criterion falls back to magnitude where discriminative signal is absent and becomes class-aware where it is present. Before pruning, each layer is narrowed in isolation and its error curve recorded, giving a threshold-free redundancy score that is combined with the parameter cost of removing one filter from that layer. This sets each layer's budget without reordering filters within the layer, so criterion and budget compose rather than interfere. On VGG19 with CIFAR-10, BMIPrune loses only 1.2 percentage points of accuracy at 60% parameter removal without fine-tuning, in line with other activation-based methods and more than 24 points ahead of uniform baselines. At 80% removal, where the baselines collapse, the loss is 7.7 points, 13 points less than the best competing activation-based method.

1 INTRODUCTION

Modern convolutional networks reach high accuracy, but their size and compute cost make them difficult to deploy on phones, embedded systems, and other devices with tight memory and energy budgets. One way to make a trained network smaller is to prune it: remove the parts that contribute least to its predictions and keep the rest (Hoefler et al., 2021). Removing entire convolutional filters is particularly attractive, because the result is simply a narrower network that runs faster on standard hardware without any special software support. The central question is which filters to remove.

Activation-based pruning answers this question by observing the network at work. A small set of images is passed through the pretrained model, and each filter is judged by how it responds, rather than by the size of its weights. This makes it possible to prune without the original training data, the loss function, or gradient-based retraining, which is valuable whenever those are unavailable or too

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

costly. The pruned network must then perform well as it is, so the choice of filters matters all the more.

A network does not, however, ask the same of every layer. Its early layers detect edges, colors, and textures that occur in images of every class, while its later layers combine these into features that increasingly signal particular classes (Yosinski et al., 2014). Whether a filter is worth keeping therefore depends on where it sits. A strongly responding edge detector that treats every class alike is doing its job well, and so is a late filter that fires rarely but only for one class.

Most activation-based criteria do not take this into account. They judge every layer by a single yardstick, and fall into one of two families. Class-agnostic criteria rank filters by the strength of their activations (Tan & Motani, 2020). They suit early layers, but in late layers they can discard filters whose responses are weak on average yet highly informative about the class. Class-aware criteria rank filters by how well their responses separate the classes (Murti et al., 2023; Mondal et al., 2022). They suit late layers, but in early layers they must rank filters that were never meant to be discriminative. There, almost every filter scores close to zero, the ordering carries little meaning, and an early layer can look expendable even though every later layer depends on it.

The same mismatch affects a second decision: how many filters each layer should lose. When scores from layers with different roles are compared directly, for example in a single global ranking, the budget follows whichever layers score lowest rather than those the network can most afford to lose. Without retraining, such a mistake is permanent. In the experiments reported here, ranking all filters globally by mean activation at 80% parameter removal on VGG16 leaves one late layer with a single channel, and the network does not recover even after fine-tuning.

This work proposes BMIPrune, a retraining-free structured pruning method that lets each layer's role decide how its filters are judged. It requires a pretrained network and a small labeled calibration set, but no training data, loss function, or gradients. BMIPrune first measures, for every layer, how much of its filters' value lies in response strength and how much in class information, and ranks filters within the layer by a mixture of the two weighted accordingly. It then sets the number of filters each layer loses by measuring, on calibration data, how much accuracy the layer loses as it is pruned. The contributions of this work are:

1. **A layer profile that is measured, not assumed.** GlobalBMI is a magnitude-weighted mutual-information functional that decomposes exactly into a bounded activation-magnitude component and a class-information component. Their ratio gives each layer a weight between magnitude and class information, estimated from calibration data before any filter is removed. On VGG networks this weight shifts steadily from magnitude towards class information with depth.
2. **A layer-aware filter ranking and allocation.** LocalBMI ranks filters within a layer by combining activation magnitude and class information according to that layer's weight. Per-layer budgets come from single-layer calibration probes, weighed against the parameters each layer's filters account for. A block- and stage-level extension applies the same idea to ResNets while leaving skip connections intact.
3. **Strong retraining-free results.** Across VGG11, VGG16, VGG19, and ResNet56 on CIFAR-10 and CIFAR-100, BMIPrune retains the most accuracy at 80% parameter removal and above in nearly all VGG configurations. On VGG19 with CIFAR-10 at 90% removal it keeps 59.6% accuracy, compared with 42.0% for GFI-AP and 14.5% for DropNet.

Section 2 presents the related work, section 3 describes GlobalBMI, LocalBMI, and the budget allocation, and Section 4 presents the experiments. Sections 5–7 discuss limitations, implications, and conclusions.

2 Related work

Structured pruning. Structured filter pruning removes entire output channels together with their downstream connections. Early criteria selected filters by weight norm (Li et al., 2017) or encouraged structured sparsity during training (Wen et al., 2016). Later work used batch-normalization scale factors (Liu et al., 2017) or the reconstruction of subsequent feature maps (Luo et al., 2017).

These methods established that the choice of filters depends on what a score measures and on whether pruning requires additional optimization.

Activation-based pruning. Activation-based pruning is a subcategory of structured pruning, in which activation statistics offer a way to inspect a pretrained network directly. APoZ identifies units that seldom produce nonzero responses (Hu et al., 2016), while DropNet uses the mean post-activation response to guide iterative filter removal (Tan & Motani, 2020). HRank scores filters by the rank of their feature maps, favoring maps with richer spatial structure (Lin et al., 2020). CHIP instead measures the independence of feature maps and removes channels whose responses contribute less distinct information (Sui et al., 2021).

Class-aware activation-based pruning. While activation statistics capture how often or how strongly a filter responds, they do not directly measure whether its responses distinguish classes. Class-aware methods address a different aspect of filter importance. Discrimination-aware channel pruning introduces discriminative supervision during optimization (Zhuang et al., 2018). GFI-AP uses class-dependent information to compute globally comparable filter scores and prune adaptively (Mondal et al., 2022). TVSPrune estimates the separability of class-conditional feature distributions and applies pruning iteratively under an accuracy constraint (Murti et al., 2023). DisCEdit similarly identifies discriminative components through distributional analysis for structured pruning and model editing (Murti & Bhattacharyya, 2024).

Mutual information. Mutual information has also informed several distinct pruning criteria. The variational information bottleneck compresses networks by learning to concentrate useful information in a subset of neurons (Dai et al., 2018). MINT measures information shared by filters in adjacent layers (Ganesh et al., 2020). Closer to label-aware selection, HRel ranks filters by mutual information between their activation maps and class labels (Sarvani et al., 2022), while RRWFP considers information-based filter relevance and redundancy (Lu et al., 2024).

Layer-wise budget allocation. A filter ranking alone does not determine how many channels to remove from each layer. For the same global parameter target, different allocations can preserve very different networks: pruning one channel also changes the parameters of its downstream connections, and layers can respond differently to the loss of channels. Allocation is therefore an established part of structured pruning. AMC searches for a layer-specific compression policy using reinforcement learning (He et al., 2018). GFI-AP uses globally comparable importance scores for adaptive pruning (Mondal et al., 2022). Work on random channel pruning also emphasizes the influence of layer configuration (Li et al., 2022).

3 METHODOLOGY

BMIPrune selects channels from a fixed, pretrained convolutional network for a specified reduction in effective parameters. Let f_θ denote the pretrained network, and let $\ell = 1, \ldots, L$ index its prunable layers, with F_ℓ channels in layer ℓ. For input x, let $a_{\ell j}(x, h, w)$ be the post-activation response of channel j at spatial position (h, w). Each feature map is summarized by its spatial average,

$$A_{\ell j}(x) = \frac{1}{H_\ell W_\ell} \sum_{h=1}^{H_\ell} \sum_{w=1}^{W_\ell} a_{\ell j}(x, h, w). \tag{1}$$

For VGG, responses are collected after batch normalization and ReLU; for the prunable interior channels of ResNets, they are collected after the first batch-normalization layer and its ReLU. The following scores use these pooled scalar responses, rather than full feature-map distributions.

3.1 GLOBALBMI: COMBINING ACTIVATION MAGNITUDE WITH DISCRIMINATIVE ABILITY

Filter importance can change with representational depth. Early visual features are often more transferable across classes and tasks, whereas later features can become more specialized (Yosinski et al., 2014). While this pattern does not establish that every early filter has low discriminative value or

that every late filter is class specific, it does motivate a layer-dependent score. At the same time, a criterion based only on class separation could undervalue a generally useful early response, while a criterion based only on response strength could undervalue a weak but distinctive later response. The balance between these effects must therefore be measured rather than assigned by layer depth.

3.1.1 FILTER MAGNITUDE VIA AVERAGE ACTIVATION

The unbounded magnitude statistic is the empirical average of the pooled responses on the calibration images $\mathcal{D}_{\text{cal}}$:

$$U_{\ell j} = \frac{1}{|\mathcal{D}_{\text{cal}}|} \sum_{(x,y)\in\mathcal{D}_{\text{cal}}} A_{\ell j}(x). \tag{2}$$

This is the mean post-activation statistic used to rank filters in DropNet (Tan & Motani, 2020). A larger $U_{\ell j}$ indicates a stronger mean response, but does not by itself identify which classes evoke that response.

3.1.2 FILTER DISCRIMINABILITY VIA BAYES MUTUAL INFORMATION

Activation magnitude alone does not reveal whether a channel responds to different classes in different ways. To capture that distinction, the distribution of the pooled response $A_{\ell j}$ is estimated separately for each class $c \in \{1, \ldots, C\}$. Each class-conditional distribution is represented by a univariate Gaussian fitted to the calibration responses:

$$p_{\ell j}(a \mid Y = c) = \mathcal{N}\big(a; \mu_{\ell jc}, \sigma^2_{\ell jc}\big)\,, \qquad \sigma^2_{\ell jc} \geq 10^{-8}. \tag{3}$$

Here $\mu_{\ell jc}$ and $\sigma^2_{\ell jc}$ are the empirical mean and variance among calibration images of class c. The variance floor prevents numerical problems when a channel produces almost identical responses for every image in that class.

Let π_c denote the proportion of class c in the calibration set. Given an observed response a, the fitted class densities and Bayes' rule assign each class the posterior probability

$$p_{\ell j}(c \mid a) = \frac{\pi_c p_{\ell j}(a \mid c)}{\sum_{k=1}^{C} \pi_k p_{\ell j}(a \mid k)}. \tag{4}$$

This posterior is inferred from one filter's activation, it is not the pretrained network's output probability. When the class-conditional response distributions overlap heavily, observing a gives little indication of its class and the posterior remains close to the prior. When the distributions are more distinguishable, some responses concentrate posterior probability on particular classes, even if their absolute activation is small.

The uncertainty about the class after observing response a is its posterior entropy

$$h_{\ell j}(a) = -\sum_{c} p_{\ell j}(c \mid a) \log p_{\ell j}(c \mid a). \tag{5}$$

Averaging this quantity over the fitted response distribution gives the information supplied by that channel about the label:

$$I_{\ell j} = I(Y; A_{\ell j}) = H(Y) - \mathbb{E}_{A_{\ell j}}[h_{\ell j}(A_{\ell j})]. \tag{6}$$

The expectation is taken over the mixture $p_{\ell j}(a) = \sum_c \pi_c\, p_{\ell j}(a \mid c)$, so each fitted class density contributes in proportion to its calibration prior. The balanced calibration partitions used here give $\pi_c = 1/C$ and $H(Y) = \log C$: mutual information is near zero when a response reveals little about the label and can reach $\log C$ when it identifies the label perfectly under the fitted model. Gauss–Hermite quadrature approximates the class-conditional expectations, and numerical estimates of $I_{\ell j}$

are clipped to $[0, \log C]$. "Bayes Mutual Information" refers to this ordinary mutual information computed from fitted densities through Bayes posteriors; it is not a new definition of mutual information or a Bayesian treatment of the network weights. In particular, a Gaussian cannot represent the exact point mass at zero produced by a post-ReLU filter, so the resulting class information remains a model-based approximation.

3.1.3 BRINGING THE TWO TERMS TOGETHER

GlobalBMI uses a bounded response transform so that activation strength $A_{\ell j}(x)$ can enter the same information-scale decomposition as the class term. Let $s > 0$ be the median of positive pooled calibration responses over all scored layers and filters (or $s = 1$ when no positive response exists), and define

$$m(a) = \frac{\max(a, 0)}{\max(a, 0) + s} \in [0, 1). \tag{7}$$

GlobalBMI assigns each filter the activation-weighted functional

$$B_{\ell j} = \log C - \mathbb{E}_{A_{\ell j}}[(1 - m(A_{\ell j}))h_{\ell j}(A_{\ell j})] \tag{8}$$

$$= \underbrace{\log C\, \mathbb{E}[m(A_{\ell j})]}_{M_{\ell j}} + \underbrace{\mathbb{E}[(1 - m(A_{\ell j}))(\log C - h_{\ell j}(A_{\ell j}))]}_{D_{\ell j}}. \tag{9}$$

The second line is an exact algebraic decomposition of the first under the fitted mixture. Both contributions are nonnegative: $M_{\ell j}$ reflects bounded mean response, while $D_{\ell j}$ reflects posterior class certainty discounted where the response is strong. In general, $D_{\ell j} \neq I_{\ell j}$.

3.2 USING GLOBALBMI TO DETERMINE LAYER PROFILES

For each layer, define its mean magnitude and discrimination masses as $\overline{M}_\ell = F_\ell^{-1} \sum_j M_{\ell j}$ and $\overline{D}_\ell = F_\ell^{-1} \sum_j D_{\ell j}$ respectively. Their relative sizes yield

$$\alpha_\ell = \frac{\overline{M}_\ell}{\overline{M}_\ell + \overline{D}_\ell}, \qquad 1 - \alpha_\ell = \frac{\overline{D}_\ell}{\overline{M}_\ell + \overline{D}_\ell}. \tag{10}$$

If the denominator vanishes numerically, both weights are set to $1/2$. Every channel contributes to this estimate before any channel is removed. The weights summarize the composition of the chosen GlobalBMI functional, not a causal measurement of how much magnitude or class information matters to prediction. Figure 1 shows the mean values for $\overline{M}_\ell$ and $\overline{D}_\ell$ across the convolutional layers of VGG16 for CIFAR-10 and CIFAR-100.

3.3 LOCALBMI: RE-RANKING FILTERS WITHIN EACH LAYER

The profile weights are applied to two within-layer rankings. For a scalar filter score $z_{\ell j}$, let $R_\ell(z_{\ell j})$ denote its tie-aware midrank, normalized as $(\mathrm{midrank}(z_{\ell j}) - 1/2)/F_\ell$. The final within-layer score is

$$S_{\ell j} = \alpha_\ell R_\ell(U_{\ell j}) + (1 - \alpha_\ell) R_\ell(I_{\ell j}). \tag{11}$$

This score is termed LocalBMI, for which smaller values indicate earlier removal within the same layer. Ranking first places the activation and information terms on a common unitless scale. It also avoids interpreting their numerical magnitudes, which have different units, as interchangeable. The weights come from GlobalBMI's M and D masses and the ranked quantities are the raw mean response U and the ordinary fitted-mixture information I.

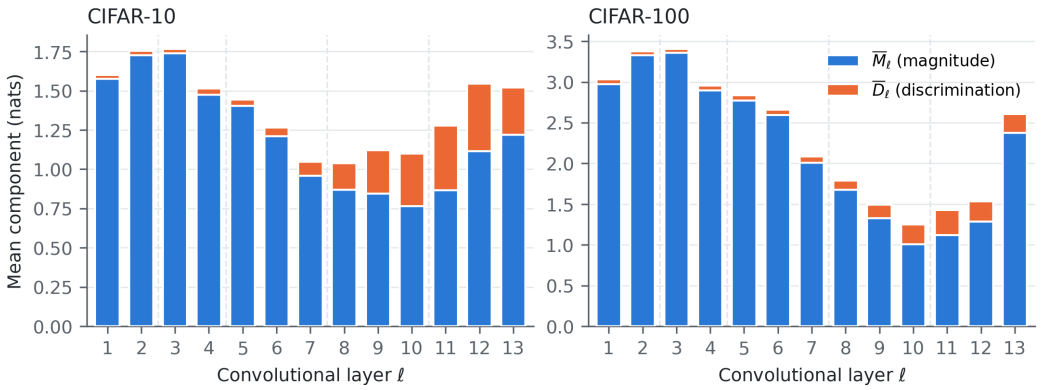

Figure 1: Per-layer GlobalBMI component means $\overline{M}_\ell$ (magnitude) and $\overline{D}_\ell$ (discrimination) for VGG16, measured on the unpruned network.

3.4 The layer-budget allocation problem

LocalBMI establishes an ordering inside each layer, but the parameter target also requires choosing which layers should lose channels. Comparing raw class-information values across layers could favor deleting generally useful early responses solely because they separate labels weakly. Even a well-defined global magnitude score does not establish that equal score values imply equal accuracy loss across layers. Moreover, the effective parameter saving from deleting a channel depends on its connections and the model's architecture. BMIPrune therefore measures how the fixed LocalBMI ordering behaves when each layer is perturbed individually.

For layer ℓ, let $\mathcal{Q}$ be a prespecified grid of *keep* fractions. At each $q \in \mathcal{Q}$, remove the lowest-ranked LocalBMI channels from that layer alone, subject to a minimum number of survivors. Copy the pretrained network, reset and recalibrate its batch-normalization running statistics on the calibration inputs under this mask, and evaluate its calibration classification error $e_\ell(q)$. The reference error e_0 is measured using the same BN-recalibration procedure on the unpruned network. For a balanced C-class partition, define

$$\bar{e}_\ell \simeq \frac{1}{q_{\max} - q_{\min}} \int_{q_{\min}}^{q_{\max}} e_\ell(q)\, dq, \tag{12}$$

$$r_\ell = \mathrm{clip}_{[0,1]} \left(1 - \frac{\bar{e}_\ell - e_0}{\max(1 - 1/C - e_0, \varepsilon)} \right). \tag{13}$$

The first quantity is computed by a trapezoidal rule on the probe grid; $\varepsilon = 10^{-8}$ in the implementation. Larger r_ℓ means less average error growth under these single-layer probes. A small probe error does not guarantee that several layers can be pruned together without interactions.

Let P_0 be the dense effective parameter count and let $c_\ell = P_0 - P(K_{\ell,1})$ where $P(K)$ is the effective parameter count under mask K and $K_{l,1}$ removes one channel from layer ℓ. For VGG, this includes the affected outgoing convolution, batch-normalization parameters, and downstream input weights; a final-layer deletion also affects the classifier. c_ℓ is min–max normalized across layers to $\widetilde{c}_\ell$, forms $u_\ell = r_\ell \widetilde{c}_\ell$, and then min–max normalizes u_ℓ to $b_\ell \in [0, 1]$. If all costs are equal, $\widetilde{c}_\ell = 1$; if all u_ℓ are equal, all bands are set to zero.

Let $Q_\ell(S_{\ell j})$ be the strict, deterministic ordinal within-layer rank of LocalBMI, scaled as $(\mathrm{rank} - 1/2)/F_\ell$. Unlike the midranks used in Eq. 11, the allocation code breaks score ties by channel index. Its global ordering score is

$$G_{\ell j} = 1 - b_\ell + b_\ell Q_\ell(S_{\ell j}). \tag{14}$$

Channels are considered in ascending G order. The effective parameter count is recomputed after each candidate deletion, so interactions in VGG between adjacent narrowed layers are reflected in the achieved sparsity. Pruning stops at the closest attainable parameter target under this order, with at least one channel retained in each prunable layer. Although b_ℓ establishes cross-layer priority, the LocalBMI order is preserved inside any layer with $b_\ell > 0$.

ResNets require a different allocation hierarchy because the stage widths and skip paths constrain which channels can be removed independently. In the implemented extension, the prunable unit is an interior channel of the first convolution in each of BasicBlocks: the corresponding first BN channel and input channel of the second convolution are removed, while block outputs and skip-connection widths remain fixed. GlobalBMI and LocalBMI are computed for these interior channels as above.

The first probe level independently prunes each block in LocalBMI order and measures its calibration-error curve. Within each of the ResNet's stages, the resulting block tolerance is combined with a positive, square-root-scaled single-channel parameter cost. This sets a fixed order over the stage's interior channels, with LocalBMI resolving which channels of a block go first. The second probe level removes channels from one entire stage at a time along that order and measures the stage error curve. Stage tolerances are combined with positive, square-root-scaled mean channel costs to form stage weights. Channels are then removed along a fixed global order derived from these weights until the target is reached; no calibration-accuracy search is used.

4 Results

4.1 Models, data, and partitions

Experiments use the CIFAR-10 and CIFAR-100 datasets (Krizhevsky & Hinton, 2009). VGG11, VGG16, and VGG19 with batch normalization (Simonyan & Zisserman, 2015; Ioffe & Szegedy, 2015) and ResNet56 (He et al., 2016) are pruned from the publicly released pretrained checkpoints (Chen, 2026), using their original input normalization. Before pruning, the full test-set accuracy of each checkpoint was verified to be within two percentage points of the value reported in the repository. All experiments use a single pretrained checkpoint per model.

Pruning decisions use only the CIFAR test set. For each dataset, the 10,000 test images are stratified into five partitions, each consisting of 7,500 calibration images and 2,500 complementary evaluation images (750/250 per class on CIFAR-10, 75/25 per class on CIFAR-100). Every method is applied to the same partition within a repeat, and image indices and their hashes are stored so that the pairing can be verified. All scores, layer probes, BN recalibration, and budget selection use only the calibration images. The evaluation images are used once to evaluate the selected mask at each target. Because the partitions overlap and share one checkpoint, the reported standard deviation measures sensitivity to the calibration split.

4.2 Pruning protocol

BMIPrune is compared with DropNet (adapted as a one-shot mean-activation ranking), GFI-AP (adapted as a one-shot score with the global restricted variant, whose global pruning fraction is searched to meet the parameter target), uniform per-layer ℓ_1-norm pruning, and uniform random pruning. Random pruning uses three independent channel-ranking draws per partition, averaged within each partition. Targets are 10, 20, …, 80, 85, 90, and 95% removal of effective parameters. For VGG, every convolutional layer is prunable; for ResNet56, only the interior channels of each BasicBlock's first convolution are pruned, leaving block outputs and skip connections at full width.

4.3 Pruning results

For each model, target, and method mean and standard deviation of evaluation accuracy across the five partitions are reported, together with the unpruned accuracy on the same evaluation images. Compression is reported as the achieved reduction in effective parameters, i.e. the parameter count of the physically narrowed network after propagating channel removals to downstream layers, rather than the fraction of zeroed weights. The achieved removal can differ slightly from the requested target. FLOPs are counted analytically as twice the multiply–accumulate operations of all convolu-

Table 1: CIFAR-100 accuracy (%) on the 2,500 held-out images directly after pruning and after 20 epochs of fine-tuning on the CIFAR-100 training set.

	VGG16, 80%		ResNet56, 40%	
Method	Pruned	Fine-tuned	Pruned	Fine-tuned
Unpruned	74.6		72.8	
BMIPrune	**18.8**	**72.2**	**34.4**	**71.4**
GFI-AP	6.1	72.1	17.2	70.2
DropNet	1.0	1.0	30.8	71.0

tional and linear layers for a 32×32 input, excluding batch normalization, activations, and pooling. Figure 2 shows the accuracy curves for the four models at the specified pruning targets. Complete tables with the accuracy, drop in accuracy (percentage points) and FLOPs at each pruning target can be found in Appendix A.1.

BMIPrune is able to retain high accuracy even at large compression rates of over 80% for VGGs trained on CIFAR-10 and, while mostly outperforming the baseline metrics and competing with other activation-based methods in early pruning stages, BMIPrune outperforms all metrics in pruning rates above 80% on nearly all VGG configurations. On ResNet56 with CIFAR-10, one-shot DropNet is stronger.

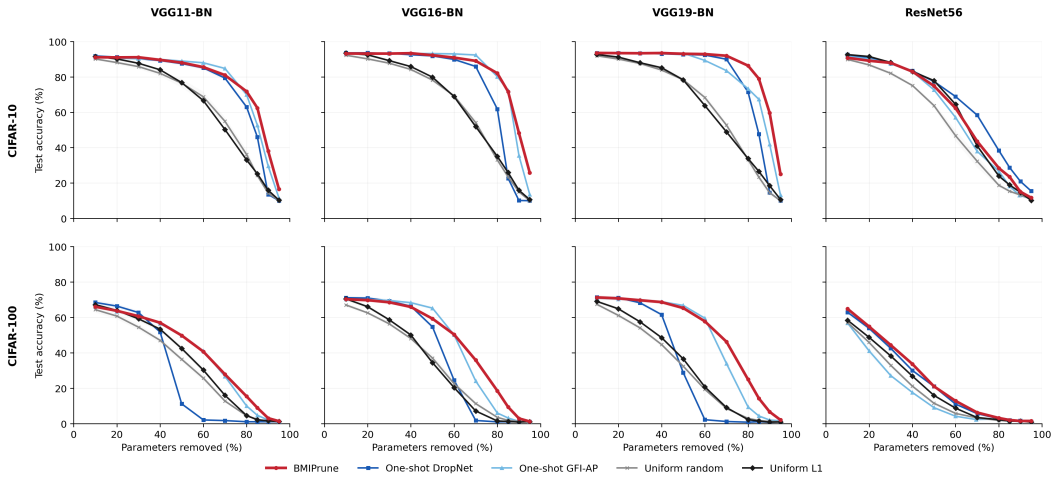

Figure 2: Accuracies at set percentages of parameter removed from VGG11/16/19 and ResNet56 for CIFAR-10 and CIFAR-100 datasets using BMIPrune, one-shot DropNet, one-shot GFI-AP, uniform random pruning, and uniform ℓ_1.

4.4 Fine-tuning

Despite BMIPrune having been developed primarily as a retraining-free method, its capabilities to retain accuracy at high and extreme pruning rates encouraged post-pruning fine-tuning. Each compact network is trained for 20 epochs on the 50,000 CIFAR training images with random cropping and horizontal flipping, SGD (momentum 0.9, weight decay 5×10^{-4}, batch size 128), and a cosine-annealed learning rate starting at 0.01. All three methods in a setting share the same seed and mini-batch order. Accuracy is reported on the 2,500 evaluation images after the final epoch. Accuracies after 20 epochs of retraining for VGG16 and ResNet56 on CIFAR-100 can be seen in Table 1. At moderate pruning level BMIPrune is able to recover most of its lost accuracies even with very few epochs. At the same time, even at extreme pruning rates, BMIPrune is able to recover a large amount of the lost accuracies. At 80% on VGG16, one-shot DropNet is not able to recover any of its lost accuracy. Retraining curves at selected pruning rates can be seen in Appendix A.2.

5 LIMITATIONS

BMIPrune has so far been evaluated on CIFAR-10 and CIFAR-100 with VGG networks and ResNet56, using one publicly released checkpoint per architecture. Extending the evaluation to larger datasets such as ImageNet, to further architectures, and to independently trained checkpoints is needed to establish generalization. The residual variant in particular prunes only the interior channels of each BasicBlock; extending it to skip-connected widths is a natural next step. Likewise, fine-tuning was examined for a small number of configurations, and a broader study is needed to characterize how the advantage before retraining carries over after it.

The main modeling assumption is that each filter's class-conditional response is Gaussian. After a ReLU, many filters produce an exact zero for a large share of images, a distribution that a Gaussian cannot represent. The fitted posteriors, and therefore the information scores and layer profiles, can be distorted for filters whose responses are sparse, skewed, or multimodal. Rectified or zero-inflated densities, or non-parametric estimates, would test how much the method depends on this assumption.

6 DISCUSSION

The layer profiles in Figure 1 support the motivation behind GlobalBMI: in early VGG layers the magnitude component dominates ($\alpha_\ell \approx 0.98$), while the discrimination component grows with depth and peaks in the last block ($\alpha_\ell \approx 0.68$ on CIFAR-10 and 0.78 on CIFAR-100), following the transition from general to class-specific features (Yosinski et al., 2014). Consequently, BMIPrune and DropNet rank filters almost identically in early layers and diverge in the deepest ones (Appendix A.3). The metric and the allocation cannot be separated: each layer's probes prune it in LocalBMI order, so the budget a layer receives already reflects the metric, and within each layer the same order decides which filters go. Their combination explains the large gap at high compression, for example 59.6% against 14.5% on VGG19 with CIFAR-10 at 90% removal. A global ranking by mean activation empties several late, parameter-heavy layers, leaving a single channel in layer 10 of VGG16 at 80% removal on CIFAR-100, whereas BMIPrune keeps every layer above the point where it becomes a bottleneck.

Without retraining, BMIPrune is competitive at moderate compression and strongest at high compression. On CIFAR-10 it stays within one to two points of the best method up to about 50% removal, while GFI-AP and DropNet lead at moderate targets on CIFAR-100, most clearly on VGG16. From 80% onward it achieves the highest accuracy on almost every VGG configuration, and on ResNet56 with CIFAR-100 it is best or joint-best at every target up to 80%. Standard deviations across partitions are mostly below 1.5 points, and the method needs only forward passes and BN recalibration on 7,500 labeled images.

Fine-tuning largely closes the gap between methods wherever the mask leaves a usable path through the network; where a layer has been removed almost entirely, as in DropNet's VGG16 mask, the network does not recover (Table 1). A pruning method's first responsibility is thus a mask that can still be trained, and BMIPrune consistently provided the strongest starting point before retraining. Future work includes a residual allocation tuned as carefully as the VGG allocation, spatially aware information estimates, joint multi-layer probes, and budgets defined by FLOPs or latency.

7 CONCLUSION

This work introduced BMIPrune, a retraining-free structured pruning method that requires only a pretrained network and a small labeled calibration set. GlobalBMI decomposes each filter's score into a magnitude and a class-information component whose balance profiles every layer; LocalBMI turns that profile into a within-layer ranking, and calibration probes turn it into per-layer budgets. Across VGG11, VGG16, VGG19, and ResNet56 on CIFAR-10 and CIFAR-100, BMIPrune retains the most accuracy at high compression in nearly all VGG configurations, often by a wide margin. This advantage arises from the metric and the allocation together: the calibration probes measure each layer's tolerance along the LocalBMI order, so the metric shapes both which filters are removed and how many each layer loses.

AI USE STATEMENT

Artificial intelligence, specifically large language models (LLMs), was used exclusively to aid and polish the written material in this manuscript.

REPRODUCIBILITY STATEMENT

A GitHub repository will be made publicly available upon acceptance of this manuscript to avoid violation of the double-blind review process, but can also be made available upon request of the reviewers.

REFERENCES

Yaofo Chen. PyTorch CIFAR Models. https://github.com/chenyaofo/pytorch-cifar-models, 2026. Accessed: 2026-09-26.

Bin Dai, Chen Zhu, Baining Guo, and David Wipf. Compressing neural networks using the variational information bottleneck. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1135–1144, 2018. URL https://proceedings.mlr.press/v80/dai18d.html.

Madan Ravi Ganesh, Jason J. Corso, and Salimeh Yasaei Sekeh. Mint: Deep network compression via mutual information-based neuron trimming, 2020. URL https://arxiv.org/abs/2003.08472.

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 (CVPR)*, pp. 770–778, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. AMC: AutoML for model compression and acceleration on mobile devices. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 784–800, 2018. URL https://openaccess.thecvf.com/content_ECCV_2018/html/Yihui_He_AMC_Automated_Model_ECCV_2018_paper.html.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks, 2021. URL https://arxiv.org/abs/2102.00554.

Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures, 2016. URL https://arxiv.org/abs/1607.03250.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456, 2015. URL https://proceedings.mlr.press/v37/ioffe15.html.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient ConvNets. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://openreview.net/forum?id=rJqFGTslg.

Yawei Li, Kamil Adamczewski, Wen Li, Shuhang Gu, Radu Timofte, and Luc Van Gool. Revisiting random channel pruning for neural network compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 191–201, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Li_Revisiting_Random_Channel_Pruning_for_Neural_Network_Compression_CVPR_2022_paper.html.

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 (CVPR)*, pp. 1529–1538, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Lin_HRank_Filter_Pruning_Using_High-Rank_Feature_Map_CVPR_2020_paper.html.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2736–2744, 2017. URL https://openaccess.thecvf.com/content_iccv_2017/html/Liu_Learning_Efficient_Convolutional_ICCV_2017_paper.html.

Jingqi Lu, Ruiqing Wang, Guanpeng Zuo, Wu Zhang, Xiu Jin, and Yuan Rao. Enhancing CNN efficiency through mutual information-based filter pruning. *Digital Signal Processing*, 151:104547, 2024. doi: 10.1016/j.dsp.2024.104547.

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 (ICCV)*, pp. 5058–5066, 2017. URL https://openaccess.thecvf.com/content_iccv_2017/html/Luo_ThiNet_A_Filter_ICCV_2017_paper.html.

Milton Mondal, Bishshoy Das, Sumantra Dutta Roy, Pushpendra Singh, Brejesh Lall, and Shiv Dutt Joshi. Adaptive cnn filter pruning using global importance metric. *Computer Vision and Image Understanding*, 222:103511, 2022. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2022.103511. URL https://www.sciencedirect.com/science/article/pii/S1077314222001035.

Chaitanya Murti and Chiranjib Bhattacharyya. DisCEdit: Model editing by identifying discriminative components. In *Advances in Neural Information Processing Systems*, volume 37, pp. 47261–47296, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/54801e196796134a2b0ae5e8adef502f-Abstract-Conference.html.

Chaitanya Murti, Tanay Narshana, and Chiranjib Bhattacharyya. TVSPrune - pruning non-discriminative filters via total variation separability of intermediate representations without fine tuning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sZI1Oj9KBKy.

C. H. Sarvani, Mrinmoy Ghorai, Shiv Ram Dubey, and S. H. Shabbeer Basha. HRel: Filter pruning based on high relevance between activation maps and class labels. *Neural Networks*, 147:186–197, 2022. doi: 10.1016/j.neunet.2021.12.017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1409.1556.

Yang Sui, Miao Yin, Yi Xie, Huy Phan, Saman Aliari Zonouz, and Bo Yuan. CHIP: CHannel independence-based pruning for compact neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24604–24616, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/ce6babd060aa46c61a5777902cca78af-Abstract.html.

Chong Min John Tan and Mehul Motani. DropNet: Reducing neural network complexity via iterative pruning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9356–9366, 2020. URL https://proceedings.mlr.press/v119/tan20a.html.

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*, volume 29, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/41bfd20a38bb1b0bec75acf0845530a7-Abstract.html.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/532a2f85b6977104bc93f8580abbb330-Paper.pdf.

Zhuangwei Zhuang, Mingkui Tan, Bohan Zhuang, Jing Liu, Yong Guo, Qingyao Wu, Junzhou Huang, and Jinhui Zhu. Discrimination-aware channel pruning for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/55a7cf9c71f1c9c495413f934dd1a158-Abstract.html.

A Appendix

A.1 Complete pruning results

The following tables show the complete pruning results for all methods, models and dataset combinations used in this paper.

Table 2: VGG11 on CIFAR-10: held-out accuracy (%), accuracy drop relative to the unpruned network (percentage points), and FLOPs (millions) at each parameter-reduction target, without fine-tuning. Means over five calibration partitions; 0% is the unpruned network. Best accuracy per target in bold.

		Parameters removed (%)											
Method		0	10	20	30	40	50	60	70	80	85	90	95
BMIPrune	Acc.	92.8	91.1	91.0	**91.1**	89.6	88.0	85.5	81.0	**71.8**	**62.4**	**38.0**	**16.5**
	Drop	0.0	1.7	1.8	1.7	3.2	4.7	7.3	11.8	21.0	30.3	54.7	76.3
	MFLOPs	306.6	293.1	278.9	264.1	239.3	210.2	183.1	157.0	127.9	103.9	70.6	28.9
DropNet	Acc.	92.8	**91.8**	**91.2**	90.3	89.2	87.5	85.1	79.3	63.0	46.0	13.5	10.0
	Drop	0.0	0.9	1.6	2.5	3.6	5.3	7.7	13.4	29.8	46.8	79.2	82.8
	MFLOPs	306.6	287.0	266.9	245.8	225.5	201.3	176.2	147.0	114.5	98.5	74.3	16.4
GFI-AP	Acc.	92.8	91.0	90.6	90.2	**89.9**	**88.9**	**88.0**	**84.8**	70.0	52.6	29.6	11.4
	Drop	0.0	1.8	2.2	2.5	2.9	3.8	4.8	8.0	22.8	40.1	63.2	81.3
	MFLOPs	306.6	298.7	288.0	270.0	251.7	233.4	217.9	196.3	166.5	141.5	102.1	47.8
ℓ_1	Acc.	92.8	91.6	90.1	87.7	84.0	76.8	66.6	50.2	33.0	25.2	15.9	10.3
	Drop	0.0	1.2	2.6	5.1	8.8	16.0	26.2	42.5	59.8	67.5	76.9	82.5
	MFLOPs	306.6	274.8	243.3	211.1	179.9	148.0	116.7	85.0	53.6	37.9	22.4	7.0
Random	Acc.	92.8	90.1	88.1	85.8	82.0	76.4	68.7	54.9	36.1	24.2	14.2	10.1
	Drop	0.0	2.6	4.7	7.0	10.8	16.3	24.0	37.9	56.7	68.6	78.5	82.6
	MFLOPs	306.6	274.8	243.3	211.1	179.9	148.0	116.7	85.0	53.6	37.9	22.4	7.0

A.2 Retraining accuracy curves

Figure 3 shows the test-set accuracies during each epoch during fine-tuning for the three tested methods for VGG16 + CIFAR-100 at 80% removed parameters, ResNet56 + CIFAR-100 at 40% removed parameters, VGG16 + CIFAR-10 at 95% removed parameters, and VGG19 + CIFAR-10 at 90% removed parameters.

A.3 Rank difference between BMIPrune and magnitude-based DropNet

Figure 4 plots the rank of each filter per layer of VGG16 trained on CIFAR-10 once for BMIPrune (which is a hybrid consisting of filter magnitude and filter discriminability) and DropNet (raw average filter magnitude). In early layers the two metrics agree almost entirely, naturally, since the discriminative factor within BMIPrune is small in those layers, while the two metrics diverge in later layers, as discriminability starts playing a larger role within each layer.

Table 3: VGG16 on CIFAR-10: held-out accuracy (%), accuracy drop relative to the unpruned network (percentage points), and FLOPs (millions) at each parameter-reduction target, without fine-tuning. Means over five calibration partitions; 0% is the unpruned network. Best accuracy per target in bold.

		Parameters removed (%)											
Method		0	10	20	30	40	50	60	70	80	85	90	95
BMIPrune	Acc.	94.2	93.2	93.2	93.2	**93.4**	92.2	90.9	89.0	**82.1**	71.6	**48.3**	**25.8**
	Drop	0.0	1.0	1.0	1.0	0.8	2.0	3.3	5.2	12.1	22.6	45.9	68.4
	MFLOPs	627.5	601.7	575.6	549.4	515.7	475.8	436.8	391.7	331.7	278.9	222.0	140.7
DropNet	Acc.	94.2	**93.6**	**93.6**	**93.3**	92.5	91.8	89.9	85.9	61.8	22.7	10.1	10.0
	Drop	0.0	0.6	0.6	0.9	1.7	2.4	4.3	8.3	32.4	71.5	84.1	84.2
	MFLOPs	627.5	592.3	563.3	529.8	494.8	456.2	414.0	359.2	304.0	263.7	214.4	132.0
GFI-AP	Acc.	94.2	93.2	93.2	93.2	93.2	**93.2**	**93.0**	**92.4**	80.2	**72.2**	35.5	13.5
	Drop	0.0	1.0	1.0	1.0	1.0	1.0	1.2	1.8	14.0	22.0	58.7	80.7
	MFLOPs	627.5	612.2	599.3	581.9	559.7	532.7	495.1	449.7	394.6	353.6	294.9	173.1
ℓ_1	Acc.	94.2	93.4	92.5	89.2	85.8	79.8	68.9	51.9	35.1	25.9	15.9	10.6
	Drop	0.0	0.8	1.7	5.0	8.4	14.4	25.3	42.3	59.1	68.3	78.3	83.6
	MFLOPs	627.5	563.7	498.4	433.7	370.6	307.3	242.9	178.5	114.5	82.9	51.3	20.0
Random	Acc.	94.2	92.2	90.2	87.7	84.2	78.2	69.3	54.1	33.0	23.5	15.0	10.0
	Drop	0.0	2.0	4.0	6.5	10.0	16.0	24.9	40.1	61.2	70.7	79.2	84.2
	MFLOPs	627.5	563.7	498.4	433.7	370.6	307.3	242.9	178.5	114.5	82.9	51.3	20.0

Table 4: VGG19 on CIFAR-10: held-out accuracy (%), accuracy drop relative to the unpruned network (percentage points), and FLOPs (millions) at each parameter-reduction target, without fine-tuning. Means over five calibration partitions; 0% is the unpruned network. Best accuracy per target in bold.

		Parameters removed (%)											
Method		0	10	20	30	40	50	60	70	80	85	90	95
BMIPrune	Acc.	94.1	**93.5**	93.4	93.3	**93.5**	93.0	**92.9**	**91.9**	**86.3**	**79.0**	**59.6**	**25.1**
	Drop	0.0	0.5	0.6	0.7	0.6	1.0	1.2	2.1	7.7	15.1	34.5	69.0
	MFLOPs	797.3	760.5	723.4	686.1	642.8	592.6	541.9	486.1	394.6	332.5	258.7	179.4
DropNet	Acc.	94.1	93.4	**93.5**	**93.4**	93.0	92.6	92.3	90.0	71.4	47.7	14.5	10.0
	Drop	0.0	0.7	0.6	0.7	1.0	1.4	1.7	4.1	22.6	46.4	79.5	84.1
	MFLOPs	797.3	746.3	696.3	647.6	596.5	548.3	499.8	436.4	367.0	318.6	259.4	172.1
GFI-AP	Acc.	94.1	93.4	93.4	**93.4**	93.4	**93.4**	89.3	83.5	73.3	67.3	42.0	13.2
	Drop	0.0	0.6	0.6	0.6	0.6	0.7	4.8	10.5	20.8	26.8	52.1	80.9
	MFLOPs	797.3	770.6	745.6	719.8	690.3	654.3	612.4	553.2	478.3	431.8	355.7	231.7
ℓ_1	Acc.	94.1	92.7	91.2	88.0	85.2	78.3	63.8	48.9	33.8	26.6	18.4	10.7
	Drop	0.0	1.4	2.9	6.1	8.9	15.7	30.3	45.1	60.2	67.5	75.6	83.4
	MFLOPs	797.3	717.4	633.9	553.2	473.2	392.4	312.0	229.9	149.8	109.6	69.4	29.2
Random	Acc.	94.1	91.9	90.1	87.6	83.9	78.4	68.3	52.7	33.2	23.3	14.5	10.0
	Drop	0.0	2.1	4.0	6.5	10.2	15.7	25.7	41.3	60.8	70.8	79.6	84.1
	MFLOPs	797.3	717.4	633.9	553.2	473.2	392.4	312.0	229.9	149.8	109.6	69.4	29.2

Table 5: ResNet56 on CIFAR-10: held-out accuracy (%), accuracy drop relative to the unpruned network (percentage points), and FLOPs (millions) at each parameter-reduction target, without fine-tuning. Means over five calibration partitions; 0% is the unpruned network. Best accuracy per target in bold.

		Parameters removed (%)											
Method		0	10	20	30	40	50	60	70	80	85	90	95
BMIPrune	Acc.	94.5	90.8	89.1	87.9	83.0	74.9	62.3	43.6	28.4	23.5	15.1	11.6
	Drop	0.0	3.7	5.4	6.6	11.5	19.6	32.2	50.9	66.1	71.0	79.4	82.9
	MFLOPs	251.5	236.7	218.9	197.8	169.6	141.2	112.9	84.4	56.2	41.9	27.9	13.8
DropNet	Acc.	94.5	**92.6**	90.5	87.9	**83.4**	77.4	**68.9**	**58.4**	**38.3**	**28.8**	**20.9**	**15.5**
	Drop	0.0	1.9	4.1	6.6	11.1	17.1	25.6	36.2	56.2	65.8	73.6	79.1
	MFLOPs	251.5	219.8	196.7	177.3	160.0	141.2	122.7	103.2	78.1	57.0	35.7	20.6
GFI-AP	Acc.	94.5	91.6	90.2	87.4	83.0	72.7	57.1	37.9	26.6	18.2	13.1	11.0
	Drop	0.0	3.0	4.4	7.1	11.5	21.9	37.5	56.6	67.9	76.3	81.4	83.5
	MFLOPs	251.5	234.5	217.6	200.3	175.9	146.9	115.8	81.6	46.9	33.7	23.1	14.4
ℓ_1	Acc.	94.5	**92.6**	**91.5**	**88.2**	83.0	**77.9**	64.4	41.0	23.9	18.9	14.7	10.1
	Drop	0.0	1.9	3.0	6.4	11.5	16.6	30.1	53.5	70.6	75.6	79.8	84.4
	MFLOPs	251.5	224.8	202.5	174.5	152.7	126.0	99.4	76.0	49.0	36.7	25.2	13.1
Random	Acc.	94.5	89.9	86.8	82.0	75.1	63.7	46.7	32.3	18.7	15.3	13.2	10.9
	Drop	0.0	4.7	7.8	12.5	19.4	30.8	47.8	62.2	75.8	79.2	81.3	83.6
	MFLOPs	251.5	224.8	202.5	174.5	152.7	126.0	99.4	76.0	49.0	36.7	25.2	13.1

Table 6: VGG11 on CIFAR-100: held-out accuracy (%), accuracy drop relative to the unpruned network (percentage points), and FLOPs (millions) at each parameter-reduction target, without fine-tuning. Means over five calibration partitions; 0% is the unpruned network. Best accuracy per target in bold.

		Parameters removed (%)											
Method		0	10	20	30	40	50	60	70	80	85	90	95
BMIPrune	Acc.	71.1	66.1	63.7	60.7	**57.0**	**49.8**	40.7	**27.8**	**15.4**	**8.9**	**3.0**	**1.3**
	Drop	0.0	5.0	7.3	10.4	14.0	21.3	30.4	43.2	55.7	62.2	68.0	69.7
	MFLOPs	306.7	294.7	280.2	265.3	238.8	210.1	184.3	157.4	123.7	100.2	64.7	24.2
DropNet	Acc.	71.1	**68.5**	**66.4**	**62.8**	51.7	11.1	2.0	1.7	1.0	1.0	1.0	1.0
	Drop	0.0	2.6	4.7	8.3	19.4	59.9	69.0	69.4	70.0	70.1	70.1	70.1
	MFLOPs	306.7	298.2	290.3	282.4	274.3	261.1	230.4	197.5	151.7	123.2	74.8	6.3
GFI-AP	Acc.	71.1	66.3	63.3	60.8	56.5	**49.8**	**41.1**	26.7	10.0	4.6	2.0	1.1
	Drop	0.0	4.8	7.8	10.3	14.5	21.3	30.0	44.4	61.1	66.4	69.1	69.9
	MFLOPs	306.7	299.3	287.9	269.0	247.6	224.9	199.7	174.0	140.3	120.3	86.6	36.5
ℓ_1	Acc.	71.1	67.2	64.0	59.2	53.4	42.4	30.2	16.0	4.5	2.1	1.6	1.1
	Drop	0.0	3.8	7.1	11.9	17.7	28.6	40.8	55.1	66.5	69.0	69.5	70.0
	MFLOPs	306.7	274.7	243.3	211.1	179.4	147.3	115.6	84.0	52.5	36.8	21.2	5.8
Random	Acc.	71.1	64.4	60.7	54.5	47.0	36.3	25.7	12.7	4.4	2.3	1.3	1.1
	Drop	0.0	6.7	10.3	16.6	24.0	34.7	45.4	58.3	66.6	68.8	69.8	69.9
	MFLOPs	306.7	274.7	243.3	211.1	179.4	147.3	115.6	84.0	52.5	36.8	21.2	5.8

Table 7: VGG16 on CIFAR-100: held-out accuracy (%), accuracy drop relative to the unpruned network (percentage points), and FLOPs (millions) at each parameter-reduction target, without fine-tuning. Means over five calibration partitions; 0% is the unpruned network. Best accuracy per target in bold.

		Parameters removed (%)											
Method		0	10	20	30	40	50	60	70	80	85	90	95
BMIPrune	Acc.	73.3	70.3	69.7	68.6	65.9	59.3	**50.4**	**35.8**	**18.5**	**9.3**	**2.9**	**1.1**
	Drop	0.0	3.0	3.5	4.7	7.4	14.0	22.9	37.4	54.8	63.9	70.4	72.1
	MFLOPs	627.5	611.2	585.7	557.6	528.1	488.7	449.1	402.6	336.0	280.4	221.6	132.9
DropNet	Acc.	73.3	**71.1**	**70.9**	69.3	66.3	54.7	24.5	1.7	0.9	1.0	1.0	1.0
	Drop	0.0	2.2	2.3	4.0	6.9	18.6	48.8	71.6	72.3	72.3	72.3	72.3
	MFLOPs	627.5	596.7	571.7	546.0	521.9	494.5	466.6	432.7	385.2	341.2	249.8	120.5
GFI-AP	Acc.	73.3	70.4	70.2	**69.6**	**68.3**	**65.2**	50.1	24.0	6.0	3.1	1.2	**1.1**
	Drop	0.0	2.9	3.1	3.7	5.0	8.0	23.2	49.3	67.3	70.2	72.1	72.2
	MFLOPs	627.5	615.0	601.4	587.0	563.8	534.1	493.7	447.0	377.2	321.9	235.9	122.3
ℓ_1	Acc.	73.3	70.5	66.1	58.7	50.1	34.4	20.3	7.1	1.5	1.2	0.8	**1.1**
	Drop	0.0	2.8	7.2	14.6	23.2	38.9	53.0	66.2	71.8	72.1	72.4	72.2
	MFLOPs	627.5	563.3	498.4	433.4	369.4	306.1	241.9	178.1	113.3	82.1	50.2	18.5
Random	Acc.	73.3	67.0	62.6	56.4	48.1	37.0	22.5	11.3	3.5	1.8	1.2	**1.1**
	Drop	0.0	6.3	10.7	16.9	25.2	36.3	50.7	62.0	69.8	71.5	72.1	72.2
	MFLOPs	627.5	563.3	498.4	433.4	369.4	306.1	241.9	178.1	113.3	82.1	50.2	18.5

Table 8: VGG19 on CIFAR-100: held-out accuracy (%), accuracy drop relative to the unpruned network (percentage points), and FLOPs (millions) at each parameter-reduction target, without fine-tuning. Means over five calibration partitions; 0% is the unpruned network. Best accuracy per target in bold.

		Parameters removed (%)											
Method		0	10	20	30	40	50	60	70	80	85	90	95
BMIPrune	Acc.	73.4	71.3	70.8	69.7	68.7	65.3	57.8	**46.3**	**24.9**	**14.2**	**6.6**	**2.0**
	Drop	0.0	2.1	2.6	3.7	4.8	8.1	15.6	27.1	48.5	59.2	66.8	71.4
	MFLOPs	797.4	771.5	732.8	693.9	654.9	609.8	556.9	505.2	405.4	340.1	262.2	174.5
DropNet	Acc.	73.4	**71.4**	**71.1**	68.3	61.6	28.8	2.2	1.2	0.8	1.0	0.7	0.9
	Drop	0.0	2.1	2.3	5.1	11.8	44.6	71.2	72.3	72.6	72.4	72.7	72.5
	MFLOPs	797.4	750.3	697.4	642.0	590.9	541.6	497.8	453.8	389.7	350.6	293.5	165.2
GFI-AP	Acc.	73.4	70.9	70.4	**69.9**	**68.9**	**66.8**	**59.6**	34.0	9.4	4.3	2.0	1.6
	Drop	0.0	2.5	3.0	3.5	4.5	6.6	13.8	39.4	64.0	69.1	71.4	71.9
	MFLOPs	797.4	778.1	760.6	741.9	715.3	675.3	615.2	552.7	476.9	419.0	325.9	184.4
ℓ_1	Acc.	73.4	69.1	64.9	57.5	48.6	36.6	20.7	9.0	2.2	1.4	0.8	0.9
	Drop	0.0	4.3	8.5	15.9	24.8	36.9	52.7	64.4	71.2	72.0	72.6	72.5
	MFLOPs	797.4	717.4	633.7	553.1	471.5	392.2	310.4	228.5	147.9	108.0	67.4	27.9
Random	Acc.	73.4	67.4	61.1	54.1	44.7	32.4	19.3	8.5	2.9	1.7	1.1	1.1
	Drop	0.0	6.0	12.3	19.3	28.7	41.0	54.1	64.9	70.5	71.7	72.3	72.4
	MFLOPs	797.4	717.4	633.7	553.1	471.5	392.2	310.4	228.5	147.9	108.0	67.4	27.9

Table 9: ResNet56 on CIFAR-100: held-out accuracy (%), accuracy drop relative to the unpruned network (percentage points), and FLOPs (millions) at each parameter-reduction target, without fine-tuning. Means over five calibration partitions; 0% is the unpruned network. Best accuracy per target in bold.

		Parameters removed (%)											
Method		0	10	20	30	40	50	60	70	80	85	90	95
BMIPrune	Acc.	72.3	**64.9**	**54.8**	**44.4**	**33.7**	**21.0**	**12.8**	**6.2**	**3.1**	**2.0**	1.4	**1.5**
	Drop	0.0	7.5	17.5	28.0	38.6	51.3	59.6	66.1	69.2	70.3	70.9	70.9
	MFLOPs	251.5	210.7	187.3	163.8	140.2	116.6	92.9	69.1	45.4	33.6	21.8	12.3
DropNet	Acc.	72.3	62.9	53.9	42.5	30.0	**21.0**	11.0	5.6	2.7	1.9	**1.7**	1.0
	Drop	0.0	9.4	18.4	29.8	42.3	51.4	61.3	66.8	69.6	70.4	70.7	71.3
	MFLOPs	251.5	221.9	200.8	178.3	158.7	138.2	113.8	86.8	53.1	41.4	31.6	19.5
GFI-AP	Acc.	72.3	56.9	41.1	27.1	17.6	9.0	4.1	2.3	2.6	1.5	1.1	1.2
	Drop	0.0	15.5	31.2	45.2	54.8	63.3	68.2	70.0	69.7	70.8	71.2	71.1
	MFLOPs	251.5	188.9	160.0	136.7	111.2	91.3	70.1	51.9	31.1	25.3	17.8	12.4
ℓ_1	Acc.	72.3	58.4	48.7	38.2	26.9	15.9	8.7	3.4	2.2	1.4	1.4	0.8
	Drop	0.0	13.9	23.6	34.1	45.5	56.4	63.6	68.9	70.2	71.0	71.0	71.5
	MFLOPs	251.5	224.7	201.9	174.2	151.4	125.8	98.6	75.3	48.5	34.6	21.7	12.3
Random	Acc.	72.3	57.1	45.9	32.9	21.2	11.5	5.7	3.2	1.8	1.5	1.1	1.1
	Drop	0.0	15.2	26.5	39.4	51.2	60.8	66.6	69.1	70.6	70.9	71.2	71.3
	MFLOPs	251.5	224.7	201.9	174.2	151.4	125.8	98.6	75.3	48.5	34.6	21.7	12.3

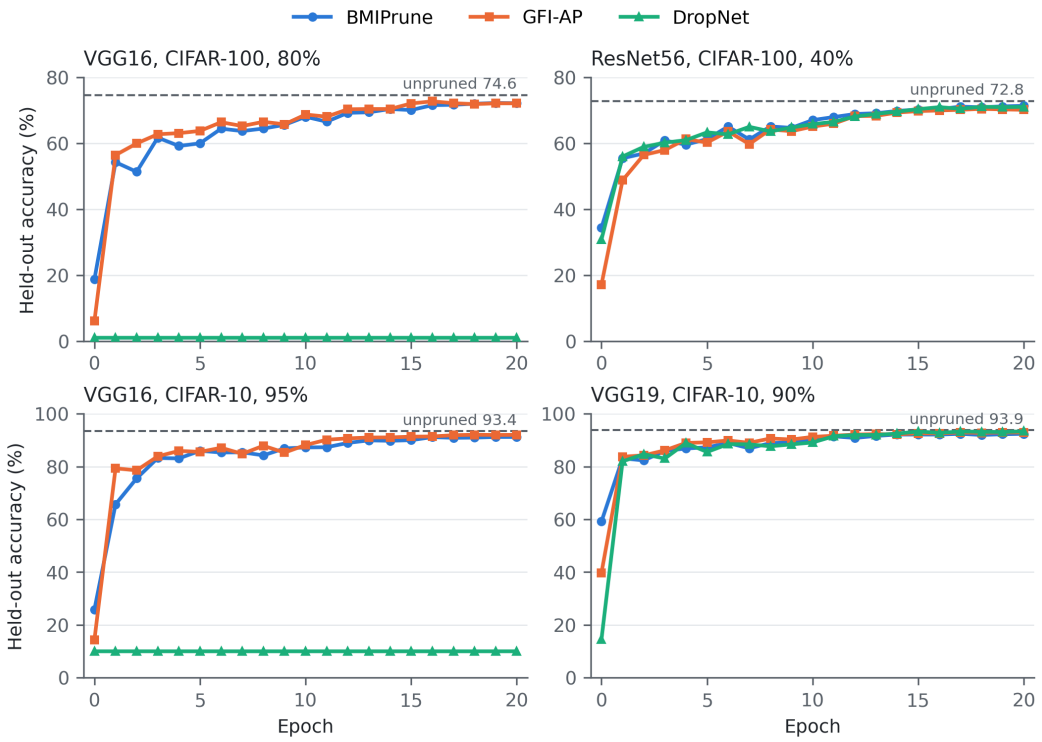

Figure 3: Accuracy curves during fine-tuning at each of the 20 epochs for all three methods for different models, datasets, and pruning ratios.

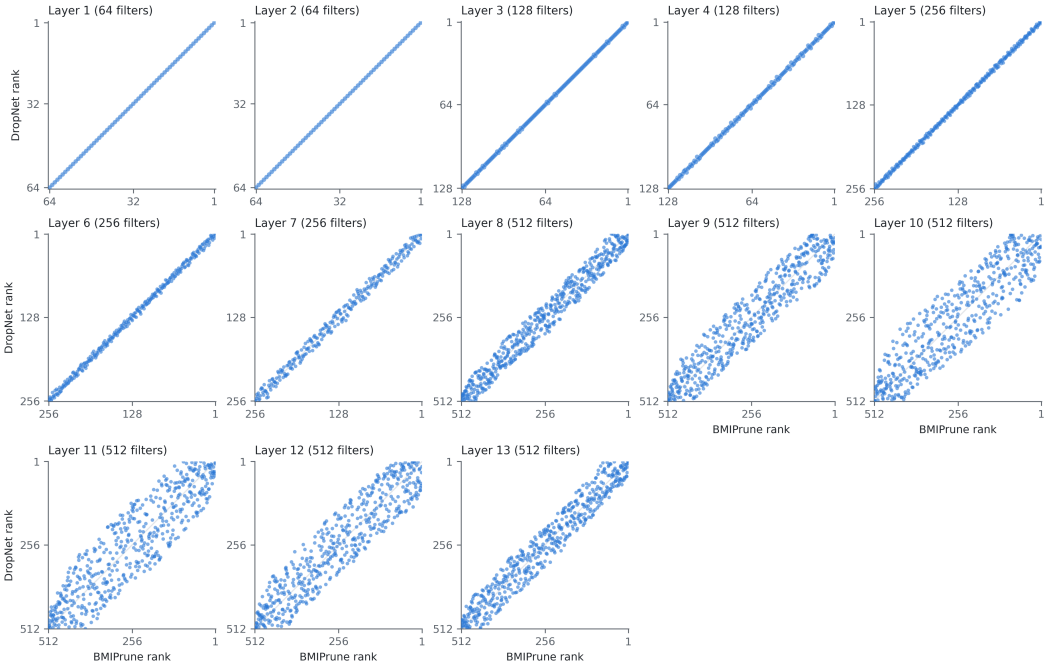

Figure 4: Filter rank per layer for VGG16 trained on CIFAR-10 for BMIPrune and DropNet.