Interpretations Steered Network Pruning via Amortized Inferred Saliency Maps

Alireza Ganjdanesh*, Shangqian Gao*, and Heng Huang

Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA 15261, USA
{alireza.ganjdanesh, shg84, heng.huang}@pitt.edu
(* indicates equal contribution)

Abstract. Convolutional Neural Networks (CNNs) compression is crucial to deploying these models in edge devices with limited resources. Existing channel pruning algorithms for CNNs have achieved plenty of success on complex models. They approach the pruning problem from various perspectives and use different metrics to guide the pruning process. However, these metrics mainly focus on the model's 'outputs' or 'weights' and neglect its 'interpretations' information. To fill in this gap, we propose to address the channel pruning problem from a novel perspective by leveraging the interpretations of a model to steer the pruning process, thereby utilizing information from both inputs and outputs of the model. However, existing interpretation methods cannot get deployed to achieve our goal as either they are inefficient for pruning or may predict non-coherent explanations. We tackle this challenge by introducing a selector model that predicts real-time smooth saliency masks for pruned models. We parameterize the distribution of explanatory masks by Radial Basis Function (RBF)-like functions to incorporate geometric prior of natural images in our selector model's inductive bias. Thus, we can obtain compact representations of explanations to reduce the computational costs of our pruning method. We leverage our selector model to steer the network pruning by maximizing the similarity of explanatory representations for the pruned and original models. Extensive experiments on CIFAR-10 and ImageNet benchmark datasets demonstrate the efficacy of our proposed method. Our implementations are available at https://github.com/Alii-Ganjj/InterpretationsSteeredPruning

Keywords: Convolutional Neural Networks - Model Compression - Efficient Deep Learning - Interpretability - Explainable AI

1 Introduction

Convolutional Neural Networks (CNNs) have been continuously achieving state-of-the-art results on various computer vision tasks [10, 13, 38, 54, 4, 44, 45], but the required resources of popular deep models [55, 16, 23] are also exploding. Their substantial computational and storage costs prohibit deploying these models in edge and mobile devices, making the CNN compression problem a crucial task. Many ideas have attempted to address this problem to reduce models' sizes while maintaining their prediction performance. These ideas can usually be classified into one of the model compression methods categories: weight pruning [15], weight quantization [7, 43], structural pruning [30], knowledge distillation [21], neural architecture search [20], *etc.*

We focus on pruning channels of CNNs (structural pruning) since it can effectively and practically reduce the computational costs of a deep model without any post-processing steps or specifically designed hardware. Although existing channel pruning methods have achieved excellent results, they do not consider the model's interpretations during the pruning process. They tackle the pruning problem from various perspectives such as reinforcement learning [20], greedy search [62], and evolutionary algorithms [8]. In addition, they have utilized a wide range of metrics like channels' norm [30], loss [11], and accuracy [36] as guidance to prune the model. Thus, they emphasize the model's outputs or weights but ignore its valuable interpretations' information.

We aim to approach the structural model pruning problem from a novel perspective by exploiting the model's interpretations (a subset of input features called saliency maps) to steer the pruning. Our intuition is that the saliency maps of the pruned model should be similar to the ones for the original model. However, the existing interpretation methods are either inefficient or unreliable for pruning. Firstly, locally linear models (*e.g.*, LIME [46] and SHAP [39]) fit a separate linear model to explain the behavior of a nonlinear classifier in the vicinity of each data point. However, they need to fit a new model in each iteration of pruning that the classifier's architecture changes, which makes them inefficient for pruning. Secondly, previous works [22, 1] empirically observed that a feature importance assignment of Gradient-based methods (*e.g.*, Grad-CAM [50] and DeepLIFT [52]) might not be more meaningful than random. Moreover, Srinivas and Fleuret [58] theoretically showed that the input gradients used by these methods might seem explanatory as they are related to an implicit generative model hidden in the classifiers [14], not their discriminative function. Thus, their usage for interpreting classifiers should be avoided. Finally, perturbation-based methods [65, 71] need multiple forward passes and rely on perturbed samples that are out-of-distribution for the trained model [22] to obtain its explanations. Hence, they are neither efficient nor reliable for pruning. Different from the mentioned methods, **Amortized Explanation Models (AEMs)** [26, 6, 64] provide a theoretical framework to obtain a model's interpretations. They train a fast saliency prediction model that can be applied in real-time systems as it can provide saliency maps with a single forward pass, making them suitable for pruning. We refer to section 2 for more discussion on interpretation methods.

In this paper, at first, we provide a new AEM method that overcomes the disadvantages of previous AEM models, and then, we employ it to prune convolutional classifiers. Previous AEMs [26, 6, 64] cannot be applied to guide pruning due to several key drawbacks. REAL-X [26] proved that L2X [6] and INVASE [64] could suffer from degenerate cases where the saliency map selector predicts meaningless explanations. Although REAL-X overcomes this problem, it generates masks independently for each input feature (pixel). Thus, it neglects the geometric prior [5] in natural images that adjacent features (pixels) often correlate to each other. We empirically show in Section 3.3 and Fig. 1 that the saliency maps predicted by REAL-X may lack visual interpretability. In addition, the provided explanations have the same size as the input image, which also adds non-trivial computational costs when used for pruning. We propose a novel AEM model to tackle these problems. In contrast with REAL-X, which assumes features' independence, we employ a proper geometric prior in our model. We use a Radial Basis Function (RBF)-like function to parameterize saliency masks' distribution. By doing so, the mask generation is no longer independent for each pixel in our framework. Moreover, it enables us to infer explanations for each image with only three parameters (center coordinates and kernel expansion), saving lots of computations. We utilize such compact saliency representations to steer network pruning by reconstruction in real-time. We also find that merging guidance from the model's interpretations and outputs can further improve the pruning results. Our experimental results on benchmark datasets illustrate that our new interpretation steered pruning method can consistently achieve superior performance compared to baselines. Our contributions are as follows:

– We propose a novel structural pruning method for CNNs designed from a new and different perspective compared to existing methods. We utilize the model's decisions' interpretations to steer the pruning procedure. By doing so, we effectively merge the guidance from the model's interpretations and outputs to discover the high-performance subnetworks.
– We introduce a new Amortized Explanation Model (AEM) such that we embed a proper geometric prior for natural images in the inductive bias of our model and enable it to predict smooth explanations for input images. We parameterize the distribution of saliency masks using RBF-like functions. Thus, our AEM can provide compact explanatory representations and save computational costs. Further, it empowers us to dynamically obtain saliency maps of pruned models and leverage them to steer the pruning procedure.
– Our experimental results on CIFAR-10 and ImageNet datasets clearly demonstrate the added value of using interpretations of CNNs when pruning them.

2 Related Works

Interpretation Methods: Interpretation methods can get classified into four [26] main categories: **1. Gradient-based** methods such as CAM [68], Grad-CAM [50], DeepLIFT [52], and LRP [3] rely on the gradients of outputs of a model *w.r.t* input features and assume features with larger gradients have more influence on the model's outcome [53, 57, 56], which is shown is not necessarily a valid assumption [51]. In addition, their feature importance assignment might not be more meaningful than random assignment [22, 1, 58], which makes them unreliable for pruning. Further, Srinivas and Fleuret [58] theoretically proved that input gradients are equal to the score function for the implicit generative model in classifiers [14] and are not related to the discriminative function of classifiers.

Thus, they are not interpretations of the model's predictions. **2. Perturbation-based** models explore the effect of perturbing input features on the model's output or inner layers to conclude their importance [65, 71, 69]. Yet, they are inefficient for pruning as they need multiple forward passes to obtain importance scores. Also, they may underestimate features' importance [52]. **3. Locally Linear Models** fit a linear model to approximate the behavior of a classifier in the vicinity of each data point [46, 39]. However, they require to fit a new model for each sample when the model's architecture changes during pruning, which makes them inefficient for pruning. Also, they rely on the classifier's output for out-of-distribution samples to train the linear model [22], which makes them undependable. **4. Amortized Explanation Models (AEMs)** [26, 64, 6, 9] overcome the inefficiencies of the previous methods by training a *global* model - called *selector* [26] - that *amortizes* the cost of inferring saliency maps for each sample by *selecting* salient input features with a single forward pass. AEMs [26, 6, 64] provide a theoretical framework to train the selector model. To do so, they use a second *predictor* model that estimates the classifier's output target distribution given an input masked by the selector model's predicted mask. L2X [6] and INVASE [64] jointly train the selector and predictor. However, REAL-X [26] proved that doing so results in degenerate cases. REAL-X overcame this problem by training the predictor model separately with random masks. However, we show in section 3.3 that its predicted masks may not be interpretable for complex image classifiers. Our conjecture for a reason is that it neglects geometric prior [5] of natural images that nearby pixels correlate more to each other.

Network Compression: Weight pruning [15] and quantization [7, 43], structural pruning [30, 60, 19, 41, 70, 42, 35, 33, 66, 59, 12], knowledge distillation [21], and NAS [20] are popular directions for compressing CNNs. Structural pruning has attracted more attention as it can readily decrease the computational burden of CNN models without any specific hardware changes. Early channel pruning methods [30] propose that the channels with larger norms are more critical and remove weights/filters with small L_1/L_2 norm. L_1 penalty can also be applied to scaling factors of batchnorm [24] to remove redundant channels [37]. Recent channel pruning methods adopt more sophisticated designs. Automatic model compression [20] learns the width of each layer with reinforcement learning. Metapruning [36] generates parameters for subnetworks and uses evolutionary algorithms to find the best subnetwork. Greedy subnetwork selection [62] greedily chooses each channel based on their L_2 norm. Pruning can be also used for fairness [67]. We refer to [32] for a more detailed discussion of pruning techniques.

Network Pruning Using Interpretations: There are a few recent works that attempt to use interpretations of a model to determine importance scores of its weights. Sabih *et al.* [48] leverage DeepLIFT [52]; Yeom *et al.* [63] use LRP [3]; and Yao *et al.* [61] utilize activation maximization [65] to determine weights' importance. However, all these methods use gradient-based methods that, as mentioned above, their predictions are unreliable and should not be used as the model's interpretations. Alqahtani *et al.* [2] visualize feature maps in the input space and use a segmentation model to find the filters that have the highest alignment with visual concepts. Nonetheless, their method needs an accurate segmentation model to find reliable importance scores for filters, which may not be available in some domains. We develop a new AEM model that is theoretically supported and improves REAL-X [26]. Moreover, in contrast with these methods, our pruning method finds the optimal subnetwork end-to-end. We also show in section 4.2 that our model outperforms [2].

3 Methodology

3.1 Overview

We present a novel pruning method in which we steer the pruning process of CNN classifiers using feature-wise interpretations of their decisions. At first, we develop a new intuitive AEM model that overcomes the limitations of REAL-X [26] (state-of-the-art AEM). The reason is that we incorporate the geometric prior of high correlation between adjacent input features (pixels) [5] in the images in the inductive bias of our AEM model. We parameterize the distribution of saliency masks using Radial Basis Function (RBF)-style functions. By doing so, we can represent interpretations (saliency maps) of input images compactly. Then, we elaborate on our pruning method in which we leverage our AEM model to provide interpretations of the original and pruned classifiers. Our intuition is that saliency maps of the original and pruned models should be similar. Thus, we propose a new loss function for pruning that encourages the pruned model to have similar saliency explanations to the original one. In the following subsections, we introduce AEM methods and empirically show the limitations of REAL-X. Then, we elaborate on our method and its intuitions to tackle the drawbacks of previous AEMs. Finally, we present our pruning scheme.

3.2 Notations

We denote our dataset as $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ such that $(x, y) \sim \mathcal{P}(\mathbf{x}, \mathbf{y})$ where $\mathcal{P}$ is the unknown underlying joint distribution over features and targets, and we assume that $x \in \mathbb{R}^D$ and $y \in \{1, 2, \dots, K\}$. We show the jth feature of sample x by x_j and represent a mask m by the indices of the input features that it preserves, *i.e.*, $m \subseteq \{1, 2, \dots, D\}$ and a masked input $m(x)$ is defined as follows:

$$m(x) = mask(x, m) = \begin{cases} x_j & j \in m \\ 0^{1} & \text{Otherwise} \end{cases} \tag{1}$$

We call the model that we aim to prune as the 'classifier' in following sections.

[1] We use zero values for the masked input features following the literature.[64, 6, 26]

3.3 Amortized Explanation Models (AEMs)

AEMs are a subgroup of Instance-Wise Feature Selection (IWFS) methods that aim to compute a mask with minimum cardinality for each input sample that preserves its outcome-related features. An outcome may be a classifier's predictions (usually calculated as a softmax distribution) for interpretation purposes. It can also be the population distribution of the targets (one-hot representations) when performing dimensionality reduction on the original raw data [6, 26, 64]. Although previous works [64, 6, 26] describe their formulation for the latter, we focus on the former in this paper.

Concretely, if $\mathcal{Q}_{class}(\mathbf{y}|\mathbf{x})$ be the classifier's conditional distribution of targets given input features, the objective of AEM models is to find a mask $m(x)$ for each sample x such that

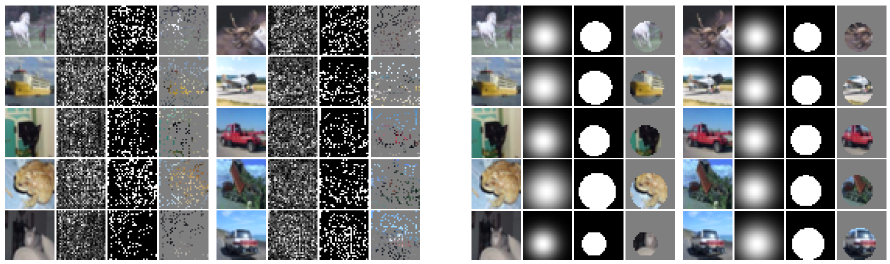
$$\mathcal{Q}_{class}(\mathbf{y}|\mathbf{x}=x) = \mathcal{Q}_{class}(\mathbf{y}|\mathbf{x}=m(x)) \tag{2}$$

AEMs tackle this problem by training a *global* model called *selector* that learns to predict a *local* (sample dependent) mask $m(x)$ for each sample x [26]. They train the selector by encouraging it to follow Eq. 2. To do so, one should quantify the discrepancy between the RHS and LHS of Eq. 2 when the selector model generates the mask m in the RHS. The LHS can be readily calculated by forwarding the sample x into the classifier. However, the classifier should not be used to compute the RHS because the masked sample $m(x)$ is an out-of-distribution input for it [26]. AEMs solve this issue by training a *predictor* model that predicts the conditional distribution of the classifier given a masked input. (RHS of Eq. 2) Then, they train the selector guided by the supervision from the predictor. We present the formulation of REAL-X [26] in supplementary.

Visualization of REAL-X Predictions: We visualize predicted explanations of REAL-X for a ResNet-56 model [16] trained on CIFAR-10 [29] in Fig. 1(a). (we refer to supplementary materials for implementation details) As can be seen, the formulation of REAL-X cannot guide the selector model to learn to select a coherent subset of input pixels of the salient parts of the images. Thus, it may not provide interpretable explanations for the classifier. Our conjecture for the cause is that the formulation of REAL-X does not include a proper inductive bias related to natural images in the selector model. Typically, nearby pixels' values and semantic information are more correlated in natural images, known as their geometric prior [5]. REAL-X does not have such a prior in its formulation because it factorizes the explanatory masks' distribution given an input x as:

$$q_{sel}(m|x;\beta) = \prod_{i=1}^{D} q_i(m_i|x;\beta) \tag{3}$$

where $q_i(m_i|x;\beta) \sim Bernoulli((f_\beta(x))_i)$, *i.e.*, the distribution over the selector's output mask is factorized as a product of marginal Bernoulli distributions over mask's elements, and the parameter for each element gets calculated independently. ($f_\beta(x)$ is the selector model parameterized by β). Hence, the selector



(a) **REAL-X** [26]   (b) **Ours**

Fig. 1: Input features selected by **a) REAL-X** [26] and **b) our model** to explain decisions of a ResNet-56 classifier for samples from CIFAR-10 [29]. In the subfigures from left to right: 1st column shows the original image. Both models output an array (2nd columns) that each value of it is the parameter of the predicted Bernoulli distribution over the corresponding mask pixel. In the 3rd column, we show the masks generated such that a pixel's value is one provided that its predicted Bernoulli parameter is bigger than 0.5 and zero otherwise. The 4th columns show the masked inputs. Our model's explanations are easier to interpret than the ones by REAL-X that may seem random for some samples.

model does not have the inductive bias that parameters of nearby Bernoulli distributions should be close to each other to make the sampled masks coherent. Instead, it should 'discover' such prior during training, which is infeasible with limited data and training epochs in practice.

3.4 Proposed AEM Model

We introduce a new selector scheme that respects the proximity geometric prior. To do so, we assume that the parameters of the Bernoulli distributions of mask pixels should have a Radial Basis Function (RBF) style functional form over the pixels. The center of the RBF kernel should be on the salient part of the image most relevant to the classifier's prediction, and the Bernoulli parameters should decrease as the pixel location gets far from the kernel's center. A parameter σ controls the area of a mask. Our assumption is reasonable for multi-class classifiers in which, typically, a single object/region in their input image determines the target class. Formally, considering a 2D mask that its coordinates are parametrized by (z, t) and the parameters of a 2D RBF kernel being (c_z, c_t, σ), we calculate the Bernoulli parameter (BP) of a pixel at location (z, t) as follows:

$$f_{BP}(z, t; c_z, c_t, \sigma) = \exp\left(\frac{-1}{2\sigma^2}[(z - c_z)^2 + (t - c_t)^2]\right) \tag{4}$$

This formulation has two crucial benefits: 1) It ensures that Bernoulli parameters of a mask's proximal pixels are close to each other. Thus, the resulting sampled masks will be much more coherent and smooth than REAL-X. 2) It simplifies the selector model's task significantly. In REAL-X, the selector should learn how to calculate Bernoulli parameters for each pixel that, for instance, will be $224 \times 224 = 50176$ independent functions for the standard ImageNet [10] training. In contrast, in our formulation, the selector should only learn to accurately estimate three values corresponding to the center's coordinates (c_z, c_t) and an expanding parameter σ for the RBF kernel. Given the estimated values, Bernoulli parameters of the output mask's pixels can be readily calculated by Eq. 4. In other words, if the input images have spatial dimensions $M * N$, and we denote the selector function (implemented by a deep neural network) with $f_{sel}(x;\beta)$, our selector's distribution over masks given input images is:

$$\begin{aligned} [c_z, c_t, \sigma] &= f_{sel}(x;\beta) \\ q_{i,j}(m_{i,j}|x;\beta) &= Bernoulli(f_{BP}(i,j;c_z,c_t,\sigma)) \\ q_{sel}(m|x;\beta) &= \prod_{i=1}^{M}\prod_{j=1}^{N} q_{i,j}(m_{i,j}|x;\beta) \end{aligned} \tag{5}$$

In Eq. 5, β denotes the selector's parameters, and we illustrate a predicted RBF kernel by our selector in Fig. 2. In summary, our intuition is that by incorporating the geometric prior in the inductive bias of our framework, the selector will search for proper functional form for Bernoulli parameters over pixels' locations in the RBF family of functions, not all possible ones. As a result, it can find the optimal functional form more readily and robustly. Moreover, our selector model can provide a real-time and compact representation (RBF parameters) for saliency maps, which enables us to efficiently compare the interpretations of the original and pruned models to steer the pruning process. (section 3.6, Fig. 3)

3.5 AEM Training

We train our selector model by encouraging it to generate an explanatory mask m for each sample x such that it follows Eq. 2. To do so, as mentioned in section 3.3, we need to estimate the classifier's conditional distribution of targets given masked inputs (RHS of Eq. 2) to train our selector model. Such an estimate can quantify the quality of a mask generated by the selector model by measuring the discrepancy between the LHS and RHS of Eq. 2.

Predictor Model: We train a predictor model to calculate the classifier's conditional distribution of targets given a masked input. (RHS of Eq. 2) As we designed our selector to predict RBF-style masks (Eq. 5), we train our predictor to predict the classifier's output distribution when the input is masked by a random RBF-style mask. Using random RBF masks allows us to mimic any potential RBF-masked input. Hence, our predictor's training objective is:

$$\min_{\theta} \mathcal{L}_{pred}(\theta) = \mathbb{E}_{x\sim\mathcal{P}(\mathbf{x})}\mathbb{E}_{c'_z,c'_t,\sigma'}[\mathbb{E}_{m'\sim\mathcal{B}(m|c'_z,c'_t,\sigma')}L_\theta(x,m'(x))] \tag{6}$$

where $L_\theta(\cdot,\ \cdot)$ and $\mathcal{B}(\cdot)$ are defined as:

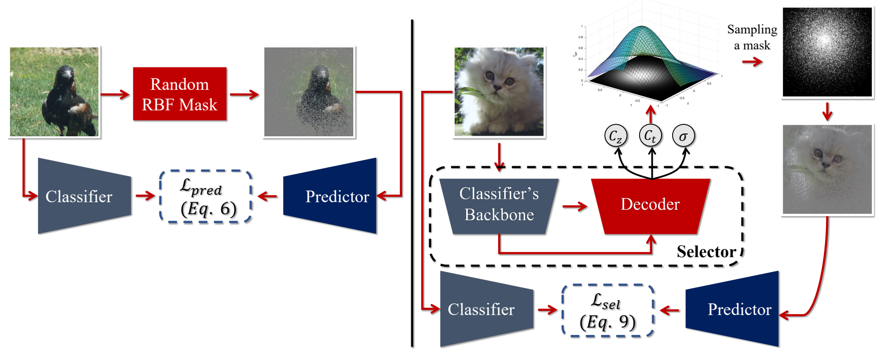


Fig. 2: Our AEM model. The goal is to train the selector model on the right (U-Net model in dashed line) to predict interpretations (saliency maps) of the classifier for each input sample. We train the selector by encouraging it to follow Eq. 2. **(Left):** We train a predictor model that learns to predict the classifier's output distribution given a masked input (RHS of Eq. 2). We do so using inputs masked by random RBF masks as our selector's masks have RBF-style. (Sec. 3.4) **(Right):** Given the trained predictor, we train the selector model using obj. 8 that enforces it to follow Eq. 2. We use the classifier's convolutional backbone as the encoder of the selector and only train its decoder for computational efficiency. Then, we use the trained decoder to prune the encoder. (Fig. 3)

$$
\begin{aligned}
&L_\theta(x, m'(x)) = KL(\mathcal{Q}_{class}(\mathbf{y}|\mathbf{x}=x),\ q_{pred}(\mathbf{y}|\mathbf{x}=m'(x);\theta)) \\
&\mathcal{B}(m|c'_z, c'_t, \sigma') = \prod_{i=1}^{M}\prod_{j=1}^{N} Bernoulli(f_{BP}(i,j;c'_z, c'_t, \sigma'))
\end{aligned}
\tag{7}
$$

Eq. (6), L_θ form the predictor's objective to learn the conditional distribution of the classifier for targets given masked inputs (RHS of Eq. 2). $\mathcal{B}(\cdot)$ generates random masks with random RBF style (f_{BP}), and KL denotes Kullback-Leibler divergence [27]. Now, we should define the distribution for the parameters c'_z, c'_t, and σ' for a random RBF function. Let us assume that the origin of our 2D coordinate system is the top left of an input image with spatial dimensions M, N. In theory, c'_z and c'_t can have any real values, and the σ' can be any positive real number in Eq. 4. However, considering that the salient part[s] is inside the image region, we are interested that the predictor learns to correctly estimate $\mathcal{Q}_{class}(\mathbf{y}|\mathbf{x}=m(x))$ (RHS of Eq. 2) when the selector predicts that the center of the RBF kernel is inside the image area. Hence, we assume that the distributions of c'_z and c'_t are uniform across image dimensions, *i.e.*, $c'_z \sim U[0, M]$ and $c'_t \sim U[0, N]$. In addition, the parameter σ' determines the degree that an RBF kernel expands on the image, and the values $\sigma' \geq 2 * max\{M, N\}$ practically provide the same Bernoulli parameters for all the mask's pixels when c'_z and c'_t are in the image region. Thus, we can reasonably assume that $\sigma' \sim U[0,\ 2 * max\{M,\ N\}]$ for training the predictor in practice.

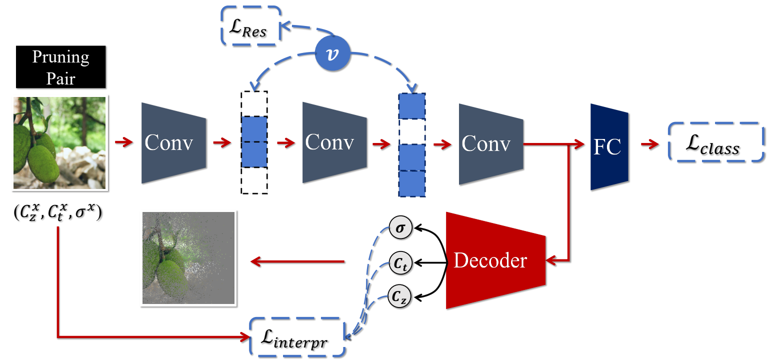


Fig. 3: Our pruning method. The classifier to be pruned is shown on top. (Conv layers and FC). The U-Net (Conv layers and the Decoder) is our trained selector model that can predict RBF parameters of the saliency map of each input for the classifier. The selector model is trained such that the pretrained backbone of the classifier is used as its encoder (Conv layers) and kept frozen during training. (see Fig. 2) Thus, we freeze the selector and classifier's weights and insert our pruning gates between the selector's encoder layers for pruning the classifier. Given a pruning pair (a sample and its RBF saliency map's parameters for the original classifier), we train the gate parameters to prune the classifier such that the pruned model have similar interpretations ($\mathcal{L}_{interpr}$) and accuracy ($\mathcal{L}_{class}$) to the original classifier while requiring lower computational resources ($\mathcal{L}_{Res}$).

Selector Training: Given a predictor model denoted by q_{pred} and trained with random RBF masks, we train our selector model with the following objective:

$$\min_{\beta} \mathcal{L}_{sel}(\beta) = \mathbb{E}_{x\sim\mathcal{P}(\mathbf{x})}\mathbb{E}_{m'\sim q_{sel}(m|x;\beta)}[L(x,m'(x)) + \lambda_1\mathcal{R}(m') + \lambda_2\mathcal{S}(m')] \quad (8)$$

such that $L(\cdot,\ \cdot)$, $\mathcal{R}(\cdot)$, and $\mathcal{S}(\cdot)$ are defined as:

$$\begin{aligned} &L(x,m'(x)) = KL(\mathcal{Q}_{class}(\mathbf{y}|\mathbf{x}=x),\ q_{pred}(\mathbf{y}|\mathbf{x}=m'(x))), \\ &\mathcal{R}(m') = ||m'||_0,\ \mathcal{S}(m') = \sum_{i=1}^{M}\sum_{j=1}^{N}[(m'_{i,j} - m'_{i+1,j})^2 + (m'_{i,j} - m'_{i,j+1})^2] \end{aligned} \quad (9)$$

$L(x,m'(x))$ encourages the selector to follow Eq. 2 as $q_{pred}(\mathbf{y}|\mathbf{x}=m'(x))$ approximates the RHS of Eq. 2 given an input masked by the RBF mask predicted by the selector. $\mathcal{R}(m')$ regularizes the number of selected features. We add the smoothness loss $\mathcal{S}(m')$ to further encourage the selector to output smooth masks. As Eq. 8 requires sampling from predicted distribution by the selector, direct backpropagation of gradients to train its parameters, β, is not possible. Thus, we use the Gumbel-Sigmoid [25, 40] trick to train the model. We use a U-Net [47] architecture to implement the selector module of our AEM model, as shown in Fig. 2. We refer to supplementary for more details of our AEM training.

3.6 Pruning

In this section, we introduce our pruning method that leverages interpretations of a classifier to steer its pruning process. Our intuition is that the interpretations (saliency maps) of the original and pruned classifiers should be similar. Thus, we design our pruning method as follows. As discussed in section 3.5 and Fig. 2, we use the convolutional backbone of the classifier as the encoder of the U-Net architecture for the selector model. We keep the encoder weights frozen and only train the decoder when training the selector model for computational efficiency. (Fig. 2) Furthermore, doing so provides us the flexibility to keep the decoder frozen and prune the encoder such that the pruned model should have similar output RBF parameters to the original model. (Fig. 3)

Formally, we employ our trained selector model to predict saliency maps of the original classifier for training samples. For each sample x_k, it provides the parameters of the RBF kernel for its saliency map as $\mathcal{C}_{x_k} = [c_z^k, c_t^k, \sigma^k]$. Then, we insert our pruning gates, parameterized by θ_g, between the layers of the encoder. We represent the architectural vector generated by the gates with $\mathbf{v}$. Finally, we prune the encoder (classifier's backbone) by regularizing the gate parameters to maintain the interpretations and accuracy of the pruned classifier similar to the original one while reducing its computational budget as follows:

$$\min_{\theta_g} L(f(x; \mathcal{W}, \mathbf{v}), y) + \gamma_1 \|\mathcal{C}_x - f_{sel}(x; \beta, \mathbf{v})\|_2^2 \ + \gamma_2 \mathcal{R}_{res}(T(\mathbf{v}), pT_{all}) \tag{10}$$

where $L(\cdot,\cdot)$ is the classification loss, $f(\cdot; \mathcal{W}, \mathbf{v})$ denotes our classifier (encoder of the U-Net and the FC layer in Fig. 3) parameterized by weights $\mathcal{W}$ and the subnetwork selection vector $\mathbf{v}$. $f_{sel}(x; \beta, \mathbf{v})$ is our trained selector model ($f_{sel}(x; \beta)$ in Eq. 5) augmented by the architecture vector $\mathbf{v}$ after inserting the pruning gates into its encoder. We calculate $\mathbf{v}$ using Gumbel-sigmoid function $g(\cdot)$: $\mathbf{v} = g(\theta_g)$ [25, 40], which controls openness or closeness of a channel. The second term in Eq. 10 utilizes the interpretations of the original and pruned classifiers to steer pruning through the selector model $f_{sel}(x; \beta, \mathbf{v})$ by encouraging the similarity of their predicted RBF parameters. $\mathcal{R}_{res}$ is the FLOPs regularization to ensure the pruned model reaches the desired FLOPs rate pT_{all}. T_{all} is the total prunable FLOPs of a model, $T(\mathbf{v})$ is the current FLOPs rate determined by the subnetwork vector $\mathbf{v}$, and p controls the pruning rate. γ_1 and γ_2 are hyperparameters to control the strength of related terms. During pruning, we only optimize θ_g and keep $\mathcal{W}$ and β frozen.

We emphasize that our amortized explanation prediction selector model, $f_{sel}(x; \beta, \mathbf{v})$, enables us to readily perform interpretation-steered pruning because it can dynamically predict each sample's saliency map's RBF parameters ($[c_z, c_t, \sigma]$) given the current subnetwork vector $\mathbf{v}$ with a single forward pass. In contrast, optimization-based explanation methods [46, 39] need to fit a new model, and perturbation-based methods [65, 69, 71] have to make multiple forward passes for the newly selected subnetwork to obtain its explanations. Therefore, they are inefficient to achieve the same goal. We provide the detailed parameterization of channels ($g(\cdot)$) and $\mathcal{R}_{res}$ in supplementary materials.

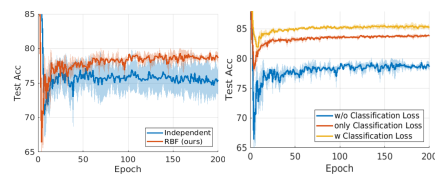
4 Experiments

We use CIFAR-10 [29] and ImageNet [10] to validate the effectiveness of our proposed model. Due to the space limit, we refer to supplementary for details of our experimental setup. We call our method ISP (**I**nterpretations **S**teered **P**runing) in the experiments.

4.1 Analysis of Different Settings

Before we formally present our experimental results compared to competitive methods, we study the effect of different design choices for our model's components on its performance. We keep the resource regularization ($\mathcal{R}_{res}$) term in obj. 10 and add/drop other ones in all settings.

In our first experiment, we explore the impact of γ_1 by only using interpretations (second term of Obj. 10) to steer the pruning. Fig. 5(a,b) and Fig. 4(a) demonstrate the results. We can observe in Fig. 5(a,b) that small γ_1 values (*e.g.,* 0.1) result in a weaker supervision signal from the interpretations and make the exploration of subnetworks unstable (showing high variance), whereas larger ones make the training smooth. Fig. 4(a) illustrates the influence of RBF/independent masks' parameterization scheme in Eq. 5 (ours)/Eq. 3 (REAL-X [26]). Our RBF-style model brings better performance than independent parameterization. The latter becomes unstable and less effective when the training proceeds. The instability happens possibly because the pruning gets trapped in some local minima due to noisy and unstructured masks. We can also observe that interpretations on their own provide stable and efficient signals for pruning.

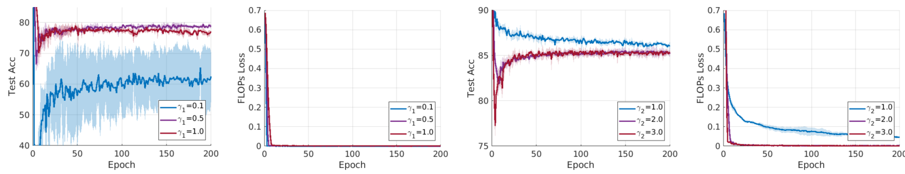


(a) (b)

Fig. 4: **(a):** Test accuracy of different masks' parameterization schemes. (RBF (ours) *vs.* Independent (REAL-X [26])) **(b):** Test accuracy w/wo using the classification loss. All results are for 3 run times with ResNet-56 on CIFAR-10. Shaded areas represent variance.

In our second experiment, we examine the impact of γ_2 while utilizing all three terms in objective 10 for pruning. Fig. 5(c, d) indicates that small γ_2 (*e.g.,* 1.0) shows higher accuracy but may not be able to push the FLOPs regularization to 0, *i.e.*, reach the predefined pruning rate p. Larger values can satisfy the resource constraint while showing acceptable performance.

Finally, we examine the performance of different combinations of components in objective 10. The results are available in Fig. 4(b). Specifically, 'w/o Classification Loss' represents using the second and third terms, 'only Classification Loss' indicates using the first and third ones, and 'w Classification Loss' means using the full objective function. It is plausible that 'only Classification Loss' performs better than only interpretations ('w/o Classification Loss') since the loss function is a 'less noisy' signal for accuracy compared to the interpretations.



(a) γ_1: accuracy (b) γ_1: $\mathcal{R}_{res}$ loss (c) γ_2: accuracy (d) γ_2: $\mathcal{R}_{res}$ loss

Fig. 5: **(a), (b):** The model's test accuracy and the FLOPs regularization term when changing γ_1, and **(c), (d):** when varying γ_2. All results are run for 3 times with ResNet-56 on CIFAR-10. Shaded areas represent variance.

Table 1: Comparison of results on CIFAR-10. Δ-Acc represents the performance changes relative to the baseline, and +/− indicates an increase/decrease, respectively.

Model	Method	Baseline Acc	Pruned Acc	Δ-Acc	Pruned FLOPs
ResNet-56	DCP-Adapt [70]	93.80%	93.81%	+0.01%	47.0%
	SCP [28]	93.69%	93.23%	−0.46%	51.5%
	FPGM [19]	93.59%	92.93%	−0.66%	52.6%
	SFP [18]	93.59%	92.26%	−1.33%	52.6%
	FPC [17]	93.59%	93.24%	−0.25%	52.9%
	HRank [34]	93.26%	92.17%	−0.09%	50.0%
	EEMC [66]	93.62%	93.68%	+0.06%	**56.0%**
	ISP (ours)	93.56%	**93.74%**	+ **0.18%**	54.0%
MobileNetV2	Uniform [70]	94.47%	94.17%	−0.30%	26.0%
	DCP [70]	94.47%	94.69%	+0.22%	26.0%
	ISP (ours)	94.53%	**94.85%**	+ **0.32%**	**44.0%**

Furthermore, incorporating interpretations enhances the supervision signal and yields the best performance. This observation indicates that interpretations contain guidance from different perspectives complementary to the classification loss that only focuses on the model's outputs.

4.2 Comparasion Results

CIFAR-10 Results: Tab. 1 summarizes the results on CIFAR-10. For **ResNet-56**, ISP outperforms baselines with a similar FLOPs pruning rate. It has a pruning rate on par with EEMC [66], the most recent baseline, while it shows higher Δ-Acc (+0.18% *vs.* +0.06%). For **MobileNet-V2**, ISP simultaneously prunes 18% more FLOPs than DCP and Uniform. It also achieves a better accuracy improvement (+0.10% higher Δ-Acc) than DCP.

ImageNet Results: We present the results on ImageNet in Tab. 2. For **ResNet-34**, ISP achieves the best trade-off between the performance and FLOPs reduction. It achieves Δ Top-1 close to Taylor [41], but ISP can prune 19.8% more FLOPs. Also, with similar FLOPs pruning rate, ISP outperforms FPGM [19] by 0.84% Δ Top-1. For **ResNet-50**, our model can achieve the largest pruning rate, 56.6%, with the best Δ Top-1/Top-5 being $-0.16\%/-0.12\%$ showing 0.33%/0.23% improvement compared to EEMC [66]. For **ResNet-101**, ISP is the only method that its pruned network has better accuracy than the original

Table 2: Comparison results on ImageNet with ResNet-34/50/101 and MobileNet-V2.

Model	Method	Baseline Top-1 Acc	Baseline Top-5 Acc	Δ-Acc Top-1	Δ-Acc Top-5	Pruned FLOPs
ResNet-34	FPGM [19]	73.92%	91.62%	-1.29%	-0.54%	41.1%
	Taylor [41]	73.31%	-	-0.48%	-	24.2%
	ISP (ours)	73.31%	91.42%	$\mathbf{-0.45\%}$	$\mathbf{-0.40\%}$	**44.0%**
ResNet-50	DCP [70]	76.01%	92.93%	-1.06%	-0.61%	55.6%
	CCP [42]	76.15%	92.87%	-0.94%	-0.45%	54.1%
	FPGM [19]	76.15%	92.87%	-1.32%	-0.55%	53.5%
	ABCP [35]	76.01%	92.96%	-2.15%	-1.27%	54.3%
	QI [2]	74.90%	92.10%	-1.31%	-0.27%	50.0%
	PFP [33]	76.13%	92.86%	-0.92%	-0.45%	44.0%
	EEMC [66]	76.15%	92.87%	-0.49%	-0.35%	56.0%
	ISP (ours)	76.13%	92.86%	$-$ **0.16%**	$-$ **0.12%**	**56.6%**
ResNet-101	FPGM [19]	77.37%	93.56%	-0.05%	0.00%	41.1%
	Taylor [41]	77.37%	-	-0.02%	-	39.8%
	QI [2]	76.40%	92.80%	-2.31%	-0.86%	50.0%
	PFP [33]	77.37%	93.56%	-0.94%	-0.44%	45.1%
	ISP (ours)	77.37%	93.56%	+ **0.40%**	+ **0.22%**	**56.8%**
MobileNet-V2	Uniform [49]	71.80%	91.00%	-2.00%	1.40%	30.0%
	AMC [20]	71.80%	-	-1.00%	-	30.0%
	CC [31]	71.88%	-	-0.97%	-	28.3%
	MetaPruning [36]	72.00%	-	-0.80%	-	**30.7%**
	ISP (ours)	71.88%	90.29%	$-$ **0.15%**	$-$ **0.08%**	29.0%

one. Also, it accomplishes the highest pruning rate, 56.8%, with a significant 11.7% gap with PFP [33]. For **MobileNetV2**, ISP has a pruning rate competitive (+29.0% *vs.* +30.7%) to MetaPruning [36] and reaches the highest Δ Top-1, with a 0.65% margin with MetaPruning. We also note that ISP significantly outperforms QI [2] (in terms of both accuracy improvement and pruning rate for ResNet-50/101) that aims to perform interpretable pruning by finding filters that are aligned with visual concepts, which illustrates the superiority of our proposed AEM model for pruning compared to other interpretation techniques.

5 Conclusions

We proposed a novel neural network pruning method that utilizes interpretations of the model as guidance for its pruning procedure. We showed that Amortized Explanation Models (AEM) are suitable for our purpose as they can provide real-time explanations of a model. We empirically showed that explanation masks of REAL-X [26], state-of-the-art AEM, might lack a meaningful structure and not be interpretable. Thus, we introduced a new AEM model that overcomes this problem by respecting the geometric prior of natural images and finding the optimal functional form over pixel's Bernoulli parameters of explanatory masks in the RBF functions' family. Finally, we leverage the predictions of our AEM model to steer the pruning process in our formulation. Our experimental results on benchmark data demonstrate that the interpretations of a parameter-heavy classifier provide valuable information to steer its pruning process, complementing the guidance from its outputs, which are the main focus of previous methods.

Acknowledgement

This work was partially supported by NSF IIS 1845666, 1852606, 1838627, 1837956, 1956002, 2217003.

References

1. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018)
2. Alqahtani, A., Xie, X., Jones, M.W., Essa, E.: Pruning cnn filters via quantifying the importance of deep visual representations. Computer Vision and Image Understanding **208**, 103220 (2021)
3. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one **10**(7), e0130140 (2015)
4. Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016)
5. Bronstein, M.M., Bruna, J., Cohen, T., Velickovic, P.: Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. CoRR **abs/2104.13478** (2021), https://arxiv.org/abs/2104.13478
6. Chen, J., Song, L., Wainwright, M., Jordan, M.: Learning to explain: An information-theoretic perspective on model interpretation. In: International Conference on Machine Learning. pp. 883–892. PMLR (2018)
7. Chen, W., Wilson, J., Tyree, S., Weinberger, K., Chen, Y.: Compressing neural networks with the hashing trick. In: International conference on machine learning. pp. 2285–2294 (2015)
8. Chin, T.W., Ding, R., Zhang, C., Marculescu, D.: Towards efficient model compression via learned global ranking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1518–1528 (2020)
9. Dabkowski, P., Gal, Y.: Real time image saliency for black box classifiers. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. pp. 6967–6976 (2017)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
11. Gao, S., Huang, F., Pei, J., Huang, H.: Discrete model compression with resource constraint for deep neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1899–1908 (2020)
12. Gao, S., Huang, F., Zhang, Y., Huang, H.: Disentangled differentiable network pruning. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022)
13. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
14. Grathwohl, W., Wang, K., Jacobsen, J., Duvenaud, D., Norouzi, M., Swersky, K.: Your classifier is secretly an energy based model and you should treat it like one. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020), https://openreview.net/forum?id=Hkxzx0NtDB
15. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. In: Advances in neural information processing systems. pp. 1135–1143 (2015)

16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
17. He, Y., Ding, Y., Liu, P., Zhu, L., Zhang, H., Yang, Y.: Learning filter pruning criteria for deep convolutional neural networks acceleration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2009–2018 (2020)
18. He, Y., Kang, G., Dong, X., Fu, Y., Yang, Y.: Soft filter pruning for accelerating deep convolutional neural networks. In: International Joint Conference on Artificial Intelligence (IJCAI). pp. 2234–2240 (2018)
19. He, Y., Liu, P., Wang, Z., Hu, Z., Yang, Y.: Filter pruning via geometric median for deep convolutional neural networks acceleration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4340–4349 (2019)
20. He, Y., Lin, J., Liu, Z., Wang, H., Li, L.J., Han, S.: Amc: Automl for model compression and acceleration on mobile devices. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 784–800 (2018)
21. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
22. Hooker, S., Erhan, D., Kindermans, P.J., Kim, B.: A benchmark for interpretability methods in deep neural networks. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019)
23. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
24. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 448–456. PMLR, Lille, France (07–09 Jul 2015), https://proceedings.mlr.press/v37/ioffe15.html
25. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017), https://openreview.net/forum?id=rkE3y85ee
26. Jethani, N., Sudarshan, M., Aphinyanaphongs, Y., Ranganath, R.: Have we learned to explain?: How interpretability methods can learn to encode predictions in their interpretations. In: International Conference on Artificial Intelligence and Statistics. pp. 1459–1467. PMLR (2021)
27. Joyce, J.M.: Kullback-leibler divergence. International encyclopedia of statistical science **720**, 722 (2011)
28. Kang, M., Han, B.: Operation-aware soft channel pruning using differentiable masks. International Conference on Machine Learning (2020)
29. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
30. Li, H., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. ICLR (2017)
31. Li, Y., Lin, S., Liu, J., Ye, Q., Wang, M., Chao, F., Yang, F., Ma, J., Tian, Q., Ji, R.: Towards compact cnns via collaborative compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6438–6447 (2021)

32. Liang, T., Glossner, J., Wang, L., Shi, S., Zhang, X.: Pruning and quantization for deep neural network acceleration: A survey. Neurocomputing **461**, 370–403 (2021)
33. Liebenwein, L., Baykal, C., Lang, H., Feldman, D., Rus, D.: Provable filter pruning for efficient neural networks. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=BJxkOlSYDH
34. Lin, M., Ji, R., Wang, Y., Zhang, Y., Zhang, B., Tian, Y., Shao, L.: Hrank: Filter pruning using high-rank feature map. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
35. Lin, M., Ji, R., Zhang, Y., Zhang, B., Wu, Y., Tian, Y.: Channel pruning via automatic structure search. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI). pp. 673 – 679 (2020)
36. Liu, Z., Mu, H., Zhang, X., Guo, Z., Yang, X., Cheng, K.T., Sun, J.: Metapruning: Meta learning for automatic neural network channel pruning. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3296–3305 (2019)
37. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: ICCV (2017)
38. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
39. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Proceedings of the 31st international conference on neural information processing systems. pp. 4768–4777 (2017)
40. Maddison, C.J., Mnih, A., Teh, Y.W.: The concrete distribution: A continuous relaxation of discrete random variables. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017), https://openreview.net/forum?id=S1jE5L5gl
41. Molchanov, P., Mallya, A., Tyree, S., Frosio, I., Kautz, J.: Importance estimation for neural network pruning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11264–11272 (2019)
42. Peng, H., Wu, J., Chen, S., Huang, J.: Collaborative channel pruning for deep networks. In: International Conference on Machine Learning. pp. 5113–5122 (2019)
43. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In: European conference on computer vision. pp. 525–542. Springer (2016)
44. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
45. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
46. Ribeiro, M.T., Singh, S., Guestrin, C.: " why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1135–1144 (2016)
47. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
48. Sabih, M., Hannig, F., Teich, J.: Utilizing explainable ai for quantization and pruning of deep neural networks. arXiv preprint arXiv:2008.09072 (2020)
49. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)

50. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision. pp. 618–626 (2017)
51. Shah, H., Jain, P., Netrapalli, P.: Do input gradients highlight discriminative features? Advances in Neural Information Processing Systems **34** (2021)
52. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: International Conference on Machine Learning. pp. 3145–3153. PMLR (2017)
53. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. In: In Workshop at International Conference on Learning Representations. Citeseer (2014)
54. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp. 568–576 (2014)
55. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1409.1556
56. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017)
57. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.A.: Striving for simplicity: The all convolutional net. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings (2015), http://arxiv.org/abs/1412.6806
58. Srinivas, S., Fleuret, F.: Rethinking the role of gradient-based attribution methods for model interpretability. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021), https://openreview.net/forum?id=dYeAHXnpWJ4
59. Sui, Y., Yin, M., Xie, Y., Phan, H., Aliari Zonouz, S., Yuan, B.: Chip: Channel independence-based pruning for compact neural networks. Advances in Neural Information Processing Systems **34**, 24604–24616 (2021)
60. Wen, W., Wu, C., Wang, Y., Chen, Y., Li, H.: Learning structured sparsity in deep neural networks. In: Advances in neural information processing systems. pp. 2074–2082 (2016)
61. Yao, K., Cao, F., Leung, Y., Liang, J.: Deep neural network compression through interpretability-based filter pruning. Pattern Recognition **119**, 108056 (2021)
62. Ye, M., Gong, C., Nie, L., Zhou, D., Klivans, A., Liu, Q.: Good subnetworks provably exist: Pruning via greedy forward selection. International Conference on Machine Learning (2020)
63. Yeom, S.K., Seegerer, P., Lapuschkin, S., Binder, A., Wiedemann, S., Müller, K.R., Samek, W.: Pruning by explaining: A novel criterion for deep neural network pruning. Pattern Recognition **115**, 107899 (2021)
64. Yoon, J., Jordon, J., van der Schaar, M.: Invase: Instance-wise variable selection using neural networks. In: International Conference on Learning Representations (2018)
65. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision. pp. 818–833. Springer (2014)
66. Zhang, Y., Gao, S., Huang, H.: Exploration and estimation for model compression. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 487–496 (2021)

67. Zhang, Y., Gao, S., Huang, H.: Recover fair deep classification models via altering pre-trained structure. In: Proceedings of the European Conference on Computer Vision (ECCV) (2022)
68. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2921–2929 (2016)
69. Zhou, J., Troyanskaya, O.G.: Predicting effects of noncoding variants with deep learning–based sequence model. Nature methods **12**(10), 931–934 (2015)
70. Zhuang, Z., Tan, M., Zhuang, B., Liu, J., Guo, Y., Wu, Q., Huang, J., Zhu, J.: Discrimination-aware channel pruning for deep neural networks. In: Advances in Neural Information Processing Systems. pp. 875–886 (2018)
71. Zintgraf, L.M., Cohen, T.S., Adel, T., Welling, M.: Visualizing deep neural network decisions: Prediction difference analysis. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017), https://openreview.net/forum?id=BJ5UeU9xx