

# Evaluating Feature Importance Estimates

Sara Hooker<sup>1,2</sup> Dumitru Erhan<sup>1</sup> Pieter-Jan Kindermans<sup>1,2</sup> Been Kim<sup>1</sup>

## Abstract

Interpretability methods should be *both* meaningful to a human and correctly explain model behavior. In this work, we propose a benchmark to evaluate the latter. We introduce **ROAR**, **RemOve And Retrain**, a formal measure of the relative accuracy of interpretability methods that estimate feature importance in deep neural networks. We evaluate commonly used interpretability methods and a set of recently proposed ensemble-based derivative approaches. Our results across several large-scale image classification datasets are consistent and thought-provoking – we find that the formal methods we consider produce estimates that are less accurate or on par with a random designation of feature importance. However, certain derivative approaches that ensemble these estimates far outperform such a random guess. The manner of ensembling remains critical, we show that some approaches do no better than the underlying method but carry a far higher computational burden.

## 1. Introduction

In a machine learning setting, a question of great interest is estimating the influence of a given input feature on the prediction made by a model. Understanding what input features are important helps improve our models, builds trust in the model prediction and isolates undesirable behavior. For certain areas such as healthcare, autonomous vehicles and credit scoring, the need for such interpretability goes beyond a “nice-to-have.” In these sensitive domains, estimates of feature importance must be both 1) meaningful to a human and 2) highly accurate, as an incorrect explanation of model behavior may have intolerable costs on human welfare.

In this work, we are concerned with 2). We propose a

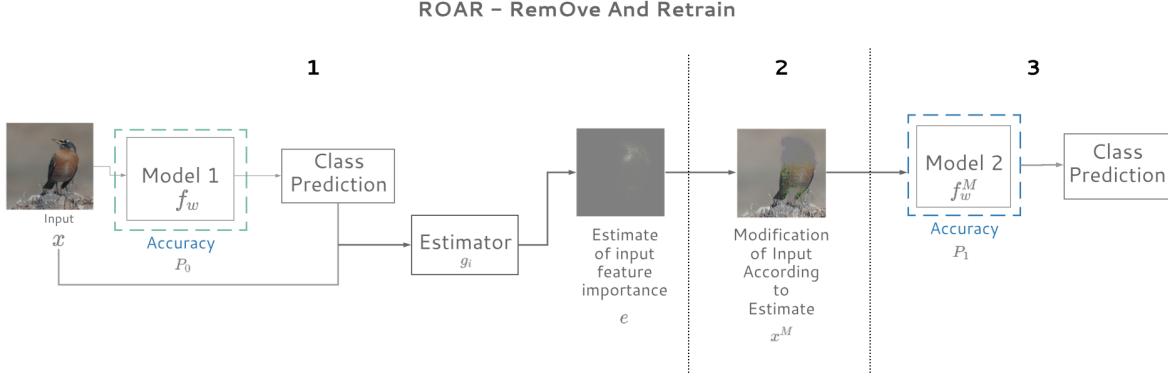
<sup>1</sup>Google Brain <sup>2</sup>Work done as part of the Google Brain AI Residency program.. Correspondence to: Sara Hooker <shooker@google.com>.

formal methodology to evaluate the accuracy of commonly used feature importance estimators for deep neural networks (DNNs). DNNs pose unique challenges for the estimation of input feature importance, as well as work such as ours that considers whether the estimate produced is reliable. This is due to both the non-linear activations present in DNNs and the large number of input features often involved in tasks where DNNs are used.

Due to this high dimensional input space, there has been limited but important work that estimates feature importance across all possible data points (Koh & Liang, 2017). Instead, numerous methods have been proposed (Baehrens et al., 2010; Bach et al., 2015; Zintgraf et al., 2017; Selvaraju et al., 2017; Sundararajan et al., 2017; Simonyan & Zisserman, 2015; Zeiler & Fergus, 2014; Springenberg et al., 2015; Kindermans et al., 2017; Montavon et al., 2017; Fong & Vedaldi, 2017; Dabkowski & Gal, 2017; Zhang et al., 2016; Shrikumar et al., 2017; Zhou et al., 2014; Ross & Doshi-Velez, 2017) which constrain ranking to the set of input features associated with a single image. These estimators produce a score for each pixel that reflects the estimated contribution to the model prediction for that image. The magnitude of the score can then be used to rank and compare the importance of all input features. More recent work (Smilkov et al., 2017; Adebayo et al., 2018a) has proposed derivative approaches that ensemble a set of estimates. These ensemble methods are often considered more appealing because they produce a “visually sharper” explanation of model behavior for cases where the scores are visualized as a natural image “heatmap.” (see Fig. 2 for a visual comparison of base vs. ensemble estimators).

However, it is challenging to evaluate whether this explanation of model behavior is reliable. If we knew what were important to the model, we would not need to estimate feature importance in the first place. Instead, in this work, we propose a measure that evaluates the approximate accuracy of the feature importance ranking according to the hypothesis that a more accurate ranking will identify a subset of features as important whose removal will degrade model performance the most.

We term this measure **ROAR**, **RemOve And Retrain**. For each estimator, ROAR replaces a fraction of all pixels that are estimated to be most important with a constant value



**Figure 1.** ROAR evaluates the relative accuracy of feature importance estimators. **1)** An interpretability method ranks the importance of each pixel to the model prediction. **2)** Use this ranking by removing a fraction of input features estimated to be most important from each image in the dataset. **3)** Train a new model on the modified inputs and measure the degradation to model performance. The most accurate estimator will identify as important pixels in the image whose modification causes the most degradation to model performance upon retraining.

that is irrelevant for the classification task. This modification (shown in Fig. 2) is repeated for each image in both the training and test set. To measure the change to model behavior subsequent to the removal of these input features, we separately train new models on the altered dataset and the original unmodified images. An approximately accurate estimator will identify as important input pixels those whose subsequent removal causes the sharpest degradation in accuracy. In Fig. 1, we illustrate the key steps in the ROAR framework.

Training a new model (from random initialization) is *crucial* in order for the constant value for which we replaced the input to be considered “uninformative.” Without retraining, it is difficult to decouple whether the model’s degradation in performance is due to the replacement value being outside of the training data manifold or due to the accuracy of the estimate. Model vulnerability to the introduction of “new evidence” has already been widely acknowledged (Dabkowski & Gal, 2017; Fong & Vedaldi, 2017).

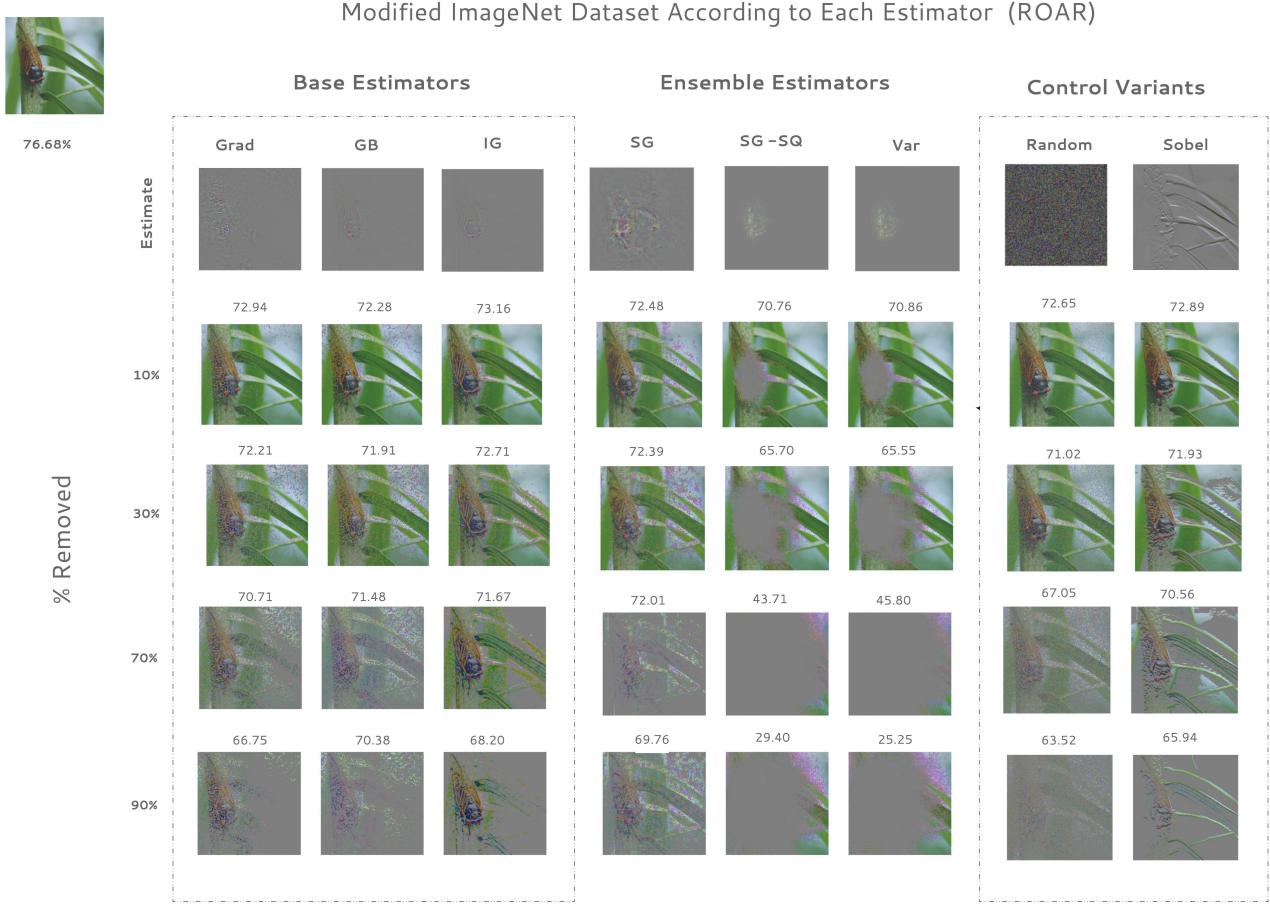
In addition to comparing the approximate accuracy of a set of estimators, we also compare estimator performance to a *random* assignment of importance and the mask produced by applying a sobel edge filter to the image. Both of these control variants produce rankings that are independent of the properties of the model we aim to interpret. Given that these methods do not depend upon the model (the sobel edge detector depends only on the input image, whereas the random estimator is independent of both model and data), the performance of these variants represent a lower bound of accuracy that an estimator could be expected to achieve. In particular, the random baseline allows us to answer the question: *is the estimator more accurate than a random guess as to which features are important?*

In a broad set of experiments across three large scale, open source image datasets—ImageNet (Deng et al., 2009), Food 101 (Bossard et al., 2014) and Birdsnap (Berg et al., 2014)—our results are consistent and thought-provoking:

- Without ensembling, the interpretability methods that we evaluate are **no better or on par with** a random assignment of importance. However, we show that certain derivative approaches that ensemble sets of these estimates **far outperform** both the underlying method and such a random guess.
- The choice of ensembling approach is paramount. Ensemble method performance is varied. SmoothGrad-Squared (unpublished variant of Classic SmoothGrad) and Vargrad (Adebayo et al., 2018a) produced large gains in accuracy, while Classic SmoothGrad (Smilkov et al., 2017)), is less accurate or on par with a single estimate but carries a far higher computational burden.
- Finally, we show training performance proves surprisingly robust to random modification of the majority of all input features. For example, after randomly replacing 90% of all ImageNet input features, we can still train a model that achieves  $63.53 \pm 0.13$  (average across 5 independent runs). These results suggest that many redundancies in the input feature space exist, however, the basic estimators that we consider are no better than a random guess at identifying them.

## 2. Related Work

Interpretability research is diverse, and many different approaches are used to gain intuition about the function implemented by a neural network. For example, one can distill or



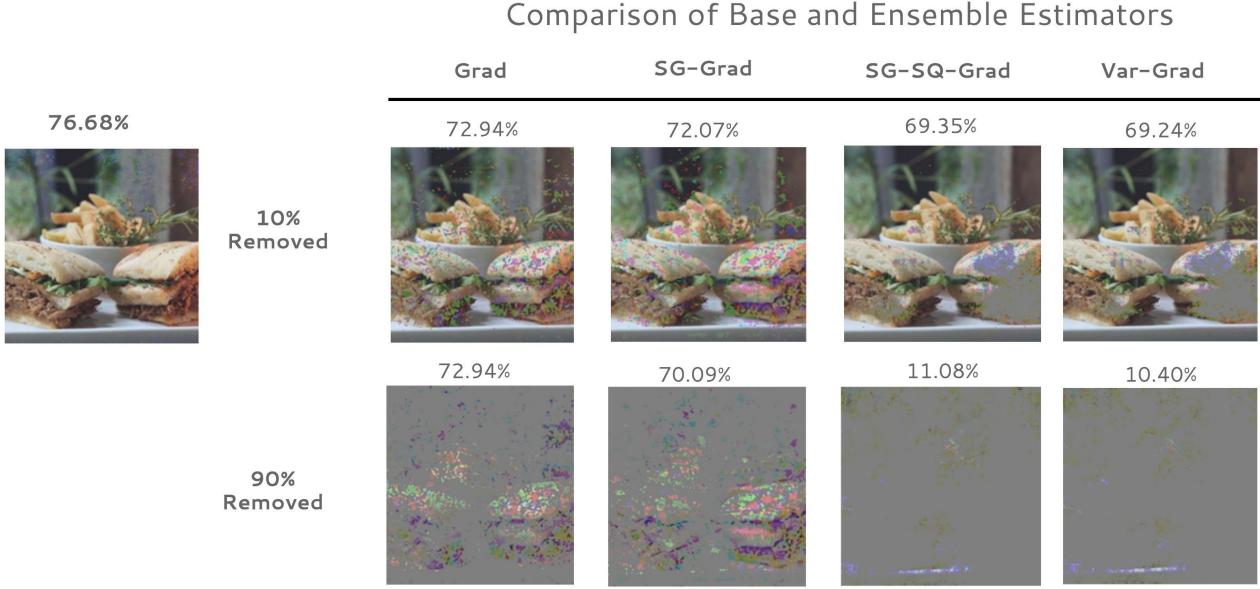
**Figure 2.** A single ImageNet image modified according to the ROAR framework. The fraction of pixels estimated to be most important by each interpretability method is replaced with the mean. Above each image, we include the average test-set accuracy for 5 ResNet-50 models independently trained on the modified dataset. **From left to right:** base estimators (gradient heatmap (GRAD), Integrated Gradients (IG), Guided Backprop (GB)), derivative approaches that ensemble a set of estimates (SmoothGrad Integrated Gradients (SG-SQ-IG), SmoothGrad-Squared Integrated Gradients (SG-SQ-IG), VarGrad Integrated Gradients (VAR-IG)) and control variants (random modification (RANDOM) and a sobel edge filter (SOBEL)). This image is best visualized in digital format.

constrain a model into a functional form that is considered more interpretable (Ba & Caruana, 2014; Frosst & Hinton, 2017; Wu et al., 2017; Ross et al., 2017). Other methods explore the role of neurons or activations in hidden layers of the network (Olah et al., 2017; Raghu et al., 2017; Morcos et al., 2018; Zhou et al., 2018), while others use high level concepts to explain prediction results (Kim et al., 2018), and finally there are also the input feature importance estimators that we evaluate in this work. These interpretability methods estimate the importance of an input feature to a specified output activation.

Without a clear way to measure the "correctness" of a feature importance estimate, comparing the relative merit of different estimators is often based upon human studies (Selvaraju et al. 2017; Ross & Doshi-Velez 2017; Lage et al. 2018

and many others) which interrogate whether the ranking is meaningful to a human. However, an explanation considered "trustworthy" does not guarantee that the same reliably explains model behavior. It has already been shown that the level of human trust in a system is decoupled from the actual performance of the algorithm (Poursabzi-Sangdeh et al., 2018; Dietvorst et al., 2014).

Recently, there has been limited but important work on frameworks to evaluate whether interpretability methods are *both* reliable and meaningful. Kindermans et al. (2017) define a unit test that constructs a narrow ground-truth in which invariance to factors that do not affect the model can be measured. Adebayo et al. (2018b) consider a set of sanity checks that measure the change to an estimate as parameters in a model or dataset labels are randomized.



**Figure 3.** A comparison of the modification of a single image from the Food 101 dataset according to a base estimator (gradient heatmap GRAD) and associated ensemble approaches (SmoothGrad Grad (SG-GRAD), SmoothGrad-Squared Grad (SG-SQ-GRAD), VarGrad Grad (SG-SQ-GRAD)). For each image, a fraction of pixels estimated to be most important by each estimator is replaced with the mean. Above each image, we include the average test-set accuracy for 5 ResNet-50 models independently train on the modified dataset. Recently proposed ensemble approaches, VarGrad and SmoothGrad-Squared, significantly improve the approximate accuracy of the estimator (removing the inputs considered most important according to these estimators degrades performance far more than a random selection). However, certain approaches like SmoothGrad require far more computation and are worse than a random ranking of importance.

Most relevant to our work are modification based evaluation measures proposed originally by Samek et al. (2017) with subsequent variations (Ancona et al., 2017; Fong & Vedaldi, 2017; Kindermans et al., 2017). In this line of work, one replaces the inputs estimated to be most important with a value considered meaningless to the task. These methods measure the subsequent degradation to the trained model at inference time.

To the best of our knowledge, unlike prior modification based evaluation measures, our benchmark requires retraining the model from random initialization on the modified dataset rather than re-scoring the modified image at inference time. Without this step, one cannot decouple whether the model’s degradation in performance is due to artifacts introduced by the value used to replace the pixels that are removed or due to the approximate accuracy of the estimator. We discuss this further in section 3.3, supported by large-scale experiments on ImageNet.

We do not modify a connected region, or patch, of connected pixels according to the aggregated estimates of importance. Instead, we simply modify the fraction of inputs estimated to be most important. Finally, we moreover modify every image in ImageNet (1.28 million training and 50,000 validation images), Birdsnap (47,386 training and 2,443

validation images) and Food 101 (75,750 million training and 25,250 validation images). All prior evaluations have involved a far smaller subset of data and the consideration of a single dataset.

### 3. Estimating Input Feature Importance

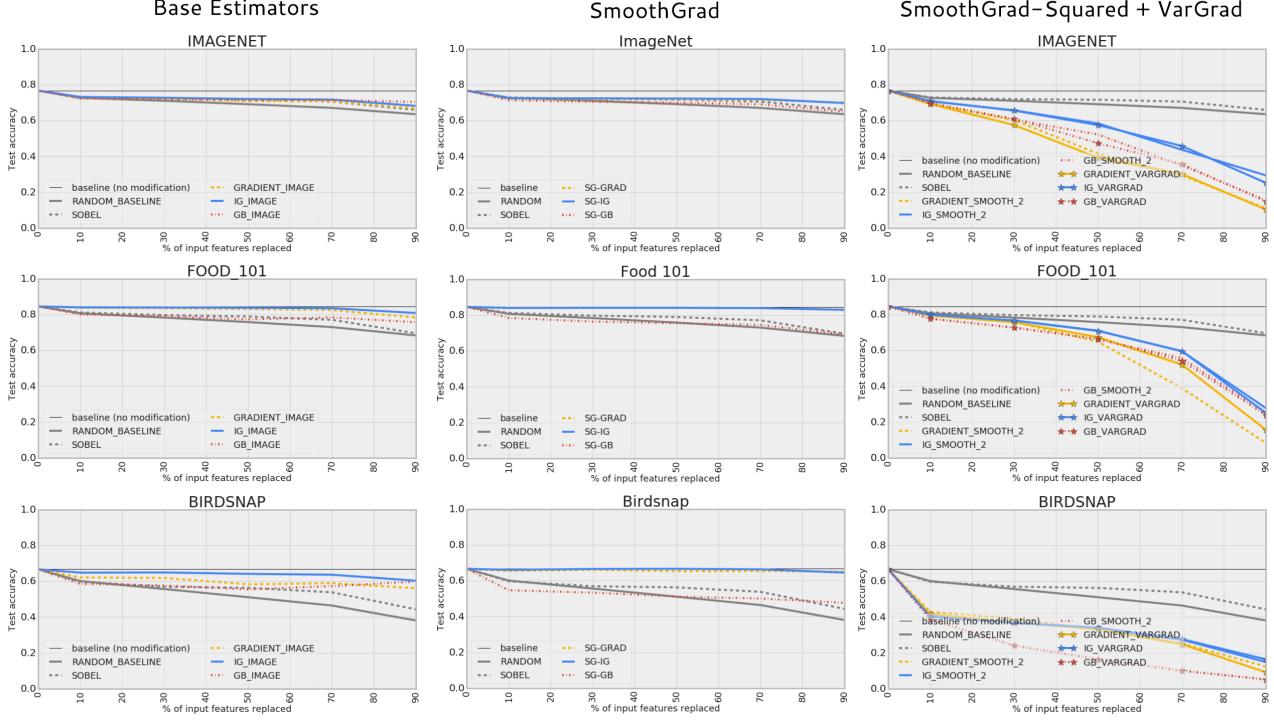
A CNN is trained to approximate the function  $F$  that maps an input variable  $X$  to an output variable  $Y$ , formally  $F : X \mapsto Y$ . Without loss of generality we represent the image input as a feature vector  $\mathbf{x} \in R^N$ .  $y \in Y$  is a discrete label vector associated with each input  $x$ . A given input image  $\mathbf{x}$  can be decomposed into a set of pixels  $\{x_i\}_{i=1}^N$ .

An estimator  $\mathcal{G}$  produces a vector of estimates  $\mathbf{e}$ .  $e_i \in \mathbf{e}$  is the estimated importance of  $x_i$  to an output activation  $A_n^l$ , where  $l$  and  $n$  designate the layer of the model and the neuron of interest respectively.  $A_n^l$  is typically specified to be the maximum pre-softmax score or the softmax probability.

#### 3.1. Evaluation Methodology

We can rank  $\mathbf{e}$  into an ordered list  $\{e_i^o\}_{i=1}^N$  so that  $e_0^o$  corresponds to the input feature  $x_i$  estimated to be most important. For a fraction  $t$  of this ordered set, we replace the corresponding values in the raw image vector  $\mathbf{x}$  with

## Only Certain Ensemble Approaches Benefit Performance



**Figure 4. Left inset:** Grad (GRAD), Integrated Gradients (IG) and Guided Backprop (GB) perform worse than a random assignment of feature importance. **Middle inset:** Surprisingly, we find that SmoothGrad (SG), a more computationally intensive ensemble approach that requires the generation of a set of estimates, is less accurate than a random assignment of importance and often worse than a single estimate (in the case of raw gradients SG-GRAD and Integrated Gradients SG-IG). **Right inset:** In contrast to SmoothGrad, certain ensemble methods, such as SmoothGrad Squared (SG-SQ) and VarGrad (VAR) produce a dramatic improvement in approximate accuracy and far outperform such random guess across all datasets considered. Applying these estimators benefit the performance of all methods.

a constant uninformative variable  $c$ . We create a family of distributions  $p(y, x^M; t_i, g_i)$ , where each distribution is defined by incrementally increasing the fraction of inputs modified  $t = [0 : 0.1 : 1]$  and varying the estimator  $g_i \subset \mathcal{G}$ .

When  $t = 1$ ,  $x^M = x$  and the test-set accuracy  $\xi(x^M|g_i)$ , will only differ from  $\xi(x)$  of a model trained on unmodified inputs by an epsilon term  $\varepsilon$  that is caused by the natural variation in training performance. When  $t = 0$ , we have replaced all input features with the constant value  $c$  and learning a representation should not be possible.

In between  $t = [0 : .1 : 1]$  we are unable to precisely determine how removing inputs will change the test-set accuracy  $\xi$  since we do not know the true distribution of importance a priori. However, we can compare the degradation of test-set accuracy between estimators for the same fraction  $t$ .

ROAR evaluates estimators according to the hypothesis that the most approximately accurate estimator  $g_i^*$  will identify a

subset of features as important whose removal will degrade model performance the most.. Thus, the most desirable estimator is the one that results in the lowest test-set accuracy  $\xi(x^M|g^*)$ , where  $x^M$  is the modified dataset given the estimator  $g$ :

$$\xi(x^M|g^*) = \min_{g \subset \mathcal{G}} \xi(x^M|g)$$

In addition, we determine an estimate  $g$  to be *better than a random assignment of importance*  $g^R$  if the test accuracy trained on the randomly modified inputs is such that:

$$\xi(x^M|g) < \xi(x^M|g^R)$$

### 3.2. Estimators Considered

In this work, our initial evaluation is constrained to a subset of estimators  $\{g_i\} \subset \mathcal{G}$  which we briefly introduce below. We selected this subset based upon the avail-

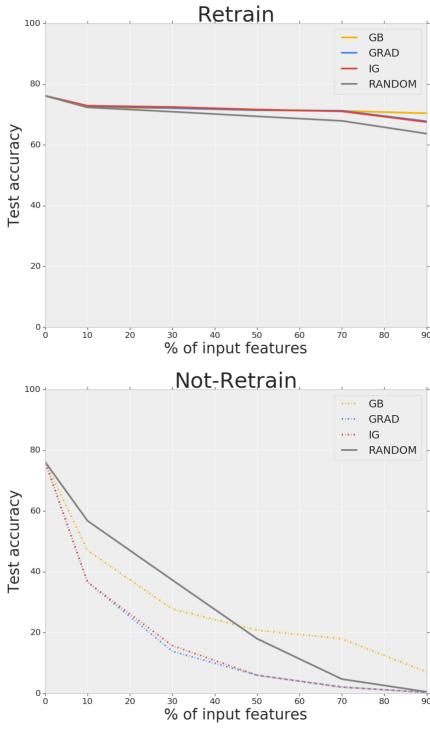


Figure 5. RETRAIN is a more rigorous benchmark because it is better able to decouple the performance of the interpretability method from the degradation caused by the modification itself. This can be seen by comparing accuracy degradation between a model not retrained on the modified inputs (NO-RETRAIN) and a model that is trained from random initialization on the modified inputs (RETRAINED). A model that is not retrained presents far higher accuracy degradation for all modification thresholds. In this case, inference is done on a different distribution than the model is trained on. It is impossible to decouple the evaluation of the method from the introduction of artefacts.

ability of open source code and the ease of implementation on a ResNet-50 architecture (He et al., 2015). We welcome the opportunity to consider additional estimators in the future, and in order to make it easy to apply ROAR to additional estimators we have open sourced our code <https://bit.ly/2tLLZB>. We briefly introduce each grouping of estimators below.

### 3.2.1. BASE ESTIMATORS

**Gradients or Sensitivity heatmaps** (Simonyan & Zisserman, 2015; Baehrens et al., 2010) (**GRAD**) are the gradient of the output activation of interest  $A_n^l$  with respect to  $x_i$ :

$$\mathbf{e} = \frac{\partial A_n^l}{\partial x_i}$$

**Guided Backprop** (Springenberg et al., 2015) (**GB**) is an example of a signal method. Signal estimators aim to visualize the input patterns that cause the neuron activation  $A_n^l$  in higher layers (Springenberg et al., 2015; Zeiler & Fergus, 2014; Kindermans et al., 2017). GB computes this by using a modified backpropagation step that stops the flow of gradients when less than zero at a ReLu gate.

**Integrated Gradients** (Sundararajan et al., 2017) (**IG**) is an example of an attribution method. Attribution estimators assign importance to input features by decomposing the output activation  $A_n^l$  into contributions from the individual input features (Bach et al., 2015; Sundararajan et al., 2017; Montavon et al., 2017; Shrikumar et al., 2016; Kindermans et al., 2017). Attribution methods require that all contributions sum to the activation of interest. This property is often termed *completeness*. Integrated gradients interpolate a set of estimates for values between a non-informative reference point  $\mathbf{x}^0$  to the actual input  $\mathbf{x}$ . This integral can be approximated by summing a set of  $k$  points at small intervals between  $\mathbf{x}^0$  and  $\mathbf{x}$ :

$$\mathbf{e} = (\mathbf{x}_i - \mathbf{x}_i^0) \times \sum_{i=1}^k \frac{\partial f_w(\mathbf{x}^0 + \frac{i}{k}(\mathbf{x} - \mathbf{x}^0))}{\partial \mathbf{x}_i} \times \frac{1}{k}$$

The final estimate  $\mathbf{e}$  will depend upon both the choice of  $k$  and the reference point  $\mathbf{x}^0$ . As suggested by Sundararajan et al. (2017), we use a black image as the reference point and set  $k$  to be 25.

### 3.2.2. DERIVATIVE APPROACHES THAT ENSEMBLE A SET OF ESTIMATES

An example of a single image modified according to ensemble approaches can be seen in Fig. 3. For all the ensemble approaches that we describe below (SG, SG-SQ, VAR), we designate a set size  $J$  of 15 estimates as suggested by (Smilkov et al., 2017). Note that the ensemble approaches described can be wrapped around any interpretability method that produces a ranking of feature importance.

#### CLASSIC SMOOTHGRAD (SG) (Smilkov et al., 2017)

SG averages a set  $J$  noisy estimates of feature importance (constructed by injecting a single input with Gaussian noise  $\eta$  independently  $J$  times):

$$\mathbf{e} = \sum_{i=1}^J (g_i(\mathbf{x} + \eta, A_n^l))$$

**SMOOTHGRAD<sup>2</sup>(SG-SQ)** is an unpublished variant of classic SmoothGrad SG which squares each estimate  $\mathbf{e}$  before averaging the estimates:

$$\mathbf{e} = \sum_{i=1}^J (g_i(\mathbf{x} + \eta, A_n^l)^2)$$

Although SG-SQ is not described in the original publication, it is the default open-source implementation of the open source code for SG: <https://bit.ly/2Hpx5ob>.

**VARGRAD (VAR)** (Adebayo et al., 2018a) employs the same methodology as classic SmoothGrad (SG) to construct a set of  $t J$  noisy estimates. However, VarGrad aggregates the estimates by computing the variance of the noisy set rather than the mean.

$$\mathbf{e} = \text{Var}(g_i(\mathbf{x} + \eta, A_n^l))$$

### 3.2.3. CONTROL VARIANTS

As a control, we compare each estimator to two rankings (a random assignment of importance and a sobel edge filter) that do not depend at all on the model parameters.

**RANDOM** A random estimator  $g^R$  replaces a fraction of all pixels selected at random from each image with a constant uninformative value.

**SOBEL EDGE FILTER** convolves a hard-coded, separable, integer filter over an image to produce a mask of derivatives that emphasizes the edges in an image. A sobel mask treated as a ranking  $\mathbf{e}$  will assign a high score to areas of the image with a high gradient (likely edges).

### 3.3. The Importance of Training a New Model.

Training the model from random initialization on each of the modified datasets is crucial. When a image is modified by replacing the original feaure with a constant value  $c$ , it may introduce artifacts or “new evidence” that distorts model behavior since inference time prediction is done on a different data distribution from that the model is trained on.

This is because the replacement value  $c$  can only be considered uninformative if the value is a variable present in the distribution  $X$  but irrelevant to the classification task:  $c \perp\!\!\!\perp Y$ . Only by training from random initialization on the modified images can we ensure that  $f_w^M$ —where  $w$  specifies the model weights—is trained on a distribution  $X$  that includes  $c$ . It is only in this case that the model can learn that  $c$  is an uninformative value. By including  $c$  in the input distribution, the estimated  $p_w(y, x^M)$  can approximate the true distribution of  $p(y, x^M)$ . In the case without retraining,  $f_w(\mathbf{x})$  has been trained on a distribution  $p(y, x)$  but is expected to approximate  $p(y, x^M)$  at inference time.

In Fig. 5 we compare the difference in performance evaluation between a model that is not re-trained on the modified inputs and the same model that is retrained from random initialization on the modified inputs (for ImageNet). For example, a random modification of 90% of all ImageNet inputs degrades accuracy to 0.5% for the model that was not retrained but when the model is retrained on the same modified inputs the accuracy only degrades to 63.77%. Without retraining the model, it is not possible to decouple the performance of the interpretability method from the degradation caused by the modification itself.

## 4. Experimental Framework and Results

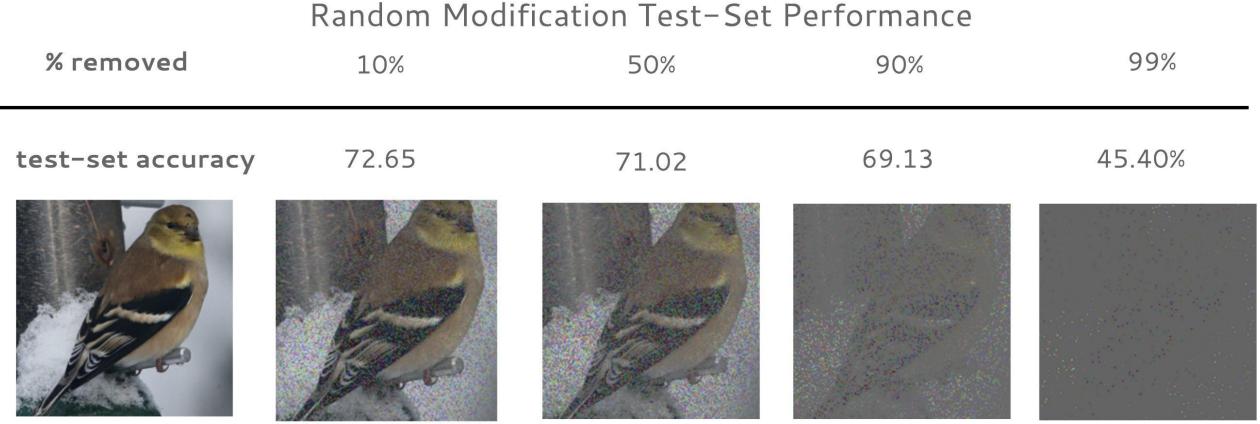
### 4.1. Experiment Framework

We use a ResNet-50 model for both generating the feature importance estimates and subsequent training on the modified inputs. ResNet-50 was chosen because of the public code implementations (in both PyTorch (Gross & Wilber, 2017) and Tensorflow (Abadi et al., 2015)) and because it can be trained to give near to state of art performance in a reasonable amount of time (Goyal et al., 2017).

For all train and validation images in the dataset we first apply test time pre-processing as used by Goyal et al. (2017). We compute an estimate  $\mathbf{e}$  for every input in the training and test set. For all estimators,  $A_n^l$  is pre-softmax activation for the model prediction. We rank each  $\mathbf{e}$  into an ordered set  $\{\mathbf{e}_i^o\}_{i=1}^N$ . For the top  $t$  fraction of this ordered set, we replace the corresponding pixels in the raw image with the per channel mean.

We evaluate ROAR on three open source image datasets: ImageNet, Birdsnap and Food 101. For each dataset and estimator, we generate 10 new train and test sets that each correspond to a different fraction of feature modification  $t = [0.1, 0.3, 0.5, 0.7, 0.9]$  and whether the most important pixels are removed or kept. We evaluate 18 estimators in total (this includes the base estimators, a set of ensemble approaches wrapped around each base and finally a set of squared estimates). In total, we generate 540 large-scale modified image datasets in order to consider all experiment variants (180 new test/train for each original dataset).

We independently train 5 ResNet-50 models from random initialization on each of these modified dataset. We report test accuracy as the average of these 5 runs. In the base implementation, the ResNet-50 trained on an unmodified ImageNet dataset achieves a mean accuracy of 76.68%. This is comparable to the performance reported by (Goyal et al., 2017). On Birdsnap and Food 101, our unmodified datasets achieve 66.65% and 84.54% respectively (average of 10 independent runs). This baseline performance is comparable to that reported by Kornblith et al. (2018).



**Figure 6.** Training is surprisingly robust to random modification. A single example from ImageNet with a varying fraction pixels selected as important at random. The test-set accuracy of a model trained on these modified inputs (averaged over 5 runs) is reported above each image. Test-set accuracy is surprisingly robust to random modification. For example, after randomly replacing 90% of all ImageNet input features, we can still train a model that achieves  $63.53 \pm 0.13$  (average across 5 independent runs). These results suggest that many redundancies in the input feature space exist, however, the basic estimators that we consider are no better than a random guess at identifying them.

## 4.2. Experimental Results

### 4.2.1. ROBUST PERFORMANCE GIVEN RANDOM MODIFICATION

The estimator  $g^R$  assigns importance at random to all inputs. Comparing estimators to this baseline allows us to answer the question: is the estimate of importance more accurate than a random guess? The performance of the random baseline is surprising and consistent across all datasets. After replacing a large portion of all inputs with a constant value, the model not only trains but still retains most of the original predictive power. For example, on ImageNet, when only 10% of all features are retained, the trained model still attains 63.53% accuracy (relative to unmodified baseline of 76.68%).

The ability of the model to extract a meaningful representation from a small random fraction of inputs suggests a case where many inputs are likely redundant. The nature of our input—an image where correlations between pixels are expected—provides one possible reason for this redundancy.

### 4.2.2. ROAR: BASE ESTIMATORS NO BETTER THAN A RANDOM GUESS WHEN RETRAINING

Surprisingly, the left inset of Fig. 4 shows that the base estimators that we consider (GB, IG, GRAD) **consistently perform worse** than the random assignment of feature importance for all thresholds  $t = [0.1, 0.3, 0.5, 0.7, 0.9]$ . This finding is consistent across all datasets. Furthermore, our

estimators fall further behind the accuracy of random guess as a larger fraction  $t$  of inputs is modified. The gap is widest when  $t = 0.9$ . Our base estimators also do not compare favorably to the performance of a sobel edge filter. Across all datasets and thresholds  $t$ , the base estimators GB, IG, GRAD perform on par or worse than SOBEL. This result is noteworthy because both the sobel filter and the random ranking have formulations that are entirely independent of the model parameters. All the base estimators that we consider have formulations that depend upon the trained model weights, and thus we would expect them to have a clear advantage in outperforming the control variants.

### Base estimators perform within very narrow range

Despite the very different formulations of base estimators that we consider, the difference between the performance of the base estimators is in a strikingly narrow range. For example, as can be seen in the right inset of Fig. 4, for Birdsnap, the range of performance between the best and worst base estimator at  $t = .9$  is only 4.22%. This range remains narrow for both Food101 and ImageNet, with a gap of 5.17% and 3.62 respectively between the most and least approximately accurate interpretability method.

### 4.2.3. ROAR: ENSEMBLE APPROACHES ARE NOT CREATED EQUAL

Ensemble approaches inevitably carry a higher computationally approach, as they require the aggregation of a set of individual estimates. These ensemble estimates are often

preferred as an interpretability tool by humans because they appear to produce “less visually noisy” explanations. However, an understanding of what these methods are actually doing or how this is related to the accuracy of the explanation is very limited. Recent work shows that VarGrad VAR produces a ranking that is actually independent of the gradient (Seo et al., 2018). We further the understanding of the advantages and disadvantages of ensemble approaches by evaluating the approximate accuracy of three methods (SG, SG-SQ and VAR).

**Classic SmoothGrad is less accurate or on par with a single estimate** In the middle inset chart of Fig. 4 is the first of a series of intriguing results. Classic SmoothGrad (SG), is the average of 15 estimates computed according to an underlying base method. However, despite the additional computational cost, SG degrades test-set accuracy less than a random guess. In addition, in some cases, SmoothGrad performs worse than a single estimate (for gradient heatmap GRAD and Integrated Gradient IG).

**SmoothGrad-Squared produced large gains in accuracy** SmoothGrad-Squared is an unpublished variation of classic SmoothGrad that squares noisy estimates before averaging. Smoothgrad-Squared, unlike SmoothGrad, produces large gains in accuracy that far outperform a random guess (right inset of Fig. 4). These gains are consistent across all estimators and datasets.

**Squaring Slightly Improves the Performance of All Base Variants** The only difference between SmoothGrad and SmoothGrad-Squared is that with the latter estimates are squared before averaging. The large gap in performance between the two is worth further consideration. In Fig. 8, we consider the effect of only squaring estimates (no ensembling). We include further discussion in the appendix, but find that when squared, an estimate gains slightly more accuracy than a random ranking of input features. However, squaring alone does not explain the large gains in accuracy that we observe when we square each estimate, and then aggregate by averaging the result.

**VarGrad is comparable in performance to SmoothGrad-Squared.** In the right inset of Fig. 4, we show that both VarGrad and SmoothGrad-Squared far outperform the two control variants (a random guess and a sobel edge filter). In addition, for all the interpretability methods we consider, a VarGrad or SmoothGrad-Squared ensemble far outperforms the approximate accuracy of a single estimate.

However, while VarGrad and SmoothGrad-Squared benefit the accuracy of all base estimators, the overall ranking of estimator performance differs by dataset. For Im-

ageNet and Food101, the best performing estimators are VarGrad or SmoothGrad-Squared when wrapped around a gradient heatmap. However, for the Birdsnap dataset, the most approximately accurate estimates are these ensemble approaches wrapped around Guided Backprop. This suggests that while certain ensembling approaches consistently improve performance, the choice of the best underlying estimator may vary by task. This deserves further consideration.

In the right inset of Fig. 4, it can also be seen that the performance of VarGrad VAR is remarkably similar to that of SmoothGrad-Squared (SG-SQ). For many of the estimators, applying SG-SQ and VAR produces virtually identical performance. It is worth revisiting the formulation of VarGrad to consider one possibility for why this would be the case. As first introduced in section 3.2.2 VarGrad is the computed as the variance of a set of noisy estimates:

$$\bar{\mathbf{e}} = \frac{1}{J} \sum_{i=1}^J (g_i(\mathbf{x} + \eta, A_n^l))$$

$$\hat{\mathbf{e}} = \sum_{i=1}^J (g_i(\mathbf{x} + \eta, A_n^l) - \bar{\mathbf{e}})^2$$

It can be seen in the equation above that the first term is in fact equivalent to SG-SQ. One case when SG-SQ and VAR would produce a similar ranking is when the sample mean of the set of estimates is small or close to zero (such that the first term dominates).

## 5. Conclusion and Future Work

In this work, we propose ROAR to evaluate the approximate accuracy of input feature importance estimators. Surprisingly, we find that the commonly used base estimators that we evaluate perform worse or on par with a random assignment of importance. Furthermore, certain ensemble approaches such as SmoothGrad are far more computationally intensive but do not improve upon a single estimate (and in some cases are worse). However, we also find that VarGrad and SmoothGrad-Squared significantly improve the approximate accuracy of a method and far outperform such a random guess. Our findings are particularly pertinent for sensitive domains where the accuracy of a explanation of model behavior is paramount. While we venture some initial consideration of why certain ensemble methods far outperform other estimator, the divergence in performance between the ensemble estimators deserve additional research treatment.

## ACKNOWLEDGMENTS

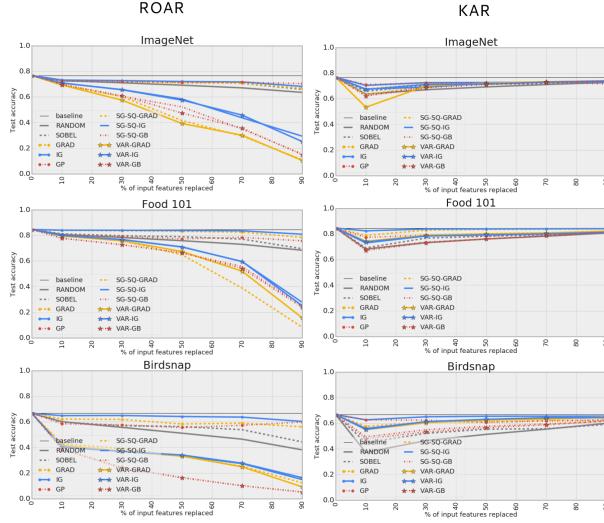
We thank Kevin Swersky, Andrew Ross, Douglas Eck, Jonas Kemp, Melissa Fabros, Julius Abedayo, Simon Kornblith, Prajit Ramachandran, Niru Maheswaranathan and

Gamaleldin Elsayed for their thoughtful feedback on earlier iterations of this work.

## References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, January 2015. URL <https://www.tensorflow.org/>. Software available from tensorflow.org.
- Adebayo, J., Gilmer, J., Goodfellow, I., and Kim, B. Local explanation methods for deep neural networks lack sensitivity to parameter values. *ICLR Workshop*, 2018a.
- Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I. J., Hardt, M., and Kim, B. Sanity checks for saliency maps. In *NeurIPS*, 2018b.
- Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. Towards better understanding of gradient-based attribution methods for Deep Neural Networks. *ArXiv e-prints*, November 2017.
- Ba, L. J. and Caruana, R. Do deep nets really need to be deep? In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 2654–2662, Cambridge, MA, USA, 2014. MIT Press. URL <http://dl.acm.org/citation.cfm?id=2969033.2969123>.
- Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.
- Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.
- Berg, T., Liu, J., Lee, S. W., Alexander, M. L., Jacobs, D. W., and Belhumeur, P. N. Birdsnap: Large-scale fine-grained visual categorization of birds. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2019–2026, 2014.
- Bossard, L., Guillaumin, M., and Van Gool, L. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.
- Dabkowski, P. and Gal, Y. Real Time Image Saliency for Black Box Classifiers. *ArXiv e-prints*, May 2017.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- Dietvorst, B., Simmons, J., and Massey, C. Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of experimental psychology. General*, 144, 11 2014. doi: 10.1037/xge0000033.
- Fong, R. C. and Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In *ICCV*, pp. 3449–3457. IEEE Computer Society, 2017.
- Frosst, N. and Hinton, G. Distilling a Neural Network Into a Soft Decision Tree. *ArXiv e-prints*, November 2017.
- Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *ArXiv e-prints*, June 2017.
- Gross, S. and Wilber, M. Training and investigating Residual Nets. <https://github.com/facebook/fb.resnet.torch>, January 2017.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep Residual Learning for Image Recognition. *ArXiv e-prints*, December 2015.
- Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and sayres, R. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2668–2677, Stockholm, Sweden, 10–15 Jul 2018. PMLR. URL <http://proceedings.mlr.press/v80/kim18d.html>.
- Kindermans, P.-J., Hooker, S., Adebayo, J., Alber, M., Schütt, K. T., Dähne, S., Erhan, D., and Kim, B. The (Un)reliability of saliency methods. *ArXiv e-prints*, November 2017.
- Kindermans, P.-J., Schütt, K. T., Alber, M., Müller, K.-R., Erhan, D., Kim, B., and Dähne, S. Learning how to explain neural networks: Patternnet and patternattribution. *arXiv preprint arXiv:1705.05598v2*, 2017.
- Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

- Kornblith, S., Shlens, J., and Le, Q. V. Do Better ImageNet Models Transfer Better? *arXiv e-prints*, art. arXiv:1805.08974, May 2018.
- Lage, I., Slavin Ross, A., Kim, B., Gershman, S. J., and Doshi-Velez, F. Human-in-the-Loop Interpretability Prior. *arXiv e-prints*, art. arXiv:1805.11571, May 2018.
- Montavon, G., Lapuschkin, S., Binder, A., Samek, W., and Müller, K.-R. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.
- Morcos, A. S., Barrett, D. G. T., Rabinowitz, N. C., and Botvinick, M. On the importance of single directions for generalization. *ArXiv e-prints*, March 2018.
- Olah, C., Mordvintsev, A., and Schubert, L. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. <https://distill.pub/2017/feature-visualization>.
- Poursabzi-Sangdeh, F., Goldstein, D. G., Hofman, J. M., Wortman Vaughan, J., and Wallach, H. Manipulating and Measuring Model Interpretability. *arXiv e-prints*, art. arXiv:1802.07810, February 2018.
- Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. SVCCA: singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *NIPS*, pp. 6078–6087, 2017.
- Ross, A. S. and Doshi-Velez, F. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. *CoRR*, abs/1711.09404, 2017.
- Ross, A. S., Hughes, M. C., and Doshi-Velez, F. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pp. 2662–2670, 2017.
- Samek, W., Binder, A., Montavon, G., Lapuschkin, S., and Müller, K. R. Evaluating the Visualization of What a Deep Neural Network Has Learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673, Nov 2017. ISSN 2162-237X. doi: 10.1109/TNNLS.2016.2599820.
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- Seo, J., Choe, J., Koo, J., Jeon, S., Kim, B., and Jeon, T. Noise-adding Methods of Saliency Map as Series of Higher Order Partial Derivative. *arXiv e-prints*, art. arXiv:1806.03000, Jun 2018.
- Shrikumar, A., Greenside, P., Shcherbina, A., and Kundaje, A. Not Just a Black Box: Learning Important Features Through Propagating Activation Differences. *ArXiv e-prints*, May 2016.
- Shrikumar, A., Greenside, P., and Kundaje, A. Learning Important Features Through Propagating Activation Differences. *ArXiv e-prints*, April 2017.
- Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.
- Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. In *ICLR*, 2015.
- Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.
- Wu, M., Hughes, M. C., Parbhoo, S., Zazzi, M., Roth, V., and Doshi-Velez, F. Beyond Sparsity: Tree Regularization of Deep Models for Interpretability. *ArXiv e-prints*, November 2017.
- Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pp. 818–833. Springer, 2014.
- Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *ArXiv e-prints*, November 2016.
- Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. Object Detectors Emerge in Deep Scene CNNs. *ArXiv e-prints*, 2014.
- Zhou, B., Sun, Y., Bau, D., and Torralba, A. Revisiting the importance of individual units in cnns via ablation. *CoRR*, abs/1806.02891, 2018. URL <http://arxiv.org/abs/1806.02891>.
- Zintgraf, L. M., Cohen, T. S., Adel, T., and Welling, M. Visualizing deep neural network decisions: Prediction difference analysis. In *ICLR*, 2017.



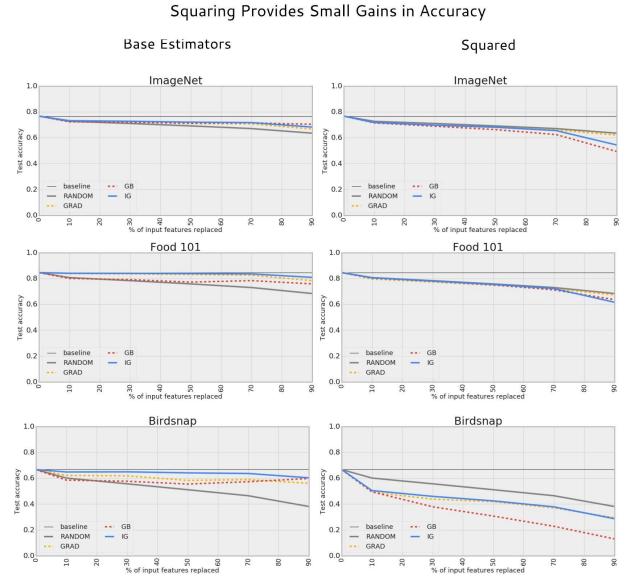
**Figure 7.** Evaluation of all estimators according to Keep And Retrain KAR vs. ROAR. **Left inset:** For KAR, Keep And Retrain, we keep a fraction of features estimated to be most important and replace the remaining features with a constant mean value. The most accurate estimator is the one that preserves model performance the most for a given fraction of inputs removed (the highest test-set accuracy). **Right inset:** For ROAR, Rem0ve And Retrain we remove features by replacing a fraction of the inputs estimated to be most important according to each estimator with a constant mean value. The most accurate estimator is the one that degrades model performance the most for a given fraction of inputs removed. Inputs modified according to KAR result in a very narrow range of model accuracy. ROAR is a more discriminative benchmark, which suggests that retaining performance when the most important pixels are removed (rather than retained) is a harder task.

## A. Supplementary Charts and Experiments

We include supplementary experiments and additional details about our training procedure, image modification process and test-set accuracy below. In addition, as can be seen in Fig. 7, we also consider the scenario where pixels are kept according to importance rather than removed.

### A.1. Generation of New Dataset

We evaluate ROAR on three open source image datasets: ImageNet, Birdsnap and Food 101. For each dataset and estimator, we generate 10 new train and test sets that each correspond to a different fraction of feature modification  $t = [0.1, 0.3, 0.5, 0.7, 0.9]$  and whether the most important pixels are removed or kept. This requires first generating a ranking of input importance for each input image according to each estimator. All of the estimators that we consider evaluate feature importance post-training. Thus, we gener-



**Figure 8.** Certain transformations of the estimate can substantially improve accuracy of all estimators. Squaring alone provides small gains to the accuracy of all estimators, and is slightly better than a random guess. **Left inset:** The three base estimators that we consider (Grad (GRAD), Integrated Gradients (IG) and Guided Backprop (GB)) perform worse than a random assignment of feature importance. At all fractions considered, a random assignment of importance degrades performance more than removing the pixels estimated to be most important by base methods. **Right inset:** Average test-set accuracy across 5 independent iterations for estimates that are squared before ranking and subsequent removal. When squared, base estimators perform slightly better than a random guess. However, this does not compare to the gains in accuracy of averaging a set of noisy estimates that are squared (SMOOTHGRAD-SQUARED)

ate the rankings according to each intepretability method using a stored checkpoint for each dataset.

We use the ranking produced by the interpretability method to modify each image in the dataset (both train and test). We rank each estimate,  $e$  into an ordered set  $\{e_i^o\}_{i=1}^N$ . For the top  $t$  fraction of this ordered set, we replace the corresponding pixels in the raw image with a per channel mean. Fig. 9 and Fig. 10 show an example of the type of modification applied to each image in the dataset for Birdsnap and Food 101 respectively. In the paper itself, we show an example of a single image from each ImageNet modification.

We evaluate 18 estimators in total (this includes the base estimators, a set of ensemble approaches wrapped around each base and finally a set of squared estimators). In total, we generate 540 large-scale modified image datasets in order to consider all experiment variants (180 new test/train for

Dataset	Top1Accuracy	Train Size	Test Size	Learning Rate	Training Steps
Birdsnap	66.65	47,386	2,443	1.0	20,000
Food_101	84.54	75,750	25,250	0.7	20,000
ImageNet	76.68	1,281,167	50,000	0.1	32,000

Table 1. The training procedure was carefully finetuned for each dataset. These hyperparameters are consistently used across all experiment variants. The baseline accuracy of each unmodified data set is reported as the average of 10 independent runs.

each original dataset).

## A.2. Training Procedure

We carefully tuned the hyperparamters of each dataset ImageNet, Birdsnap and Food 101 separately. We find that the Birdsnap and Food 101 converge within the same amount of training steps and a larger learning rate than ImageNet. These are detailed in Table. 1. These hyper parameters, along with the mean accuracy reported on the unmodified dataset, are used consistently across all estimators. ImageNet dataset achieves a mean accuracy of 76.68%. This is comparable to the performance reported by (Goyal et al., 2017). On Birdsnap and Food 101, our unmodified datasets achieve 66.65% and 84.54% respectively. The baseline test-set accuracy for Food101 or Birdsnap is comparable to that reported by Kornblith et al. (2018).

In Table. 2, we include the test-set performance for each experiment variant that we consider. The test-set accuracy reported is the average of 5 independent runs.

## A.3. Evaluating Keeping Rather Than Removing Information

In addition to ROAR, as can be seen in Fig. 7, we evaluate the opposite approach of **KAR**, **K**eep And **R**etrain. While ROAR removes features by replacing a fraction of inputs estimated to be most important, KAR preserves the inputs considered to be most important. Since we keep the important information rather than remove it, *minimizing degradation to test-set accuracy* is desirable.

In the right inset chart of Fig. 7 we plot KAR on the same curve as ROAR to enable a more intuitive comparison between the benchmarks. The comparison suggests that KAR appears to be a poor discriminator between estimators. The x-axis indicates the fraction of features that are preserved/removed for KAR/ROAR respectively.

We find that KAR is a far weaker discriminator of performance; all base estimators and the ensemble variants perform in a similar range to each other. These findings suggest that the task of identifying features to preserve is an easier benchmark to fulfill than accurately identifying a fraction of input that will cause the maximum damage to the model performance.

## A.4. Squaring Alone Slightly Improves the Performance of All Base Variants

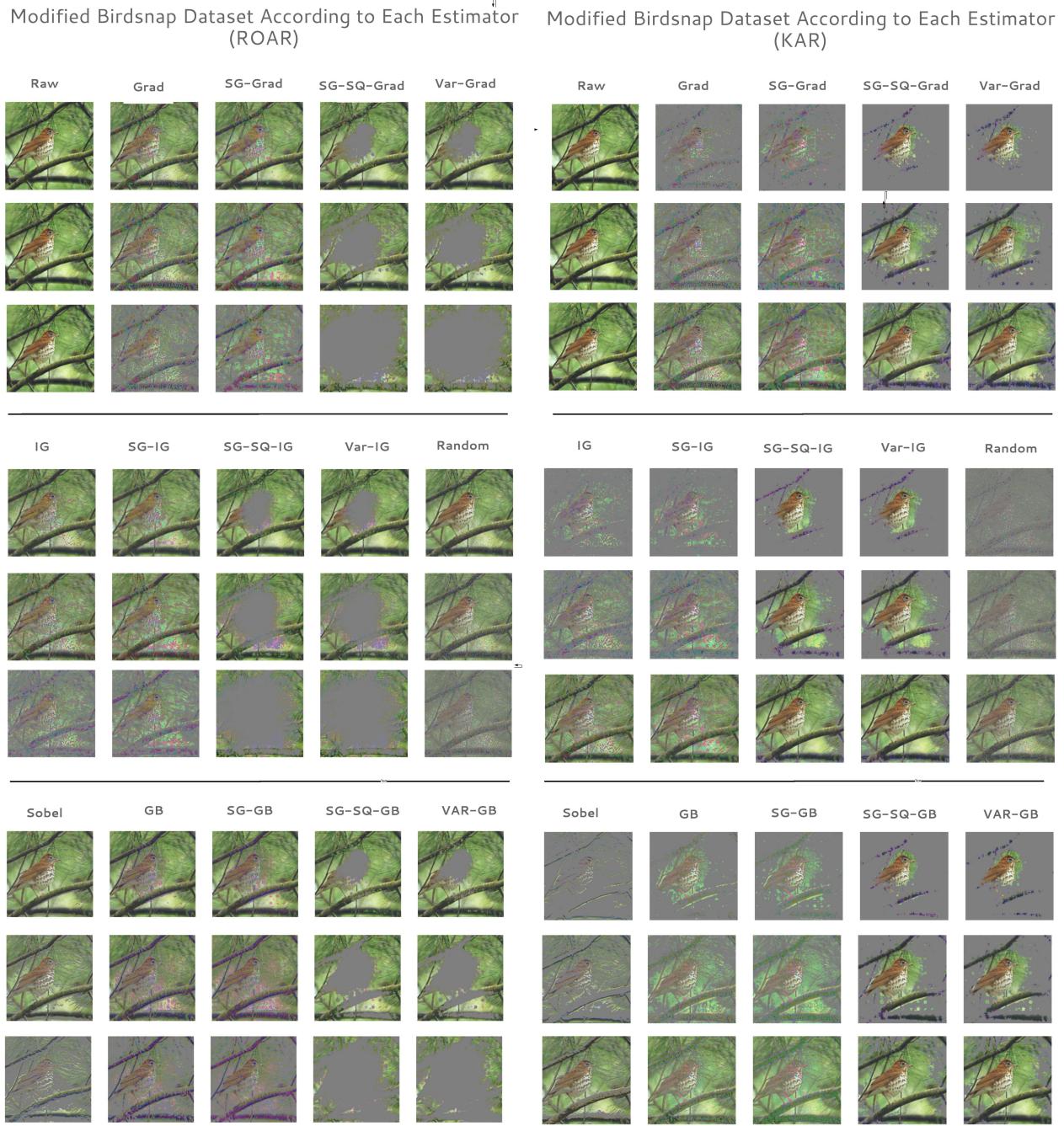
The surprising performance of SmoothGrad-Squared (SG-SQ) deserves further investigation; why is averaging a set of squared noisy estimates so effective at improving the accuracy of the ranking? To disentangle whether both squaring and then averaging are required, we explore whether we achieve similar performance gains by **only** squaring the estimate.

Squaring of a single estimate, with no ensembling, benefits the accuracy of all estimators that we considered. In the right inset chart of Fig. 8, we can see that squared estimates perform better than the raw estimate. When squared, an estimate gains slightly more accuracy than a random ranking of input features. In particular, squaring benefits GB; at  $t = .9$  performance of SQ-GB relative to GB improves by  $8.43\% \pm 0.97$ .

Squaring is an equivalent transformation to taking the absolute value of the estimate before ranking all inputs. After squaring, negative estimates become positive, and the ranking then only depends upon the magnitude and not the direction of the estimate. The benefits gained by squaring furthers our understanding of how the direction of GB, IG and GRAD values should be treated. For all these estimators, estimates are very much a reflection of the weights of the network. The magnitude may be far more telling of feature importance than direction; a negative signal may be just as important as positive contributions towards a model’s prediction. While squaring improves the accuracy of all estimators, the transformation does not explain the large gains in accuracy that we observe when we average a set of noisy squared estimates.



**Figure 9.** A single example from each dataset generated from modifying Food 101 according to both ROAR and KAR. We show the modification for base estimators (Integrated Gradients (IG), Guided Backprop (GB), Gradient Heatmap (GRAD) and derivative ensemble approaches - SmoothGrad, (SG-GRAD, SG-IG, SG-GB), SmoothGrad-Squared (SG-SQ-GRAD, SG-SQ-IG, SG-SQ-GB) and VarGrad (VAR-GRAD, VAR-IG, VAR-GB). In addition, we consider two control variants a random baseline, a sobel edge filter.



*Figure 10.* A single example from each dataset generated from modifying Imagenet according to the ROAR and KAR. We show the modification for base estimators (Integrated Gradients (IG), Guided Backprop (GB), Gradient Heatmap (GRAD) and derivative ensemble approaches - SmoothGrad, (SG-GRAD, SG-IG, SG-GB), SmoothGrad-Squared (SG-SQ-GRAD, SG-SQ-IG, SG-SQ-GB) and VarGrad (VAR-GRAD, VAR-IG, VAR-GB). In addition, we consider two control variants a random baseline, a sobel edge filter.

	Threshold	Keep					Remove				
		10.0	30.0	50.0	70.0	90.0	10.0	30.0	50.0	70.0	90.0
<b>Birdsnap</b>	Random	37.24	46.41	51.29	55.38	59.92	60.11	55.65	51.10	46.45	38.12
	Sobel	44.81	52.11	55.36	55.69	59.08	59.73	56.94	56.30	53.82	44.33
	GRAD	57.51	61.10	60.79	61.96	62.49	62.12	61.82	58.29	58.91	56.08
	IG	62.64	65.02	65.42	65.46	65.50	64.79	64.91	64.12	63.64	60.30
	GP	62.59	62.35	60.76	61.78	62.44	58.47	57.64	55.47	57.28	59.76
	SG-GRAD	64.64	65.87	65.32	65.49	65.78	65.44	66.08	65.33	65.44	65.02
	SG-IG	65.36	66.45	66.38	66.37	66.35	66.11	66.56	66.65	66.37	64.54
	SG-GB	52.86	56.44	58.32	59.20	60.35	54.67	53.37	51.13	50.07	47.71
	SG-SQ-GRAD	55.32	60.79	62.13	63.63	64.99	42.88	39.14	32.98	25.34	12.40
	SG-SQ-IG	55.89	61.02	62.68	63.63	64.43	40.85	36.94	33.37	27.38	14.93
	SG-SQ-GB	49.32	54.94	57.62	59.41	61.66	38.80	24.09	16.54	10.11	5.21
	VAR-GRAD	55.03	60.36	62.59	63.16	64.85	41.71	37.04	33.24	24.84	9.23
	VAR-IG	55.21	61.22	63.04	64.29	64.31	40.21	36.85	34.09	27.71	16.43
	VAR-GB	47.76	53.27	56.53	58.68	61.69	38.63	24.12	16.29	10.16	5.20
<b>Food_101</b>	Random	68.13	73.15	76.00	78.21	80.61	80.66	78.30	75.80	72.98	68.37
	Sobel	69.08	76.70	78.16	79.30	80.90	81.17	79.69	78.91	77.06	69.58
	GRAD	78.82	82.89	83.43	83.68	83.88	83.79	83.50	83.09	82.48	78.36
	IG	82.35	83.80	83.90	83.99	84.07	84.01	83.95	83.78	83.52	80.87
	GP	77.31	79.00	78.33	79.86	81.16	80.06	79.12	77.25	78.43	75.69
	SG-GRAD	83.30	83.87	84.01	84.05	83.96	83.97	84.00	83.97	83.83	83.14
	SG-IG	83.27	83.91	84.06	84.05	83.96	83.98	84.04	84.05	83.90	82.90
	SG-GB	71.44	75.96	77.26	78.65	80.12	78.35	76.39	75.44	74.50	69.19
	SG-SQ-GRAD	73.05	79.20	80.18	80.80	82.13	79.29	75.83	64.83	38.88	8.34
	SG-SQ-IG	72.93	78.36	79.33	80.02	81.30	79.73	76.73	70.98	59.55	27.81
	SG-SQ-GB	68.10	73.69	76.02	78.51	81.22	77.68	72.81	66.24	55.73	24.95
	VAR-GRAD	74.24	78.86	79.97	80.61	82.10	79.55	75.67	67.40	52.05	15.69
	VAR-IG	73.65	78.28	79.31	79.99	81.23	79.87	76.60	70.85	59.57	25.15
	VAR-GB	67.08	73.00	76.01	78.54	81.44	77.76	72.56	66.36	54.18	23.88
<b>ImageNet</b>	Random	63.60	66.98	69.18	71.03	72.69	72.65	71.02	69.13	67.06	63.53
	Sobel	65.79	70.40	71.40	71.60	72.65	72.89	71.94	71.61	70.56	65.94
	GRAD	67.63	71.45	72.02	72.85	73.46	72.94	72.22	70.97	70.72	66.75
	IG	70.38	72.51	72.66	72.88	73.32	73.17	72.72	72.03	71.68	68.20
	GP	71.03	72.45	72.28	72.69	71.56	72.29	71.91	71.18	71.48	70.38
	SG-GRAD	70.47	71.94	72.14	72.35	72.44	72.08	71.94	71.77	71.51	70.10
	SG-IG	70.98	72.30	72.49	72.60	72.67	72.49	72.39	72.26	72.02	69.77
	SG-GB	66.97	70.68	71.52	71.86	72.57	71.28	70.45	69.98	69.02	64.93
	SG-SQ-GRAD	63.25	69.79	72.20	73.18	73.96	69.35	60.28	41.55	29.45	11.09
	SG-SQ-IG	67.55	68.96	72.24	73.09	73.80	70.76	65.71	58.34	43.71	29.41
	SG-SQ-GB	62.42	68.96	71.17	72.72	73.77	69.74	60.56	52.21	34.98	15.53
	VAR-GRAD	53.38	69.86	72.15	73.22	73.92	69.24	57.48	39.23	30.13	10.41
	VAR-IG	67.17	71.07	71.48	72.93	73.87	70.87	65.56	57.49	45.80	25.25
	VAR-GB	62.09	68.51	71.09	72.59	73.85	69.67	60.94	47.39	35.68	14.93

Table 2. Average test-set accuracy across 5 independent runs for all estimators and datasets considered. ROAR requires removing a fraction of pixels estimated to be most important. KAR differs in that the pixels estimated to be most important are kept rather than removed. The fraction removed/kept is indicated by the threshold. The estimators we report results for are base estimators (Integrated Gradients (IG), Guided Backprop (GB), Gradient Heatmap (GRAD) and derivative ensemble approaches - SmoothGrad, (SG-GRAD, SG-IG, SG-GB), SmoothGrad-Squared (SG-SQ-GRAD, SG-SQ-IG, SG-SQ-GB) and VarGrad (VAR-GRAD, VAR-IG, VAR-GB).