A Consistent and Efficient Evaluation Strategy for Attribution Methods

Yao Rong [*1] **Tobias Leemann** [*1] **Vadim Borisov** [1] **Gjergji Kasneci** [1] **Enkelejda Kasneci** [1]

Abstract

With a variety of local feature attribution methods being proposed in recent years, follow-up work suggested several evaluation strategies. To assess the attribution quality across different attribution techniques, the most popular among these evaluation strategies in the image domain use pixel perturbations. However, recent advances discovered that different evaluation strategies produce conflicting rankings of attribution methods and can be prohibitively expensive to compute. In this work, we present an information-theoretic analysis of evaluation strategies based on pixel perturbations. Our findings reveal that the results are strongly affected by information leakage through the shape of the removed pixels as opposed to their actual values. Using our theoretical insights, we propose a novel evaluation framework termed Remove and Debias (ROAD) which offers two contributions: First, it mitigates the impact of the confounders, which entails higher consistency among evaluation strategies. Second, ROAD does not require the computationally expensive retraining step and saves up to 99 % in computational costs compared to the state-of-the-art. We release our source code at `https://github.com/tleemann/road_evaluation`.

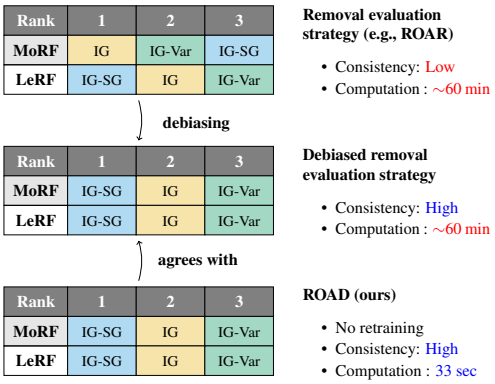


Figure 1. Comparison between previous removal and retraining evaluation strategies (**Top**) and ours (**Bottom**). Previously, rankings of different attribution methods, Integrated Gradients (IG) (Sundararajan et al., 2017) and its two variants SmoothGrad (IG-SG) (Smilkov et al., 2017), SmoothGrad^2 (IG-SQ) (Hooker et al., 2019), are highly inconsistent with respect to hyperparameters such as the removal orders Most Relevant First (MoRF) and Least Relevant First (LeRF). Our ROAD strategy achieves a consistent ranking using only 1% of the previously required resources.

1. Introduction

Explainable Artificial Intelligence (XAI) has become a widely discussed research topic (Adadi & Berrada, 2018). Specifically, feature attribution methods (Springenberg et al., 2015; Ribeiro et al., 2016; Lundberg & Lee, 2017; Sundararajan et al., 2017; Selvaraju et al., 2017) that quantify the importance of input features to a model's decision are widely used. Such local explanations can help to analyze and debug predictive models (Bhatt et al., 2020b; Adebayo et al., 2020), e.g., in the medical domain (Eitel et al., 2019), in recommender systems (Afchar & Hennequin, 2020), and many other applications. With an increasing number of feature attribution methods proposed in the literature, the need for sound strategies to evaluate these methods is also increasing (Nguyen & Martínez, 2020; Hase & Bansal, 2020; Yeh et al., 2019; Hooker et al., 2019).

Evaluation strategies, proposed to compare different attribution methods, commonly follow an ablation approach by perturbing the input features, e.g., image pixels, deemed most or least important. Specifically, perturbing pixels assigned high importance should decrease predictive quality whereas perturbing unimportant pixels, should hardly affect the predictions. These measures aim to capture the *fidelity* of explanations (Tomsett et al., 2020), i.e., how well the explanation genuinely reflects the prediction of the

[*]Equal contribution [1]Department of Computer Science, University of Tübingen, Tübingen, Germany. Correspondence to: Yao Rong <yao.rong@uni-tuebingen.de>, Tobias Leemann <tobias.leemann@uni-tuebingen.de>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

underlying model. Fidelity based on a single data sample is known as local fidelity, while global fidelity is measured on the whole data set (Tomsett et al., 2020).

The outcome of evaluation strategies is highly sensitive to parameters such as the perturbation function and order. Depending on the order chosen, i.e., *most relevant pixels first* or *least relevant pixels first*, such removal strategies often lead to highly contradictory results. For instance, local attribution methods that seem to perform well in one order may perform rather poorly in the other (Tomsett et al., 2020; Haug et al., 2021; Hooker et al., 2019). This inconsistency makes it hard for researchers to impartially compare between different attribution methods and it is not well understood where the inconsistencies stem from. Moreover, for conducting the global fidelity check, a retraining step is required by some methods (Hooker et al., 2019), which is prohibitively expensive in practice (Tomsett et al., 2020). These two drawbacks and our improvements are illustrated in Figure 1.

In this paper, we aim to overcome these shortcomings and make the evaluation more consistent and efficient. To this end, we propose a new debiased strategy that compensates for confounders causing inconsistencies. Furthermore, we show that in the debiased setting, we can skip the retraining without significant changes in the results. This results in drastic efficiency gains as shown in the lower part of Figure 1. We argue that it is crucial for the community to have sound evaluation strategies that do not suffer from limited accessibility due the required compute capacity. Specifically, we make the following contributions:

- We examine the mechanisms underlying the evaluation strategies based on perturbation by conducting a rigorous information-theoretic analysis, and formally reveal that results can be significantly confounded.
- To compensate for this confounder, we propose the Noisy Linear Imputation strategy and empirically prove its efficiency and effectiveness. The proposed strategy significantly decreases the sensitivity to hyperparameters such as the removal order.
- We generalize our findings to a novel evaluation strategy, ROAD (RemOve And Debias), which can be used to objectively and efficiently evaluate several attribution methods. Compared to previous evaluation strategies requiring retraining, e.g., Remove and Retrain (ROAR) (Hooker et al., 2019), ROAD saves 99 % of the computational costs.

2. Related Work

There is a plethora of works on different explanation techniques (Tjoa & Guan, 2020), especially attribution methods that assign importance scores to each input features. Popular approaches have been proposed by Springenberg et al. (2015); Lapuschkin et al. (2015); Ribeiro et al. (2016); Kasneci & Gottron (2016); Sundararajan et al. (2017); Fong & Vedaldi (2017); Shrikumar et al. (2017); Smilkov et al. (2017); Petsiuk et al. (2018); Adebayo et al. (2018); Chen et al. (2018); Xu et al. (2020); Covert et al. (2021), and many more.

With the growing number of attribution methods, various scholars have presented desiderata that explanations should fulfill (Bhatt et al., 2020a; Nguyen & Martínez, 2020; Fel et al., 2021; Afchar et al., 2021; Nauta et al., 2022). Doshi-Velez & Kim (2017) consider two subcategories in this field, namely *human-grounded* metrics relying on human judgment and *functional-grounded* metrics. The latter do not require a human-generated ground truth that can be hard or even impossible to obtain. Metrics of this type frequently rely on the idea that if the most important part of the image is changed, the output probability of the given black-box model should also change in return. Examples include the Sensitivity-n measure proposed by Ancona et al. (2017) and the infidelity and max-sensitivity metrics by Yeh et al. (2019). Samek et al. (2016) and Petsiuk et al. (2018) also propose to perturb the pixels in the input image according to the importance scores. However, Hooker et al. (2019) show that the perturbation introduces artifacts and results in a distribution shift, putting these no-retraining approaches in question. They propose the Remove and Retrain (ROAR) framework with an extensive model retraining step to adapt to the distribution shift. Therefore, we distinguish between evaluation methods with *retraining* and *no-retraining* approaches. ROAR has been adopted in several recent studies (Hartley et al., 2020; Izzo et al., 2020; Meng et al., 2021; Schramowski et al., 2020; Srinivas & Fleuret, 2019) and variations are being proposed in concurrent work (Shah et al., 2021).

Only few papers have used and compared different evaluation strategies for attribution methods and a sound theoretical explanation for the differences between them is still missing. Sturmfels et al. (2020) assess different baselines for feature attribution applying the Integrated Gradient method (Sundararajan et al., 2017). They also observe that changing the hyperparameter settings can lead to varying results. Haug et al. (2021) draw the same conclusion for attributions on tabular data. Tomsett et al. (2020) compute the consistency among different, no-retraining evaluation strategies and report an alarmingly low agreement. In this work, we conduct a rigorous analysis of reasons for existing inconsistency and provide a solution to reduce it, which is not studied in previous works. Moreover, our solution also reduces high computational costs caused by retraining.

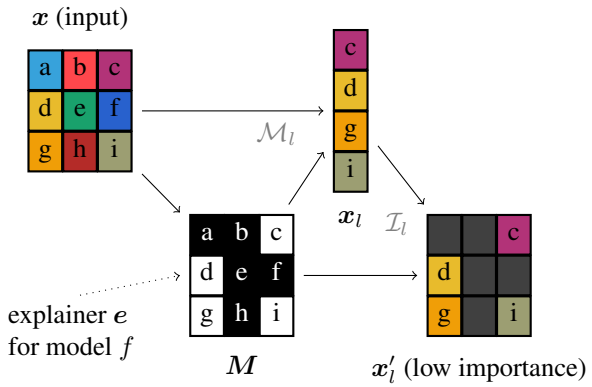


Figure 2. Our analytical model of feature removal evaluation (MoRF order shown): The input image $\boldsymbol{x}$ (9 pixels a–i) is explained by an explanation method that returns a mask $\boldsymbol{M}$ indicating important pixels (black). The remaining, less important pixel values $\boldsymbol{x}_l$ can be extracted from the image using the masking operator $\mathcal{M}_l$ and transformed via the imputation operator $\mathcal{I}_l$ to an imputed variant of the input $\boldsymbol{x}'_l$, which determines the evaluation outcome. This model allows to separate the information in the feature *values* from that contained in the binary mask $\boldsymbol{M}$.

3. Preliminaries

In this section, we formally define the pixel-perturbation strategies considered by the following analysis.

3.1. Retraining Evaluation Strategies

We consider a pixel removal strategy, where pixels are successively replaced by imputed values. Consistent with the literature (Tomsett et al., 2020; Samek et al., 2016), we consider two removal orders: **MoRF** (Most Relevant First) or **LeRF** (Least Relevant First), where the subsequent removal starts with the most important pixels for the former and the least important ones for the latter. We now provide a formal definition of MoRF with retraining, i.e., the ROAR benchmark, that will be used throughout our analysis. We always use the MoRF order in the analysis presented in this paper. However, an analogous analysis of its counterpart LeRF is possible without much additional effort and can be found in the appendix.

To ease our derivations, we describe the procedure by a series of operations that can be analyzed independently. A classifier $f : \mathbb{R}^d \to \{1, \dots, c\}$ maps inputs $\boldsymbol{x} \in \mathbb{R}^d$ to labels $C \in \{1, \dots, c\}$, where c is the number of classes. A feature attribution explanation for the prediction assigns each input dimension an importance value. In the MoRF setting, the features are ordered in a descending order of importance. Subsequently, the k most important features per instance are selected for removal, where $0 \leq k \leq d$ is successively increased during the benchmark. However, for the moment we consider only one fixed value of k. Thus,

C	Class label random variable
I	Mutual information
$\mathcal{I}$	Imputation operator
$\boldsymbol{M}$	Binary mask in $\{0,1\}^d$
$\mathcal{M}$	Mask selection operator (takes out relevant features)
$\boldsymbol{x}$	Input features in $\mathbb{R}^d$
$\boldsymbol{x}_l$	Low importance features only in $\mathbb{R}^{d-k}$
$\boldsymbol{x}'_l$	Imputed low importance features in $\mathbb{R}^d$

Table 1. Overview of the notation used in this work.

we can model the explanation $\boldsymbol{e}_k$ as a choice of features via a binary mask $\boldsymbol{M} = \boldsymbol{e}_k(f, \boldsymbol{x}) \in \{0,1\}^d$, with the corresponding value set to one, if the corresponding feature is among the top-k, and to zero otherwise. Furthermore, suppose $\mathcal{M}_l : \{0,1\}^d \times \mathbb{R}^d \to \mathbb{R}^{d-k}$ to be the selection operator for the least important dimensions indicated in the mask and $\boldsymbol{x}_l = \mathcal{M}_l(\boldsymbol{M}, \boldsymbol{x})$ to be a vector containing only the remaining features as shown in Figure 2. We suppose that the features preserve their internal order in $\boldsymbol{x}_l$, i.e., features are ordered ascendingly by their original input indices. This definition allows to separately consider the information flow in the feature *mask* $\boldsymbol{M}$ and that in the feature *values* $\boldsymbol{x}_l$.

The ROAR approach measures the accuracy of a newly trained classifier f' on modified samples $\boldsymbol{x}'_l := \mathcal{I}_l(\boldsymbol{M}, \boldsymbol{x_l})$, where $\mathcal{I}_l : \{0,1\}^d \times \mathbb{R}^{d-k} \to \mathbb{R}^d$ is an imputation operator that redistributes all inputs in the vector $\boldsymbol{x}_l$ to their original positions and sets the remainder to some filling value. In the special case of zero imputation, $\boldsymbol{x}'_l = \mathcal{I}_l(\boldsymbol{M}, \mathcal{M}_l(\boldsymbol{M}, \boldsymbol{x})) = (1 - \boldsymbol{M}) \odot \boldsymbol{x}$. This means the top-$k$ features are discarded. For a better evaluation result, the accuracy should drop quickly with increasing k, indicating that the most influential features were successfully removed.

3.2. Information Theory

We now briefly revisit the central concepts of information theory that will be handy for our analysis and introduce the notation. The fundamental quantity in information theory is the entropy H of a discrete random variable X with support $\operatorname{supp}\{X\}$,

$$H(X) := -\sum_{x \in \operatorname{supp}\{X\}} P(X = x) \log P(X = x). \quad (1)$$

The entropy corresponds to the information gained through observation of a realization of this variable. If the random variable considered can be easily inferred, we use $p(x)$ as a shorthand for $P(X = x)$. Furthermore, we denote the joint entropy between random variables X and Y by $H(X, Y)$, which is equivalent to the entropy of their joint distribution. In accordance with Cover & Thomas (2006), we always

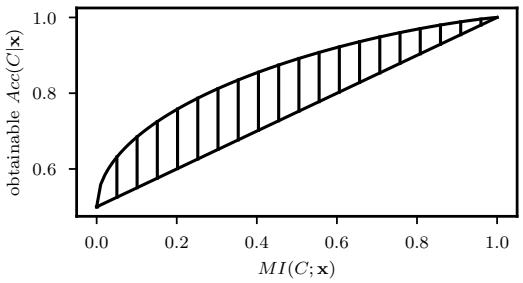


Figure 3. Relation between Mutual Information (MI) and obtainable accuracy for the two-class problem with equal class priors. The knowledge of the MI $I(\boldsymbol{x};C)$ implies strong bounds for the obtainable accuracy. This connection permits to use MI as a surrogate for the obtainable accuracy in the perturbation strategy in our analysis. Figure adapted from Meyen (2016).

separate random variables by comma to denote the joint distribution of multiple of variables.

The conditional entropy $H(X|Y)$ is the expected amount of information left in a variable, given the observation of a condition Y. The most central concept in our analysis will be mutual information (MI), i.e., the amount of information in one random variable shared with another. For example, by $I(\boldsymbol{x};C) := H(C) - H\left(C|\boldsymbol{x}\right)$, we denote the MI between the complete feature vector and the class variable C. We separate arguments by a semicolon and allow single random variables or sets of random variables as arguments to all the defined quantities. For sets, we always consider the joint distribution of their member variables. Please confer Cover & Thomas (2006) for a more profound introduction. We provide a short overview of our notation in Table 1.

4. Analysis

In this section, we show that the pixel perturbation strategies are susceptible to a previously unknown confounder: The binary mask itself can leak class information that might in not be present in the feature values. After making the connection between the accuracy and mutual information as a theoretical tool in Section 4.1, we formally derive the confounder and identify this leakage on real data in Section 4.2. We subsequently show how to mitigate it through Minimally Revealing Imputation in Section 4.3.

4.1. On the Relation Between Accuracy and Mutual Information

To begin our analysis of the presented strategies and their underlying mechanisms, we first establish the relation between classification accuracy and the mutual information. It is well-known that the classification performance of an optimal classifier in the Bayesian sense (assigning the class with the highest posterior) is dependent on the MI between features and labels (Hellman & Raviv, 1970; Vergara & Estévez, 2014; Meyen, 2016). Nevertheless, the relationship is not a function, but comes in form of upper and lower bounds of the obtainable accuracy. For the simple two-class problem, the bounds are shown in Figure 3 (cf. Appendix A.1 for derivations). They impose strong limits on the optimal classification performance, if the mutual information $I(\boldsymbol{x};C)$ is known.

For the pixel removal strategies that use retraining, this allows us to analyze the frameworks using MI as a surrogate for the attainable accuracy because higher MI almost always leads to higher accuracy. In the MoRF setting with retraining, $I(\boldsymbol{x}'_l;C)$ will play a key role, because it quantifies the information left in the least important features and thus determines obtainable accuracy which is the outcome of the evaluation. Low mutual information $I(\boldsymbol{x}'_l;C)$ results in a sharp drop in accuracy and good benchmarking results:

$$\downarrow I(\boldsymbol{x}'_l;C) \;\; \Rightarrow \uparrow \text{MoRF benchmark.}$$

Therefore, in the MoRF setting low mutual information of $\boldsymbol{x}'_l$ and C is desirable[1].

4.2. Class Information Leakage through Masking

We demonstrate that it is easily possible to leak class information only through the mask's shape and to harshly manipulate the evaluation score. Therefore, we start by separating the influence of the mask from that of the feature values. Our derivation relies on the multi-information $I(C;\boldsymbol{x}'_l;\boldsymbol{M})$, which is defined by Vergara & Estévez (2014) as follows:

$$I(C;\boldsymbol{x}'_l;\boldsymbol{M}) = I(C;\boldsymbol{x}'_l|\boldsymbol{M}) - I(C;\boldsymbol{x}'_l) \tag{2}$$
$$I(C;\boldsymbol{x}'_l;\boldsymbol{M}) = I(C;\boldsymbol{M}|\boldsymbol{x}'_l) - I(C;\boldsymbol{M}). \tag{3}$$

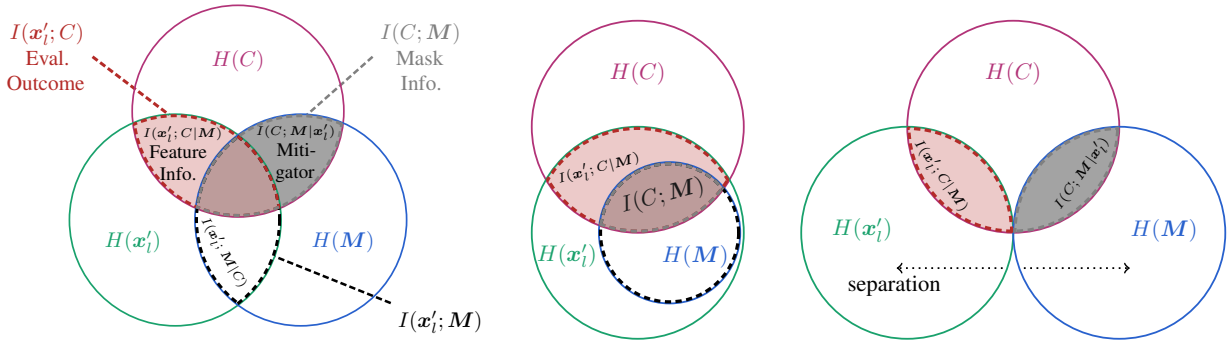
Setting Equation (2) and Equation (3) equal, we arrive at the identity:

$$\underbrace{I(\boldsymbol{x}'_l;C)}_{\text{Eval. Outcome}} = \underbrace{I(C;\boldsymbol{x}'_l|\boldsymbol{M})}_{\text{Feature Info.}} + \underbrace{I(C;\boldsymbol{M})}_{\text{Mask Info.}} - \underbrace{I(C;\boldsymbol{M}|\boldsymbol{x}'_l)}_{\text{Mitigator}}. \tag{4}$$

The quantities involved are visualized in Figure 4a. The first term "Feature Information" is the class information contained in the features (and not in the mask) that we wish to estimate. The second term "Mask Information" shows that class-discriminative information in the mask can have a high impact on the result. This influence can be compensated by the "Mitigator" term.

Class Information Leakage If the Mask Information term is superior to the Mitigator, $I(C;\boldsymbol{M}) > I(C;\boldsymbol{M}|\boldsymbol{x}'_l)$,

[1] In LeRF, a higher accuracy and thus higher $I(\boldsymbol{x}'_l;C)$ is beneficial



(a) General Case (b) Invertible Imputation (c) Minimally Revealing Imputation

Figure 4. The Evaluation Outcome $I(\boldsymbol{x}_l'; C)$ (red area), is confounded by the Mask Information $I(C; \boldsymbol{M})$ (gray area) when there is some overlap (a). Only the Feature Information $I(\boldsymbol{x}_l'; C|\boldsymbol{M})$, the part of the Outcome not overlapping (light red area), should actually be assessed. In the worst case (which we term Invertible Imputation), the Mask Information is entirely contained in the Outcome (b). Separating the information in the imputed image $\boldsymbol{x}_l'$ and the mask $\boldsymbol{M}$ allows to reduce the overlap and the influence (c).

the evaluation outcome is unfairly increased to a value not justified by the selected features. We term this phenomenon *Class Information Leakage*, as some discriminative information is "leaked" through the used binary mask $\boldsymbol{M}$.

The Mitigator can entirely vanish when the mask is perfectly inferable from the imputed image $\boldsymbol{x}_l'$. This results in a non-compensated effect of Class Information Leakage. We define this imputation operation as follows:

Condition 4.1. *Invertible Imputation. Let* $\mathcal{I}_l : \{0,1\}^d \times \mathbb{R}^{d-k} \to \mathbb{R}^d$ *be the imputation operator that takes the least important features as an input. We suppose that there are inverse functions* $\mathcal{I}_{l,M}^{-1}$ *and* $\mathcal{I}_{l,x}^{-1}$*, such that*

$$\boldsymbol{x}_l' = \mathcal{I}_l\left(\boldsymbol{M}, \boldsymbol{x}_l\right) \Leftrightarrow \boldsymbol{M} = \mathcal{I}_{l,M}^{-1}(\boldsymbol{x}_l') \wedge \boldsymbol{x}_l = \mathcal{I}_{l,x}^{-1}(\boldsymbol{x}_l').$$

If, for instance, the pixels removed are set to some reserved value indicating their absence, the imputation operator is invertible, as the mask can be reconstructed. Therefore, $H(\boldsymbol{M}|\boldsymbol{x}_l')=H\left(\mathcal{I}_{l,M}^{-1}(\boldsymbol{x}_l')|\boldsymbol{x}_l'\right)=0$. In this case, also the Mitigator $I\left(C; \boldsymbol{M}|\boldsymbol{x}_l'\right) = 0$, because it is bounded by $0 = H(\boldsymbol{M}|\boldsymbol{x}_l') \geq I\left(C; \boldsymbol{M}|\boldsymbol{x}_l'\right) \geq 0$. The Feature Information term is constrained to be positive. Thus, the Mask Information has a non-negligible impact on the Evaluation Outcome because a higher Mask Information term will always increase it. This case is depicted in Figure 4b.

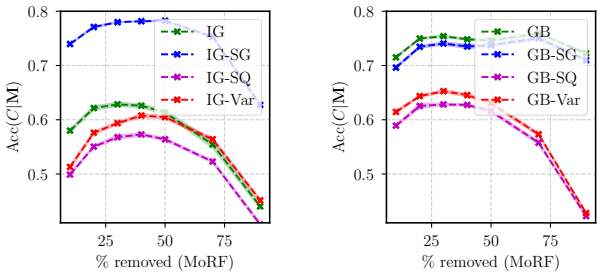
We can create a simple example that shows how evaluation scores are influenced: Imagine a two-class problem that consists of detecting whether an object is located on the left or the right side of an image. A reasonable attribution method masks out pixels on the left or the right depending on the location of the object. In this case, the retraining step can lead to a classifier that infers the class just from the location of the masked out pixels and obtain high accuracy. This explanation map will be rated far worse in MoRF (no accuracy drop) than it might actually be. In the context of amortized explanation methods, a similar finding has been made by Jethani et al. (2021). We theoretically showed that this problem also arises in evaluation strategies and empirically demonstrate that the leakage is significant for popular attribution methods on real data in Section 5.1.

4.3. Reduction of Information Leakage

To tackle this problem, we follow an intuitive approach: If we cannot guarantee that there is no class information contained in the mask itself, we have to stop it from leaking the class information into the imputed images. Therefore, we make sure that the mask used cannot be easily inferred from the imputed image. We would like to set $I(\boldsymbol{x}_l'; \boldsymbol{M}) = 0$, i.e., the mask is independent of the imputed vector allowing to separate the effects as shown in Figure 4c. Unfortunately, this is not possible in general: If both should be dependent on the class label, they will also have to share a minimal amount of information (that regarding the class). However, we can demand conditional independence and make $I(\boldsymbol{x}_l'; \boldsymbol{M})$ as small as possible.

Condition 4.2. *Minimally Revealing Imputation. Let* $\mathcal{I}_l : \{0,1\}^d \times \mathbb{R}^{d-k} \to \mathbb{R}^d$ *be the infilling operator that takes the least important features as an input. Suppose* $\boldsymbol{x}_l'$ *and* $\boldsymbol{M}$ *are independent given the class information* $I\left(\boldsymbol{x}_l'; \boldsymbol{M}|C\right) = 0$ *and* $I\left(\boldsymbol{x}_l'; \boldsymbol{M}\right) \approx 0$.

In this case, $I(C; \boldsymbol{M}) - I\left(C; \boldsymbol{M}|\boldsymbol{x}_l'\right) = I\left(\boldsymbol{x}_l'; \boldsymbol{M}\right) - I\left(\boldsymbol{x}_l'; \boldsymbol{M}|C\right) \approx 0$, which implies $I(C; \boldsymbol{M}) \approx I\left(C; \boldsymbol{M}|\boldsymbol{x}_l'\right)$ (also cf. Figure 4c), indicating that the Mitigator effectively compensates the Mask Information term.



(a) Integrated Gradients (IG)

(b) Guided Backpropagation (GB)

Figure 5. Accuracy of a trained classifier only using the binary masks $\boldsymbol{M}$ without feature values as input on the CIFAR-10 data set. Binary masks $\boldsymbol{M}$ were computed for different variants of IG and GB. Only the masks contain enough information to reach an accuracy of almost up to 80 % (compared to 85 % with full images) highlighting that the feature values do not play an important role in the evaluation. This underlines the necessity to compensate for this confounder.

5. Debiasing Evaluation Strategies for Local Attribution Methods

With the theoretical analysis in Section 4, we can better understand where the biases come from, and thus mitigate them. Building on the derivations, we now show the strong impact of the Class Information Leakage introduced in Section 4.2 on a real-world data set to highlight the necessity to compensate for this confounder. We explain how we reduce its influence by proposing a novel imputation operator termed *Noisy Linear Imputation*.

5.1. Extent of Class Information Leakage

To empirically confirm our findings, we performed experiments on CIFAR-10 (Krizhevsky et al., 2009). We use the same attribution methods as in Hooker et al. (2019): Integrated Gradients (IG) (Sundararajan et al., 2017) and Guided Backprop (GB) (Springenberg et al., 2015) serve as base explanations, and three ensembling strategies for each are used in addition: SmoothGrad (SG) (Smilkov et al., 2017), SmoothGrad^2 (SQ) (Hooker et al., 2019) and VarGrad (Var) (Adebayo et al., 2018). In total, we consider eight attribution methods and provide details and parameters in the supplementary material.

We empirically show that with fixed value imputation with the global mean, the explanation masks are leaking class information. This takes two steps: (1) We show that the Mask Information $I(C; \boldsymbol{M})$ is extremely high. (2) We verify that the Mitigator is small by testing the *Invertible Imputation* Condition, which implies that class information is leaked into the evaluation outcome through $I(C; \boldsymbol{M})$.

To assess the class information in the mask, we train a ResNet-18 (He et al., 2016) that uses only binary masks $\boldsymbol{M}$ (no pixel values $\boldsymbol{x}_l$) to predict the class. As we discussed previously, the accuracy of a classifier can be used as a surrogate for the calculation of MI, which is prohibitively expensive for high-dimensional data. The curves[2] are shown in Figure 5. Stunningly, the mask alone results in high accuracy curves that reach almost 80 % for IG-SG, only some percent below the accuracy of the classifier on the full inputs. This allows us to conclude that the Mask Information $I\left(C; \boldsymbol{M}\right)$ is almost as high as our Evaluation Outcome $I\left(C; \boldsymbol{x}'_l\right)$.

To show that the Mitigator is almost zero which leads to class information leakage, we test the *Invertible Imputation* condition. Therefore, the inverse function $\mathcal{I}^{-1}_{l,M}$ that predicts the imputation mask from the imputed image is required (having this function, finding $\mathcal{I}^{-1}_{l,x}$ is trivial). For the fixed value imputation, an approximate inverse is simple: Setting all pixels in the mask to 0 if the corresponding image pixel has the filling value (which has to be inferred from the distribution). For a stronger verification, we train an imputation predictor network consisting of three convolutional layers, which predicts for each pixel if it was imputed or original. As Figure 6e (blue curve) shows, the miss-classification rate when using fixed value imputation is almost zero, i.e., the network can easily recognize the pixels that were imputed. According to our analysis, in this setting close to *Invertible Imputation*, the Mitigator will be negligibly small.

This leads us to the conclusion that the mask-related leakage fundamentally influences many previous evaluations using fixed value imputation (Shrikumar et al., 2017; Petsiuk et al., 2018; Hooker et al., 2019) and it is essential to stop the information leakage through the masks.

5.2. Debiasing with Noisy Linear Imputation

To reduce the Class Information Leakage, we propose a better-suited imputation operator $\mathcal{I}_l$ that adheres to the *Minimally Revealing Imputation* condition we derived. The remaining process is left unchanged and stays as depicted in Figure 2. However, we face three requirements: (1) We have to get closer to the theoretical condition of Minimally Revealing Imputation. (2) The imputation strategy needs to be highly efficient, since the imputation module has to be run for each image in the data set. (3) We wish to have as few hyper-parameters as possible (preferably none to rule out another confounding factor).

We devise a new strategy called *Noisy Linear Imputation*, which fulfills the above goals. In this way, our model addresses some of the fundamental problems of existing

[2]Standard Errors are indicated by shaded areas in all figures. However, they are often hardly visible due to their low magnitude.

strategies. Intuitively, we search a way to make more subtle imputations that cannot be easily recognized and result in lower $I\left(\boldsymbol{x}_l'; \boldsymbol{M}\right)$. To this end, we suppose that each pixel can be approximated by the weighted mean of its neighbors (cf. Figure 6d) as image pixels are highly correlated[3]:

$$\begin{aligned}\boldsymbol{x}_{i,j} = w_d \left(\boldsymbol{x}_{i,j+1} + \boldsymbol{x}_{i,j-1} + \boldsymbol{x}_{i+1,j} + \boldsymbol{x}_{i-1,j}\right) \\ + w_i \left(\boldsymbol{x}_{i+1,j+1} + \boldsymbol{x}_{i-1,j+1} + \boldsymbol{x}_{i+1,j-1} + \boldsymbol{x}_{i-1,j-1}\right)\end{aligned}$$

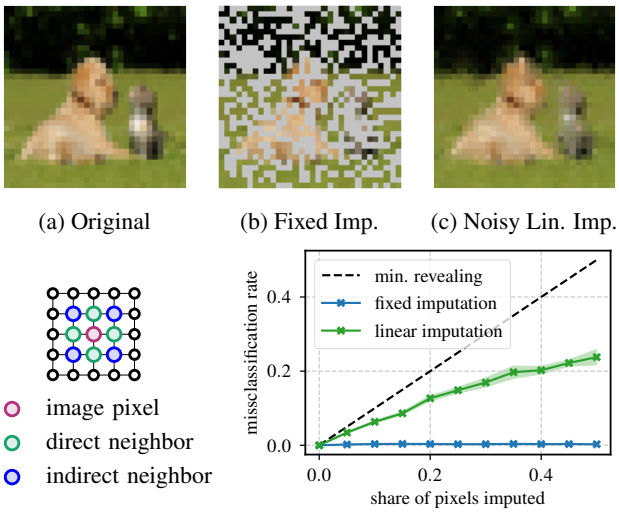
where w_d, w_i are constant coefficients for direct neighbors and indirect, diagonal neighbors. When setting up a single equation for each removed pixel we arrive at an equation system. For known pixels, we directly plug in their values and only consider each removed pixel as an unknown variable. When neighboring pixels are removed, the equations become connected and cannot be solved independently. Nevertheless, the resulting system is sparse and can be efficiently solved, even for a large number of missing pixels. To choose the neighbor weights for the linear interpolation, we draw inspiration from the graph structure (see Figure 6d): Indirect neighbors have distance 2 from the original node in the graph and direct neighbors have distance 1. Hence, we gave the direct neighbors twice the weight of the diagonal ones. Because the weights need to some up to 1 for a weighted interpolation, this leads to $w_d{=}\frac{1}{6}$ and $w_i{=}\frac{1}{12}$. We add a small random noise ($\sigma = 0.1$) to the solution to ensure that the linear dependency cannot be learned by the model.

Figure 6 (top) provides an example of an imputed sample. From the imputed version in Figure 6c, inference on the mask is significantly harder than the one imputed with fixed values as in Figure 6b. We again train the imputation predictor for verification and show the results in Figure 6e. We confirm that our strategy lies significantly closer to the optimal, Minimally Revealing Imputation. Admittedly, there are even more sophisticated imputation strategies, for example building on Generative Adversarial Networks (GANs) such as Generative Adversarial Imputation Nets (GAIN) proposed by Yoon et al. (2018). However, our strategy already achieves considerable improvements and is highly efficient, because it does not require training of a GAN model. For completeness, we include additional experiments with GAN imputation in Appendix B.

6. Experiments

Having established that our Noisy Linear Imputation fulfills its purpose, in this section, we show that it entails even more benefits in practice. We first highlight how it makes results among different evaluation strategies more consistent in Section 6.1. We then present another considerable advantage in Section 6.2: its agreement with a no-retraining evaluation

[3]In fact, for direct and indirect neighbors, $\rho{=}0.89$ and $\rho{=}0.82$ respectively on CIFAR-10



(a) Original (b) Fixed Imp. (c) Noisy Lin. Imp.

(d) Graphical model used to derive our imputation

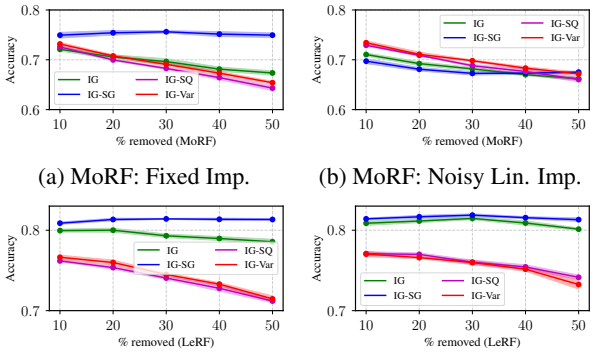
(e) Misclassification rate of the imputation predictor for different shares of randomly imputed pixels (CIFAR-10)

Figure 6. The considered imputation operators. When 50 % of the original image (a) are removed, they can either be imputed by a fixed value (b) or by our proposed Noisy Linear strategy (c,d). Training of an imputation predictor (e) shows that it is much harder to tell which pixels are original and which were imputed when using our proposed imputation model. This is closer to the optimal, minimally revealing imputation (black). Hence, by using imputed samples of this kind, Class Information Leakage is reduced.

strategy is sufficiently high, so that the retraining step is no longer required. We name this debiased and no-retraining evaluation framework ROAD (RemOve And Debias). All experiments in this section were conducted on CIFAR-10 using the eight attribution methods mentioned. We also use Food-101 (Bossard et al., 2014), a large-scale dataset of high-resolution images, to validate the generalizability of our method. To this end, we train over 1000 models from scratch on data imputed using the strategies, explanations and removal percentages. Since the results on Food-101 also support the findings from CIFAR-10, we include them in Appendix D.

6.1. Consistency under Removal Orders

As we aim for evaluation strategies that are less prone to the hyperparameter setting and allow for a consistent ranking, we study the consistency of evaluation results under the different removal orders MoRF and LeRF. Figure 7 depicts the obtained curves (using "Retrain"). For a clear view, we only show four curves of attribution methods based on IG with retraining and up to 50% pixels are removed. We include the full curves for the IG with its derivatives as well as GB with derivatives in Appendix C. The results using the common fixed value imputation shown in Figure 7a and



(a) MoRF: Fixed Imp. (b) MoRF: Noisy Lin. Imp.

(c) LeRF: Fixed Imp. (d) LeRF: Noisy Lin. Imp.

Figure 7. Consistency comparison using fixed value vs. Noisy Linear Imputation. The higher accuracy is better in LeRF, while the lower is better in MoRF. Comparing (a) and (c), fixed value imputation gives different rankings in MoRF and LeRF orders: IG-SG is the best in LeRF but the worst in MoRF. Comparing (b) and (d), Noisy Linear Imputation changes the outcome considerably and yields a consistent ranking in MoRF and LeRF.

Figure 7c. The results with our Noisy Linear Imputation are shown in Figure 7b and Figure 7d. In MoRF, a sharp drop in the beginning indicates a better attribution method, while a slight drop is desirable in LeRF. Hence, using fixed imputation, the ranking in MoRF is IG, IG-Var, IG-SQ, IG-SG, whereas the ranking in LeRF is IG-SG, IG, IG-SQ, and IG-Var. We see, for instance, that IG-SG is the worst in MoRF and the best in LeRF. When using the Noisy Linear Imputation, the inconsistency vanishes. The ranking in MoRF is: IG-SG, IG, IG-SQ, and IG-Var, which is the same as in LeRF.

We quantitatively compute the consistency among all eight attribution methods with and without retraining. Concretely, we compute the ranks (from 1=best to 8=worst) of our explanation methods for each percentage of perturbed pixels. We then calculate the Spearman Rank correlation between different evaluation strategies. As shown in Table 2, the correlation score of the fixed value imputation is -0.01 when using retraining and 0.01 when no retraining is applied. This indicates no consistency in the rankings. When we deploy our Noisy Linear Imputation, the results change drastically: The correlation score is improved to 0.61 and 0.58 with and without retraining, respectively. This might imply that the information leakage is responsible for a major share of the inconsistency.

6.2. Efficiency

When we apply our Noisy Linear Imputation, we additionally reduce the difference between evaluation with and without retraining. This can be attributed to the reduced distribution shift incurred when using an almost *Minimally Revealing Imputation*. If all pixels were perfectly imputed,

Retrain MoRF vs. LeRF		No-Retrain MoRF vs. LeRF	
fixed	lin	fixed	lin
-0.01±0.01	**0.61**±0.01	0.01±0.00	**0.58**±0.01

Table 2. Spearman rank correlation between evaluation strategies. There is almost no agreement between MoRF and LeRF when using fixed imputation (as in previous works). When using our imputation ("lin"), consistency across MoRF and LeRF orders increases drastically.

MoRF Retain vs. No-Retr.		LeRF Retain vs. No-Retr.	
fixed	lin	fixed	lin
0.15±0.01	**0.84**±0.01	0.09±0.01	**0.94**±0.01

Table 3. Spearman rank correlation between evaluation with and without retraining. Our Noisy Linear Imputation ("lin") also results only in marginal differences between "Retrain" and "No-Retrain". We conclude that the retraining step is no longer necessary.

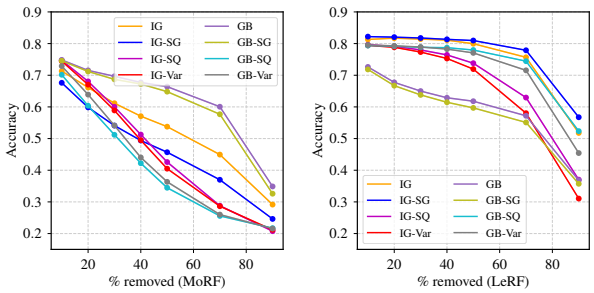
the resulting image would not be out-of-distribution. Since we are interested in the rankings of attribution methods, we again compute Spearman correlation between the rankings obtained with and without retraining and show it in Table 3. The order remains almost always intact between the "Retrain" with Noisy Linear Imputation and the "No-Retrain" variant with Noisy Linear Imputation resulting in a rank correlation of 0.84 in using MoRF and 0.94 in LeRF. This leads us to the conclusion that "No-Retrain" and "Retrain" end up with a highly similar ranking when using Noisy Linear Imputation. Thus, we conclude that the retraining step is not longer justified and can be skipped without significant distortion of the results. Qualitative results are shown in Appendix C.3, cf. Figure 17 (CIFAR-10) and Figure 23 (Food-101).

These results allow us to introduce a novel evaluation framework. We refer to the removal with Noisy Linear Imputation and no retraining as ROAD – Remove and Debias. We showed that ROAD is highly consistent with the compensated results of the ROAR, but comes at an enormous advantage: The retraining step is no longer required. This permits to save a vast amount of computation time. In our experiments, evaluation using the ROAD took only $0.7\,\%$ of the resources required for ROAR, as given by the runtimes in Table 4 obtained on the same hardware (single Nvidia GTX 2080Ti and 8 Cores).

In the end, we illustrate the evaluation results using ROAD among all eight attribution methods in MoRF and LeRF in Figure 8. In MoRF, the best ones are IG-SG, GB-SQ, GB-Var and IG, which have lower accuracies in the beginning, whereas they have higher accuracies in LeRF.

Strategy	Retrain		No-Retrain	
	fixed†	lin	fixed	lin⋆
Time	3903±117 s	4686±2 s	18.0±0.1 s	33.3±0.1 s
Relative	100 %	120 %	0.5 %	0.9 %

Table 4. Mean runtime (5 runs) for evaluating a single explanation method (IG). † refers to ROAR, and ⋆ to our ROAD.



(a) MoRF (b) LeRF

Figure 8. Evaluation results in MoRF (a) and LeRF (b) using our ROAD framework.

GB and GB-Var both perform badly in MoRF and LeRF. We see that some inconsistencies still remain, which cannot be compensated by the current imputation. However, the evaluation strategies might also consider different characteristics of an attribution method (e.g., one might be particularly good at identifying irrelevant pixels), which is why perfect agreement might not even be desirable.

7. Conclusion and Outlook

We introduced ROAD, an evaluation approach for measuring global fidelity among attribution explanations. ROAD comes with two key advantages over existing methods: (1) it is highly efficient, e.g., permitting a 99% runtime reduction w.r.t. ROAR, and (2) it circumvents the Class Information Leakage issue, which was thoroughly analyzed in this work. We believe the ROAD framework will be beneficial to the research community because it unifies several methods and is more consistent under varying removal orders. Moreover, it is broadly accessible due to its low resource requirements. ROAD is open-source[4], and can be readily implemented in practical use-cases. Going forward, we plan to investigate more sophisticated imputation models in ROAD as well as other evaluation metrics besides fidelity.

Acknowledgements

We acknowledge the support by the Cluster of Excellence - Machine Learning: New Perspectives for Science, EXC number 2064/1 - Project number 390727645, and the support of the Training Center for Machine Learning (TCML) Tübingen, funded by the German Federal Ministry of Education and Research (BMBF) with grant number 01IS17054, which provided substantial resources for running our large-scale Food-101 experiment.

[4]An official implementation is also included in the Quantus framework (Hedström et al., 2022)

References

Adadi, A. and Berrada, M. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018.

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Adebayo, J., Muelly, M., Liccardi, I., and Kim, B. Debugging tests for model explanations. *arXiv preprint arXiv:2011.05429*, 2020.

Afchar, D. and Hennequin, R. Making neural networks interpretable with attribution: application to implicit signals prediction. In *Fourteenth ACM Conference on Recommender Systems*, pp. 220–229, 2020.

Afchar, D., Guigue, V., and Hennequin, R. Towards rigorous interpretations: a formalisation of feature attribution. In *International Conference on Machine Learning*, pp. 76–86. PMLR, 2021.

Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*, 2017.

Bhatt, U., Weller, A., and Moura, J. M. Evaluating and aggregating feature-based model explanations. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 3016 –3022, 2020a.

Bhatt, U., Xiang, A., Sharma, S., Weller, A., Taly, A., Jia, Y., Ghosh, J., Puri, R., Moura, J. M., and Eckersley, P. Explainable machine learning in deployment. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 648–657, 2020b.

Bossard, L., Guillaumin, M., and Gool, L. V. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pp. 446–461. Springer, 2014.

Chen, J., Song, L., Wainwright, M. J., and Jordan, M. I. L-shapley and c-shapley: Efficient model interpretation for structured data. In *International Conference on Learning Representations*, 2018.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. John Wiley and Sons, 2006. doi: 10.1002/047174882X.

Covert, I., Lundberg, S., and Lee, S.-I. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90, 2021.

Doshi-Velez, F. and Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Eitel, F., Ritter, K., Alzheimer's Disease Neuroimaging Initiative (ADNI), et al. Testing the robustness of attribution methods for convolutional neural networks in mri-based alzheimer's disease classification. In *Interpretability of Machine Intelligence in Medical Image Computing and Multimodal Learning for Clinical Decision Support*, pp. 3–11. Springer, 2019.

Fel, T., Vigouroux, D., Cadène, R., and Serre, T. How good is your explanation? algorithmic stability measures to assess the qualityof explanations for deep neural networks. 2021.

Fong, R. C. and Vedaldi, A. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pp. 3429–3437, 2017.

Hartley, T., Sidorov, K., Willis, C., and Marshall, D. Explaining failure: Investigation of surprise and expectation in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 12–13, 2020.

Hase, P. and Bansal, M. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5540–5552, 2020.

Haug, J., Zürn, S., El-Jiz, P., and Kasneci, G. On baselines for local feature attributions. *AAAI Workshop on Explainable Agency in AI Workshop*, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hedström, A., Weber, L., Bareeva, D., Motzkus, F., Samek, W., Lapuschkin, S., and Höhne, M. M.-C. Quantus: an explainable AI toolkit for responsible evaluation of neural network explanations. *arXiv preprint arXiv:2202.06861*, 2022.

Hellman, M. E. and Raviv, J. Probability of Error, Equivocation, and the Chernoff Bound. *IEEE Transactions on Information Theory*, 16(4):368–372, 1970.

Hooker, S., Erhan, D., Kindermans, P. J., and Kim, B. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Izzo, C., Lipani, A., Okhrati, R., and Medda, F. A baseline for shapely values in mlps: from missingness to neutrality. *arXiv preprint arXiv:2006.04896*, 2020.

Jethani, N., Sudarshan, M., Aphinyanaphongs, Y., and 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.

Kachuee, M., Karkkainen, K., Goldstein, O., Darabi, S., and Sarrafzadeh, M. Generative imputation and stochastic prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Kasneci, G. and Gottron, T. Licon: A linear weighting scheme for the contribution ofinput variables in deep artificial neural networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 45–54, 2016.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lapuschkin, 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.

Lundberg, S. M. and 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.

Meng, C., Trinh, L., Xu, N., and Liu, Y. Mimic-if: Interpretability and fairness evaluation of deep learning models on mimic-iv dataset. *arXiv preprint arXiv:2102.06761*, 2021.

Meyen, S. Relation between classification accuracy and mutual information in equally weighted classification task. Master's thesis, University of Hamburg, 2016. URL https://osf.io/zru7b/.

Nauta, M., Trienes, J., Pathak, S., Nguyen, E., Peters, M., Schmitt, Y., Schlötterer, J., van Keulen, M., and Seifert, C. From anecdotal evidence to quantitative evaluation

methods: A systematic review on evaluating explainable ai. *arXiv preprint arXiv:2201.08164*, 2022.

Nguyen, A. P. and Martínez, M. R. On quantitative aspects of model interpretability. *arXiv preprint arXiv:2007.07584*, 2020.

Petsiuk, V., Das, A., and Saenko, K. Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

Ribeiro, M. T., Singh, S., and 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.

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, 2016.

Schramowski, P., Stammer, W., Teso, S., Brugger, A., Herbert, F., Shao, X., Luigs, H.-G., Mahlein, A.-K., and Kersting, K. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, 2020.

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 *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Shah, H., Jain, P., and Netrapalli, P. Do input gradients highlight discriminative features?, 2021.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pp. 3145–3153. PMLR, 2017.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., and Wattenberg, M. Smoothgrad: removing noise by adding noise. In *Workshop on Visualization for Deep Learning, ICML*, 2017.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*, 2015.

Srinivas, S. and Fleuret, F. Full-gradient representation for neural network visualization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Sturmfels, P., Lundberg, S., and Lee, S.-I. Visualizing the impact of feature attribution baselines. *Distill*, 5(1):e22, 2020.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pp. 3319–3328. PMLR, 2017.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pp. 1139–1147. PMLR, 2013.

Tjoa, E. and Guan, C. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Tomsett, R., Harborne, D., Chakraborty, S., Gurram, P., and Preece, A. Sanity checks for saliency metrics. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6021–6029, 2020.

Vergara, J. R. and Estévez, P. A. A review of feature selection methods based on mutual information. *Neural Computing and Applications*, 2014.

Xu, S., Venugopalan, S., and Sundararajan, M. Attribution in scale and space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9680–9689, 2020.

Yeh, C.-K., Hsieh, C.-Y., Suggala, A., Inouye, D. I., and Ravikumar, P. K. On the (in) fidelity and sensitivity of explanations. *Advances in Neural Information Processing Systems*, 32:10967–10978, 2019.

Yoon, J., Jordon, J., and Schaar, M. Gain: Missing data imputation using generative adversarial nets. In *International Conference on Machine Learning*, pp. 5689–5698. PMLR, 2018.

A. Additional Theory

A.1. Formulation of the MI Bounds for the Binary Case

As we discussed in our main paper, the relationship between Mutual Information (MI) and accuracy is not a function, but comes in form of upper and lower bounds of the obtainable accuracy. If, for example, the binary classification case with equal class priors $p(C=0) = p(C=1) = \frac{1}{2}$ is considered, the following bounds can be derived (Hellman & Raviv, 1970; Meyen, 2016):

$$\frac{I(\boldsymbol{x}; C) + 1}{2} \leq \mathrm{Acc}(C|\boldsymbol{x}) \leq H_2^{-1}(1 - I(\boldsymbol{x}; C)), \tag{5}$$

where $H_2^{-1} : [0,1] \to \left[\frac{1}{2}, 1\right]$ is the inverse of the binary entropy with support $\left[\frac{1}{2}, 1\right]$. For completeness, we restate the proof of this upper bound in Appendix A.2.

A.2. Reproduction of the proof of the relation between mutual and accuracy in the binary case

In this section, we reproduce the proofs for the upper and lower bounds of bayesian classifier accuracy given a certain amount of mutual information from the master's thesis by (Meyen, 2016) for completeness. The upper bound given there is tighter than the bounds present in the literature.
We consider the following setting (C, $\boldsymbol{x}$ are random variables):

- binary classification problem, $C \in \Omega_C = \{0, 1\}$
- equal class priors $P(C=0) = \frac{1}{2}, P(C=1) = \frac{1}{2}$
- discrete features $\boldsymbol{x}$ (which can be the product of multiple random variables)
- support set $\Omega_x = \mathrm{supp}\,\{\boldsymbol{x}\}$ of *countable* size

We first prove the following Lemma:

Lemma A.1. *Let the assumptions stated above be true. Then, the mutual information is the weighted mean of a function of the conditional accuracies* $\mathrm{Acc}(C|s)$*, where* $s \in \Omega_x$*:*

$$I(C; \boldsymbol{x}) = \sum_{s \in \Omega_S} p(s) \left(1 - H_2\left[\mathrm{Acc}(C|s)\right]\right)$$

In this formulation, $p(s)$ is a shorthand for $P(\boldsymbol{x} = s)$ and $H_2(p) := -p \log p - (1-p)\log(1-p)$ is the entropy for a binary random variable.
Proof.

$$\begin{aligned}
I(C; \boldsymbol{x}) &= H(C) - H(C|\boldsymbol{x}) && (6)\\
&= \sum_{c \in \Omega_C} p(c) \log \frac{1}{p(c)} - \sum_{s \in \Omega_S} p(s) \sum_{c \in \Omega_C} p(c|s) \log \frac{1}{p(c|s)} && (7)\\
&= \sum_{s \in \Omega_x} p(s) \left[\sum_{c \in \Omega_C} p(c) \log \frac{1}{p(c)} - \sum_{c \in \Omega_C} p(c|s) \log \frac{1}{p(c|s)} \right] && (8)\\
&= \sum_{s \in \Omega_x} p(s) \left[H(C) - H(C|s)\right] && (9)
\end{aligned}$$

In our consideration, $\Omega_C = \{0, 1\}$ and $P(C=0) = \frac{1}{2}, P(C=1) = \frac{1}{2}$, so $H(C) = 1$. Additionally, the bayesian classifier rule yields

$$acc(C|s) = \begin{cases} P(C=0|s), & \text{for } P(C=1|s) \leq 0.5 \\ P(C=1|s), & \text{for } P(C=1|s) > 0.5 \end{cases} \tag{10}$$

and

$$H(C|s) = -P(C=0|s)\log P(C=0|s) - P(C=1|s)\log P(C=1|s) \tag{11}$$
$$= H_2(P(C=0|s)) = H_2(P(C=1|s)) \tag{12}$$
$$= H_2(acc(C|s)) \tag{13}$$

Plugging in the results $H(C) = 1$ and $H(C|s) = H_2(\mathrm{Acc}(C|s))$, we obtain the proposed lemma. □

For the derivation of upper and lower bounds, Jenssen's inequality is used. $1 - H_2(\cdot)$ is a convex function and the $\{p(s)\}_{s\in\Omega_x}$ are convex multipliers, i.e., they are non-negative and sum up to one. Then,

$$1 - H_2\left(\mathrm{Acc}(C|\boldsymbol{x})\right) = 1 - H_2\left(\sum_{s\in\Omega_x} p(s)\,\mathrm{Acc}(C|s)\right) \tag{14}$$
$$\leq \sum_{s\in\Omega_x} p(s)\left[1 - H_2\left(\mathrm{Acc}(C|s)\right)\right] = I(\boldsymbol{x};C) \tag{15}$$

We can restate this equation in terms of accuracy.

$$H_2\left(\mathrm{Acc}(C|\boldsymbol{x})\right) \geq 1 - I(C;\boldsymbol{x}) \tag{16}$$

Using that $H_2(\cdot)$ is decreasing monotonically on the interval $\left[\frac{1}{2}, 1\right]$, so its inverse H_2^{-1} exists, and that $\mathrm{Acc}(C|s) \geq 0.5$:

$$\mathrm{Acc}(C|\boldsymbol{x}) \leq H_2^{-1}\left(1 - I(C;\boldsymbol{x})\right). \tag{17}$$

The inequality sign is flipped again, due to the inverse being monotonically decreasing. Note that the bounds derived for the special case are much tighter than the general ones provided by Vergara & Estévez (2014) and Cover & Thomas (2006, Chapter 2.10), that are not of any use, because they are even less strict than the trivial bound $\mathrm{Acc}(C|\boldsymbol{x}) \leq 1$, for the simple case considered here.

For the lower bound, we refer the reader to Hellman & Raviv (1970, eqn. 18), where the term I corresponds to $H(C|\boldsymbol{x}) = H(C) - I(C;\boldsymbol{x})$ in our notation. Rewriting the result from Hellman & Raviv (1970) in our notation, we obtain

$$1 - \mathrm{Acc}(C|\boldsymbol{x}) \leq \frac{H(C) - I(C;\boldsymbol{x})}{2}. \tag{18}$$

Using $H(C) = 1$ and rearranging yields

$$1 - \mathrm{Acc}(C|\boldsymbol{x}) \leq \frac{1 - I(C;\boldsymbol{x})}{2} \tag{19}$$

and

$$\mathrm{Acc}(C|\boldsymbol{x}) \geq \frac{I(C;\boldsymbol{x}) + 1}{2}. \tag{20}$$

□

A.3. Analysis of the LeRF Ordering

In this section, we analyze the masking impact for the case of the Least Relevant First (LeRF) ordering. We first provide a definition for the operators involved as we did for the Most Relevant First (MoRF) case. In the LeRF setting, the k least important important features per instance are removed. We model the explanation as a choice of features via a binary mask $\boldsymbol{M} = \boldsymbol{e}(f, \boldsymbol{x}) \in \{0,1\}^d$, with the corresponding value set to one, if the corresponding feature is among the top-k, and to zero otherwise. Furthermore, suppose $\mathcal{M}_h : \{0,1\}^d \times \mathbb{R}^d \to \mathbb{R}^k$ to be the selection operator for the **h**ighly important dimensions indicated in the mask and $\boldsymbol{x}_h = \mathcal{M}_h(\boldsymbol{M}, \boldsymbol{x})$ to be a vector containing only the remaining, highly important features as shown in Figure 9. We suppose that the features preserve their internal order in $\boldsymbol{x}_h$, i.e., features are ordered ascendingly by their original input indices.

The LeRF approach with retraining (also called "Keep and Retrain", KAR, by Hooker et al. (2019)) measures the accuracy of a newly trained classifier f' on modified samples $\boldsymbol{x}'_h := \mathcal{I}_h(\boldsymbol{M}, \boldsymbol{x_h})$, where $\mathcal{I}_h : \{0,1\}^d \times \mathbb{R}^k \to \mathbb{R}^d$ is an imputation

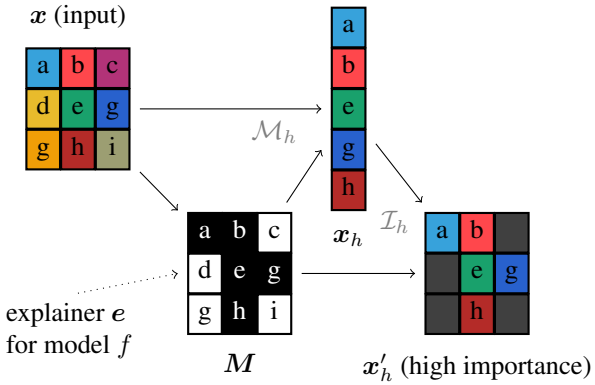


Figure 9. Analogous analytical model of feature removal in the opposite order (LeRF): The input image $\boldsymbol{x}$ is explained by an explanation method that returns a mask $\boldsymbol{M}$ indicating important pixels. The remaining, highly important pixels can be extracted from the image using the masking operator $\mathcal{M}_h$ and transformed to a modified variant of the input $\boldsymbol{x}'_h$ via the imputation operator $\mathcal{I}_h$.

operator that redistributes all inputs in the vector $\boldsymbol{x}_h$ to their original positions and sets the remainder to some filling value. This means only the top-k features are kept. For a better evaluation result, the accuracy should increase quickly with increasing k, indicating the most influential features are present. Accuracy should not increase much for the high values of k, because inserting the low importance features should not have a large effect (equivalently, this means it should not drop much when the least important features are removed). Overall, higher accuracies indicate better attributions in the LeRF setting.

For the LeRF benchmark, the quantity of interest in our analysis will be $I(\boldsymbol{x}'_h; C)$, the class information contained in the filled-in version of the selected high important features. We want to maximize $I(\boldsymbol{x}'_h; C)$ to obtain a good score,

$$\uparrow I(\boldsymbol{x}'_h; C) \;\; \Rightarrow \uparrow \text{LeRF benchmark.}$$

As before, we can apply the following, general identity:

$$\underbrace{I(\boldsymbol{x}'_h; C)}_{\text{Evaluation Outcome}} = \underbrace{I(C; \boldsymbol{x}'_h|\boldsymbol{M})}_{\text{Feature Info.}} + \underbrace{I(C; \boldsymbol{M})}_{\text{Mask Info.}} - \underbrace{I(C; \boldsymbol{M}|\boldsymbol{x}'_h)}_{\text{Mitigator}}. \tag{21}$$

The interpretation of the terms is analogous to that in our main paper.

Class-Leaking Explanation Map For the case of the class-leaking map, we again require the imputation operator to be invertible:

Example A.2. *Invertible Imputation. Let $\mathcal{I}_h : \{0,1\}^d \times \mathbb{R}^k \to \mathbb{R}^d$ be the imputation operator that takes the highly important features as an input. We suppose that there are inverse functions $\mathcal{I}^{-1}_{h,M}$ and $\mathcal{I}^{-1}_{h,x}$, such that*

$$\boldsymbol{x}'_h = \mathcal{I}_h\left(\boldsymbol{M}, \boldsymbol{x}_h\right) \Leftrightarrow \boldsymbol{M} = \mathcal{I}^{-1}_{h,M}(\boldsymbol{x}'_h) \wedge \boldsymbol{x}_h = \mathcal{I}^{-1}_{h,x}(\boldsymbol{x}'_h).$$

If, for instance, the pixels removed are set to some reserved value indicating their absence, the infilling operator is invertible. In this case, also the Mitigator $I\left(C; \boldsymbol{M}|\boldsymbol{x}'_h\right) = 0$ (see Section 4.3 for details). The "Feature Info" term is constrained to be positive. Thus, the Mask Information has a non-negligible impact on the Evaluation Goal, because a higher Mask term will always increase it.

We can create a another example of a spurious explanation map that shows how evaluation scores are influenced even worse for LeRF: Suppose an explanation map that starts masking out pixels at the top for class zero and at the bottom for class one. Thus, a retrained model will be able to infer the category just from the shape of the masked pixels and obtain the best possible accuracy and thus score in the LeRF setting. However, it does not provide a reasonable attribution for the importance of the features.

B. GAN Imputation

We also use Generative Adversarial Imputation Nets (GAIN) proposed by Yoon et al. (2018) as an imputation operator. We first train a GAIN model on CIFAR-10. To find the best-performing setup, we run a hyperparameter selection for the GAIN model. We keep all the default parameters identified by Kachuee et al. (2020), but search for the value of `alpha` (α), which can be seen as a weight factor for the reconstruction loss of the non-imputed pixels in the GAN, and the `hint_rate` (hr) parameter, which provides the Discriminator with hints to balance the difficulty of the tasks. We train the models for 100 epochs which resulted in converged MSEs and Frechet Inception Distances (FIDs). We use MSE to the original pixels to assess the generative quality of the model. Kachuee et al. (2020) reported low values for both these parameters to perform well, but did not provide the exact values. We extended their value ranges to $\alpha = 100$ and performed and exhaustive search. The results for the GAIN models on CIFAR-10 can be seen in Table 5. For the experiments we used the best setup with $\alpha = 100$ and $hr = 0.01$.

	α=0.1	α=1	α=10	α=100
hr=0.01	0.0131	0.0164	0.0090	**0.0085**
hr=0.1	0.0113	0.0133	0.0131	0.0101
hr=0.3	0.0172	0.0183	0.0151	0.0127
hr=0.9	0.0303	0.0484	0.0379	0.0088

Table 5. Mean-Squared-Errors for GAIN on CIFAR-10 using different hyperparameter choices.

In Figure 10, we demonstrate imputation results using three operators for one image (a) from CIFAR-10. Compared to the fixed value imputation (b) and noisy linear imputation (c), GAN imputation (d) yields most natural imputed image. Although it cannot perfectly reconstruct the original image, for example the background is noisy and the body color is different from the original one, it is not easy to deduce the mask from (d). A trained imputation predictor also verifies that GAN imputation is closest to the optimal condition, Minimally Revealing Imputation.

However, there are drawbacks of the GAN imputation. It may introduce some new "features" that do not exist in the original sample. For instance the dog in (d) has new patterns on its body. Moreover, it does not give very good results when too many pixels are removed (cf. Figure 12). The GAIN training again requires tuning hyperparameter settings and is highly expensive. Therefore, this model does not allow for the desired improvements (few hyperparameters, efficiency). Compared to GAN, our Noisy Linear imputation does not have these drawbacks. Considering all these factors, we recommend to use Noisy Linear Imputation in the evaluation framework.

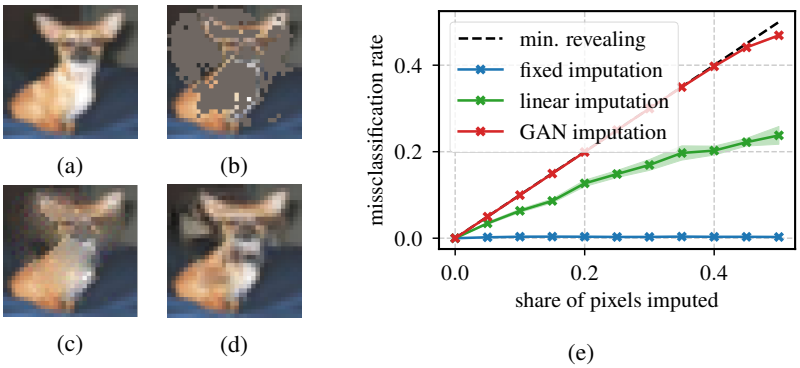


Figure 10. The considered imputation operators. When 30 % of the original image (a) are removed, they can either be completed by a fixed value (b) or by our proposed Noisy Linear imputation (c) or GAN imputation (d). Training of an imputation predictor (e) shows that it is much harder to tell which pixels are original and which were imputed when using our proposed imputation models, which is closer to the theoretical optimum (black). Hence, Class Information Leakage is reduced by our imputation methods.

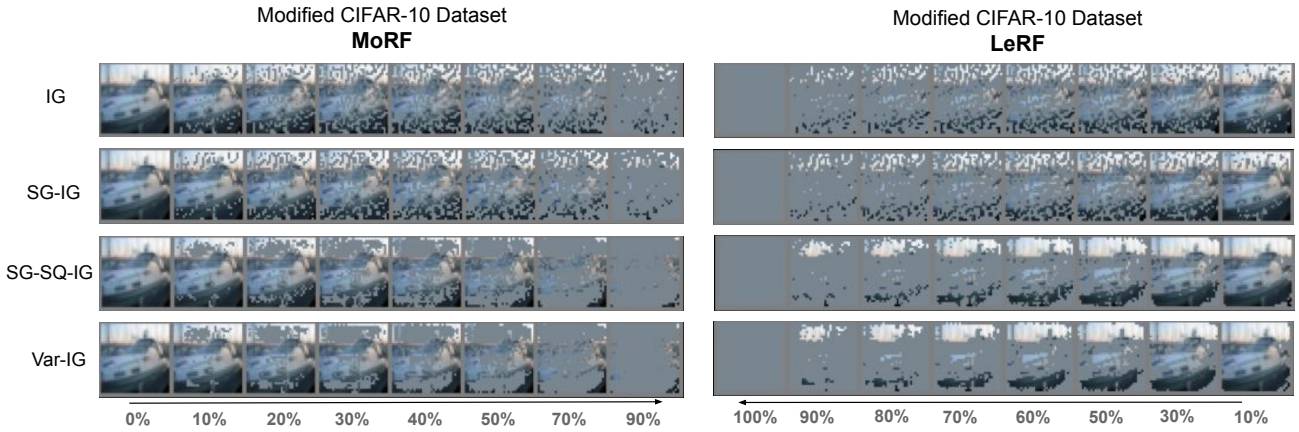


Figure 11. Illustration of modified data set in MoRF/LeRF and fixed value imputation settings. **Left**: Modifications in the MoRF framework. **Right**: Modifications in the LeRF framework. **Top to Bottom**: Modifications using Integrated Gradient (IG) (Sundararajan et al., 2017) and three ensemble variants of IG: SmoothGrad (SG-IG) (Smilkov et al., 2017), SmoothGrad^2 (SG-SQ-IG) (Hooker et al., 2019), and VarGrad (Var-IG) (Adebayo et al., 2018). The percentage of pixels that are removed or kept is given at the bottom.

C. Additional Experiments on CIFAR-10

C.1. Implementation Details

In this section, we report implementation details on CIFAR-10 as well as additional results for comparison between fixed value imputation and our *Noisy Linear Imputation*. We also include GAN imputation results. In Figure 12, an overview of using three different imputations with different perturbation percentages are illustrated.

We train a vanilla ResNet-18 (He et al., 2016) on CIFAR-10 and compute different explanations using the trained model. The model is trained with the initial learning rate of 0.01 and the SGD optimizer (Sutskever et al., 2013). We decrease the learning rate by factor 0.1 after 25 and train the model for 40 epochs on one GPU. The trained model achieves a test set accuracy of $84.5\,\%$ (comparable to the model in (Tomsett et al., 2020)). For attributions, we use the same settings as in (Hooker et al., 2019): As base explanations we implement Integrated Gradient (IG) (Sundararajan et al., 2017) and Guided Backprop (GB) (Springenberg et al., 2015). Additionally, we use three ensembling strategies for each: SmoothGrad (SG) (Smilkov et al., 2017), SmoothGrad^2 (SG-SQ) (Hooker et al., 2019) and VarGrad (Var) (Adebayo et al., 2018). For each explanation method, we modify the data set using the fraction of pixels $\eta = [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9]$. Figure 11 illustrates the modified images by using four different explanations in the GB-family within MoRF and LeRF orders (fixed mean value imputation is used).

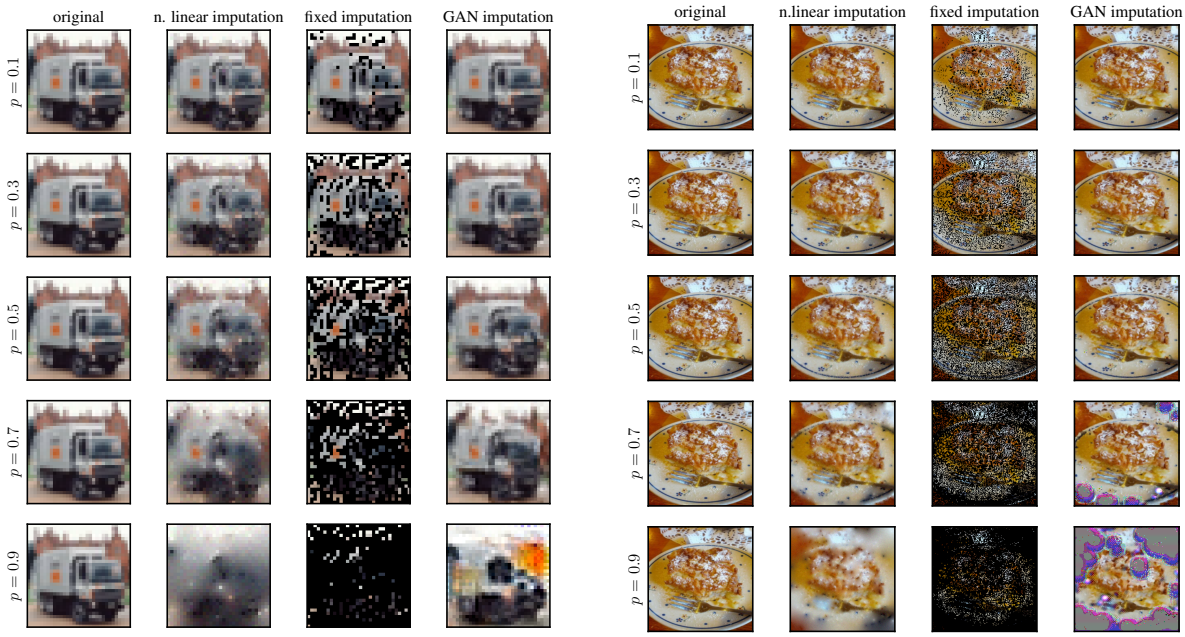
We use $N = 5$ **runs** and report averaged results for all CIFAR-10 experiments in our paper and indicate the standard errors (which are very small) as an area behind our plots. In Table 6 and Table 7, we show the mean accuracy and its standard deviation at each the fraction of pixels η for IG-SG and GB-SG explanations. For other explanations we used, the standard deviation at each η in the magnitude of below one percent as well. Mean runtimes (average over 5 runs) for evaluating one explanation method (IG) using all three imputation methods are listed in Table 8.

C.2. Correlation Analysis

In Table 9, we show a full view of the Spearman Correlation of rankings between all twelve different evaluation strategies ("Retrain"/"No-Retrain", MoRF/LeRF, and fixed value/Noisy Linear/GAN imputation) used in this paper. In this work, our primary focus was on consistency between the respective Retraining/No-Retraining Methods and the consistency between MoRF/LeRF and we mark the results used in the main paper in bold.

C.3. Extended Figures

In this section, we include full qualitative results of using four variants in evaluation strategies ("Retrain"/"No-Retrain", MoRF/LeRF) for three different imputation operators (fixed value/Noisy Linear/GAN imputation). In Figure 13, the full



(a) Sample from CIFAR-10

(b) Sample from Food-101

Figure 12. Sample images from CIFAR-10 and Food-101 imputed with the three methods considered in this work for different percentages. The missing pixels are determined by the IG attribution method (in MoRF order). While the GAN leads to sharper images for the early percentage values, where the linearly imputed samples become more blurry. Artefacts are introduced for high missingness percentages (0.9) in GAN imputation, which may distort the results of the evaluation once again. Therefore, we decide to stick to the Noisy Linear Imputation that operates more stably.

plots of IG-family attribution methods using fixed value imputation are shown, while Figure 16 illustrates for the GB-based attribution methods. Figure 14 and Figure 17 show the evaluation results when using our Noisy Linear Imputation for IG- and GB-family attribution methods, respectively. From results, we see that using our Noisy Linear Imputation, the consistency between the evaluation rankings conducted in MoRF and LeRF with and without retraining increases, for instance in Figure 14 compared to Figure 13.

D. Additional Experiments on Food-101

D.1. Implementation Details

We trained a vanilla ResNet-50 (He et al., 2016) on Food-101 (Bossard et al., 2014). Concretely, we trained the model using the SGD optimizer. Additionally the model was trained with the initial learning rate of 0.01. The learning rate was reduced by factor of 0.1 after every 10 epochs. In total, we trained 40 epochs with a batch size of 32 and the model achieved the accuracy of 81.67% on the test set. To run the GAN imputation operator, we first trained a GAIN model on Food-101 as introduced in Appendix B. We used the hyper-parameters $\alpha = 100$ and $hr = 0.1$ and trained the GAIN model with the batch size of 32 for 100 epochs. We computed the eight explanations and run ROAD and ROAR evaluation using the same settings as introduced in Appendix C.1 for CIFAR-10.

D.2. Correlation Analysis

In Table 10, we show a full view of Spearman Correlation of rankings given by eight different evaluation strategies ("Retrain"/"No-Retrain", MoRF/LeRF, and fixed/Noisy Linear/GAN imputation) on Food-101. In the table, results marked in bold indicate the consistency of using three imputation operators. We observe that the consistency between the respective Retrain and No-Retrain methods is still very high, which confirms that the efficiency gains reported in the main paper can be realized for larger data sets. Consistency between MoRf/LeRF is improved (over fixed imputation) when using retraining, but decreases slightly when the No-Retraining approach is used. Because the curves are often very close on this dataset

		10	20	30	40	50	70	90
Retrain MoRF	fixed	74.94±0.57	75.42±0.45	75.62±0.24	75.16±0.50	74.95±0.45	73.73±0.48	65.18±0.85
	lin	69.72±0.49	68.10±0.34	67.28±0.34	67.32±0.22	67.52±0.22	66.46±0.54	60.37±0.51
	gan	74.78±0.31	73.16±0.22	72.02±0.03	71.40±0.23	70.72±0.30	68.44±0.43	59.37±0.44
No-Retrain MoRF	fixed	44.06±0.04	29.81±0.03	21.99±0.03	17.35±0.02	14.67±0.01	11.50±0.04	10.90±0.03
	lin	67.66±0.02	59.94±0.03	54.05±0.05	49.46±0.04	45.63±0.06	36.87±0.05	24.55±0.04
	gan	74.53±0.04	71.41±0.04	69.10±0.06	67.55±0.09	66.55±0.07	60.73±0.12	25.46±0.10
Retrain LeRF	fixed	80.88±0.14	81.34±0.15	81.41±0.01	81.36±0.14	81.34±0.11	80.95±0.01	76.86±0.34
	lin	81.41±0.10	81.67±0.18	81.88±0.16	81.56±0.13	81.31±0.22	79.89±0.23	72.83±0.36
	gan	81.05±0.22	80.99±0.15	80.14±0.16	79.25±0.18	78.24±0.22	74.92±0.15	68.69±0.21
No-Retrain LeRF	fixed	74.34±0.02	69.04±0.03	64.06±0.04	59.86±0.03	57.59±0.03	53.81±0.06	46.74±0.02
	lin	82.20±0.04	82.04±0.03	81.76±0.08	81.34±0.06	80.97±0.03	77.89±0.07	56.74±0.13
	gan	80.80±0.02	80.38±0.03	79.90±0.02	78.85±0.07	77.47±0.08	71.14±0.10	32.96±0.17

Table 6. Mean accuracy at each η by using IG-SG in all methods with standard deviations of five individual runs. For LeRF, the accuracy is at (1-η).

		10	20	30	40	50	70	90
Retrain MoRF	fixed	76.30±0.43	75.60±0.27	74.89±0.29	74.27±0.29	73.37±0.28	72.15±0.09	67.99±0.24
	lin	72.83±0.37	71.87±0.41	71.58±0.19	70.98±0.15	70.47±0.20	67.81±0.45	59.38±0.46
	gan	76.64±0.13	75.44±0.13	74.73±0.28	73.69±0.30	72.85±0.34	68.97±0.08	56.81±0.30
No-Retrain MoRF	fix	73.03±0.03	66.72±0.03	58.72±0.07	52.51±0.04	48.52±0.08	48.79±0.06	44.43±0.06
	lin	74.57±0.08	71.18±0.06	68.70±0.08	67.24±0.08	64.82±0.11	57.68±0.06	32.59±0.09
	gan	76.57±0.03	74.70±0.04	72.51±0.09	71.19±0.07	69.64±0.08	60.89±0.15	21.11±0.16
Retrain LeRF	fixed	72.39±0.39	71.76±0.41	71.21±0.30	70.26±0.50	69.83±0.22	68.32±0.45	63.29±0.56
	lin	72.86±0.24	71.63±0.27	70.67±0.42	70.08±0.30	69.82±0.22	68.10±0.18	60.12±0.34
	gan	75.97±0.27	74.73±0.27	73.41±0.24	72.74±0.34	72.20±0.28	69.89±0.26	57.57±0.24
No-Retrain LeRF	fixed	69.61±0.04	64.90±0.02	57.88±0.05	51.67±0.09	46.93±0.06	42.40±0.09	37.10±0.03
	lin	71.84±0.06	66.71±0.08	63.79±0.05	61.46±0.09	59.69±0.09	55.09±0.06	35.72±0.13
	gan	75.13±0.02	72.13±0.05	70.25±0.05	68.56±0.08	67.35±0.08	62.32±0.13	24.61±0.19

Table 7. Mean accuracy at each η by using GB-SG in all methods with standard deviations of five individual runs. For LeRF, the accuracy is at (1-η).

(in particular for the No-Retraining setup), small differences might already lead to a change in the ranking and the results are in general noisier than on CIFAR-10. In summary, we observe similar trends, although the consistency gain between MoRF/LeRF in No-Retrain is not as pronounced. Nevertheless, a perfect agreement between MoRF/LeRF might not be desirable.

D.3. Extended Figures

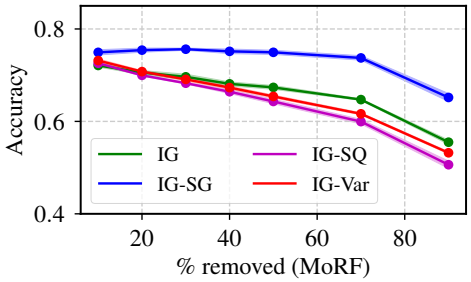
Full qualitative results of using four variants in evaluation strategies ("Retrain"/"No-Retrain", MoRF/LeRF) for three different imputation operators (fixed value/Noisy Linear/GAN imputation) are listed from Figure 19 to Figure 24. Figure 20 and Figure 23 show the evaluation results when using our Noisy Linear Imputation for IG- and GB-family attribution methods, respectively. From results, we see that using our Noisy Linear Imputation, the consistency between the evaluation results using "Retrain" and "No-Retrain" are more consistent compared to using the fixed value imputation. Therefore, retraining can be safely skipped by using our Noisy Linear Imputation.

Strategy	Retrain			No-Retrain		
	fixed†	lin	gan	fixed	lin*	gan
Time	3903±117 s	4686±2 s	6421±74 s	18.0±0.1 s	33.3±0.1 s	35.0±0.1 s
Relative	100 %	120 %	164 %	0.5 %	0.9 %	0.9 %

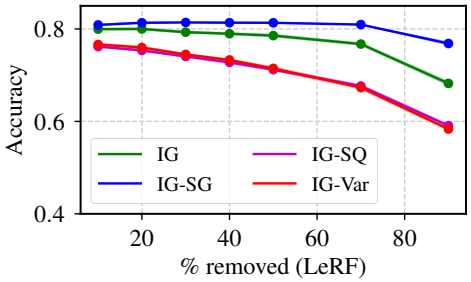
Table 8. Mean runtime (5 runs) for evaluating a single explanation method (IG) on three imputation operators. † refers to ROAR, and $\star$ to our ROAD.

		Retrain MoRF			No-Retrain MoRF			Retrain LeRF			No-Retrain LeRF		
		fixed†	lin	gan	fixed	lin*	gan	fixed	lin	gan	fixed	lin	gan
Retrain MoRF	fixed†	1.00 ±0.00											
	lin	0.68 ±0.02	1.00 ±0.00										
	gan	0.76 ±0.01	0.82 ±0.01	1.00 ±0.00									
No-Retrain MoRF	fixed	**0.15** ±0.01	0.38 ±0.02	0.23 ±0.01	1.00 ±0.00								
	lin*	0.66 ±0.01	**0.84** ±0.01	0.86 ±0.01	0.43 ±0.01	1.00 ±0.00							
	gan	0.65 ±0.01	0.62 ±0.01	0.84 ±0.01	0.14 ±0.01	0.78 ±0.01	1.00 ±0.00						
Retrain LeRF	fixed	**-0.01** ±0.01	0.48 ±0.02	0.28 ±0.02	0.66 ±0.00	0.47 ±0.02	0.13 ±0.01	1.00 ±0.00					
	lin	0.16 ±0.01	**0.61** ±0.01	0.34 ±0.01	0.78 ±0.01	0.50 ±0.01	0.10 ±0.01	0.87 ±0.01	1.00 ±0.01				
	gan	0.15 ±0.01	0.59 ±0.01	0.32 ±0.01	0.74 ±0.00	0.50 ±0.01	0.10 ±0.01	0.90 ±0.01	0.96 ±0.01	1.00 ±0.00			
No-Retrain LeRF	fixed	0.49 ±0.01	0.44 ±0.01	0.69 ±0.01	**0.01** ±0.00	0.60 ±0.00	0.77 ±0.00	**0.09** ±0.01	0.03 ±0.01	-0.03 ±0.00	1.00 ±0.00		
	lin	0.21 ±0.01	0.60 ±0.01	0.38 ±0.01	0.81 ±0.00	**0.58** ±0.01	0.22 ±0.01	0.85 ±0.00	**0.94** ±0.01	0.91 ±0.00	0.10 ±0.00	1.00 ±0.00	
	gan	0.05 ±0.01	0.47 ±0.01	0.17 ±0.01	0.69 ±0.00	0.36 ±0.00	-0.07 ±0.01	0.85 ±0.00	0.86 ±0.01	0.90 ±0.01	-0.14 ±0.00	0.79 ±0.00	1.00 ±0.00

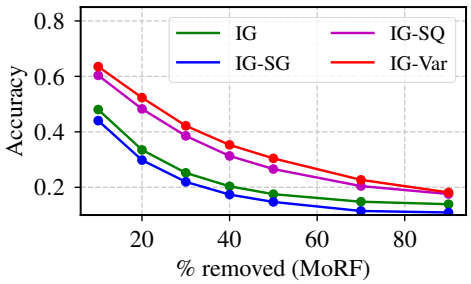
Table 9. **CIFAR-10**: Rank Correlations between all evaluation strategies used with standard deviations computed by considering the rankings obtained through five consecutive runs as independent. Results indicated in bold correspond to those reported in the main paper. The ROAR benchmark is marked by † and our ROAD by *.



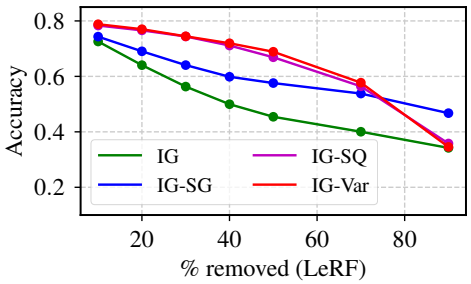
(a) MoRF, Retrain



(b) LeRF, Retrain

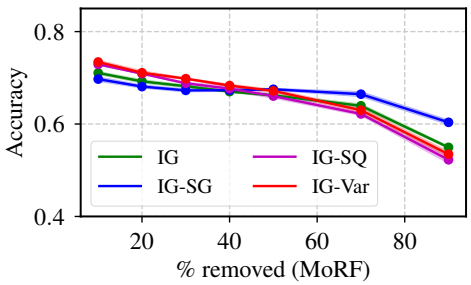


(c) MoRF, No-Retrain

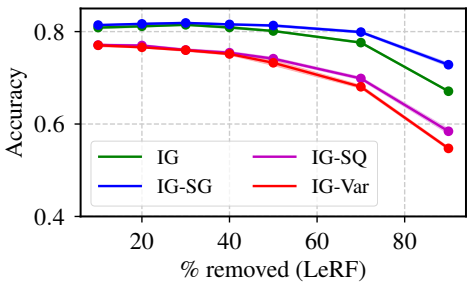


(d) LeRF, No-Retrain

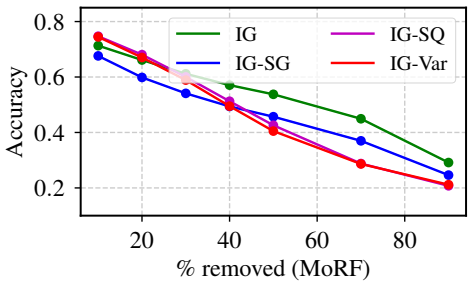
Figure 13. Consistency comparison using **Fixed Value** imputation on **IG**-based methods on CIFAR-10



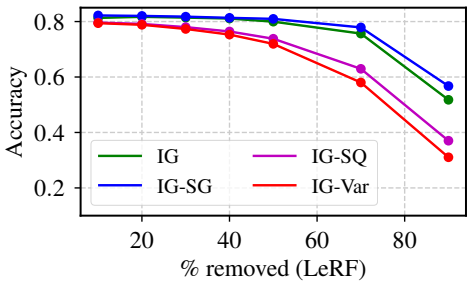
(a) MoRF, Retrain



(b) LeRF, Retrain

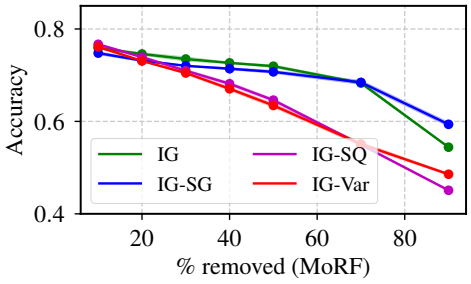


(c) MoRF, No-Retrain

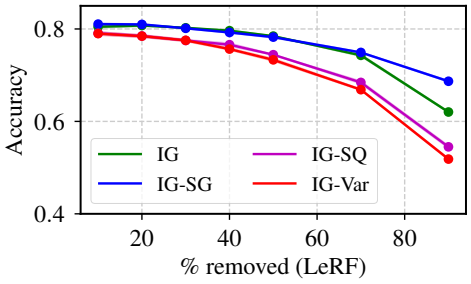


(d) LeRF, No-Retrain

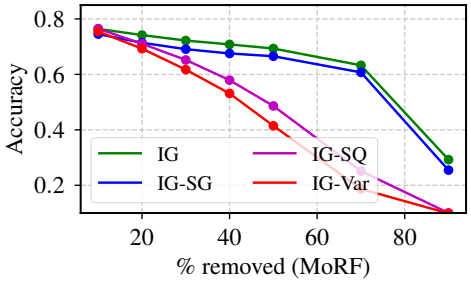
Figure 14. Consistency comparison using **Noisy Linear** imputation on **IG**-based methods on CIFAR-10



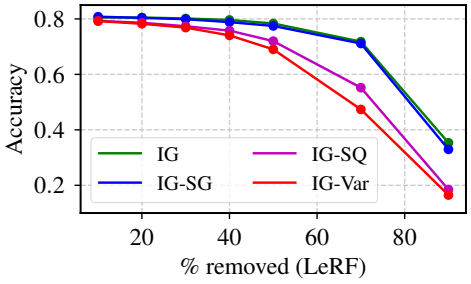
(a) MoRF, Retrain



(b) LeRF, Retrain

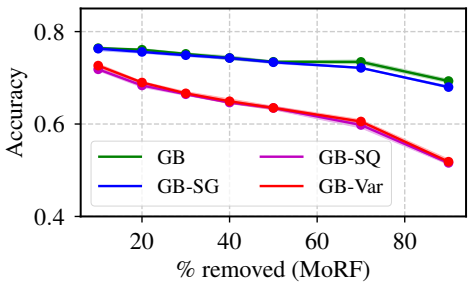


(c) MoRF, No-Retrain

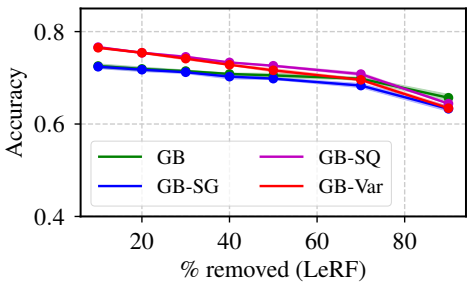


(d) LeRF, No-Retrain

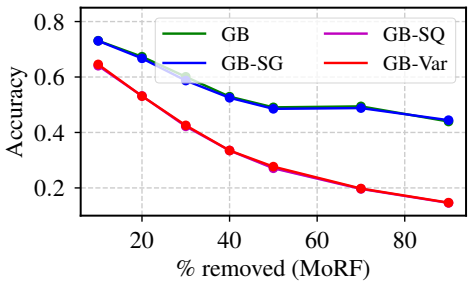
Figure 15. Consistency comparison using **GAN** imputation on **IG**-based methods on CIFAR-10



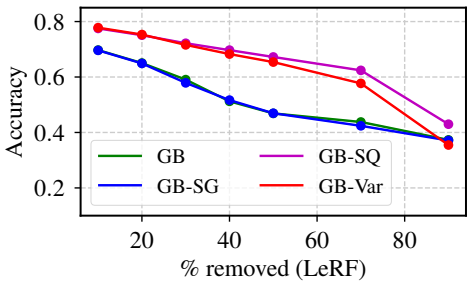
(a) MoRF, Retrain



(b) LeRF, Retrain

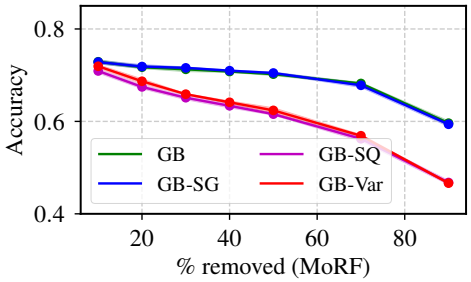


(c) MoRF, No-Retrain

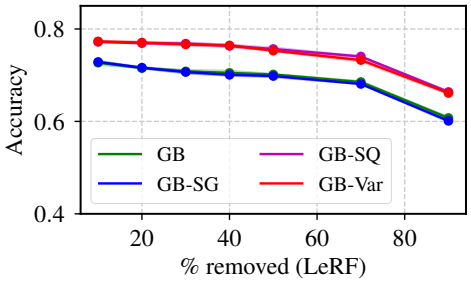


(d) LeRF, No-Retrain

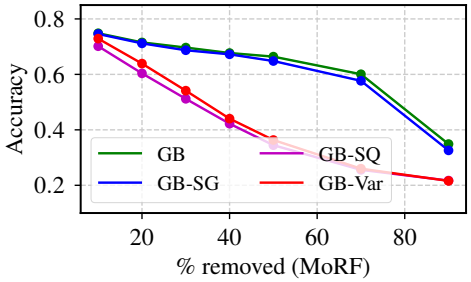
Figure 16. Consistency comparison using **Fixed Value** imputation on **GB**-based methods on CIFAR-10



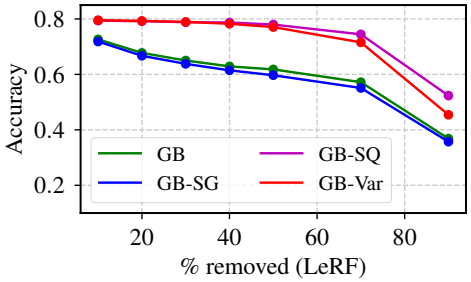
(a) MoRF, Retrain



(b) LeRF, Retrain

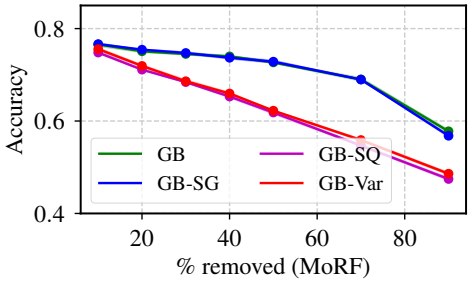


(c) MoRF, No-Retrain

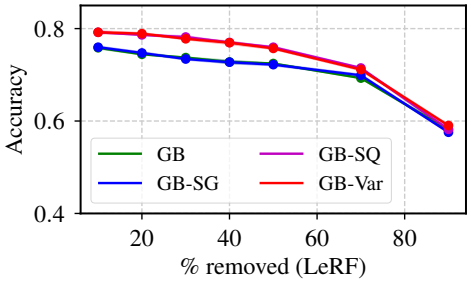


(d) LeRF, No-Retrain

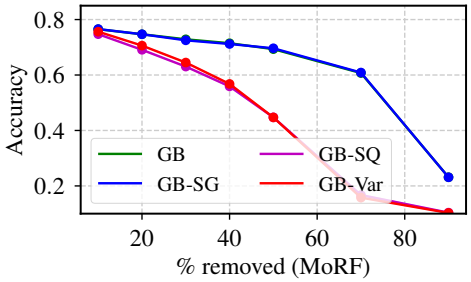
Figure 17. Consistency comparison using **Noisy Linear** imputation on **GB**-based methods on CIFAR-10



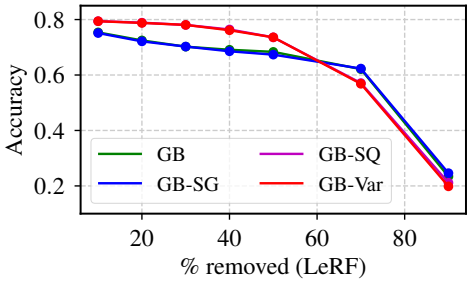
(a) MoRF, Retrain



(b) LeRF, Retrain



(c) MoRF, No-Retrain

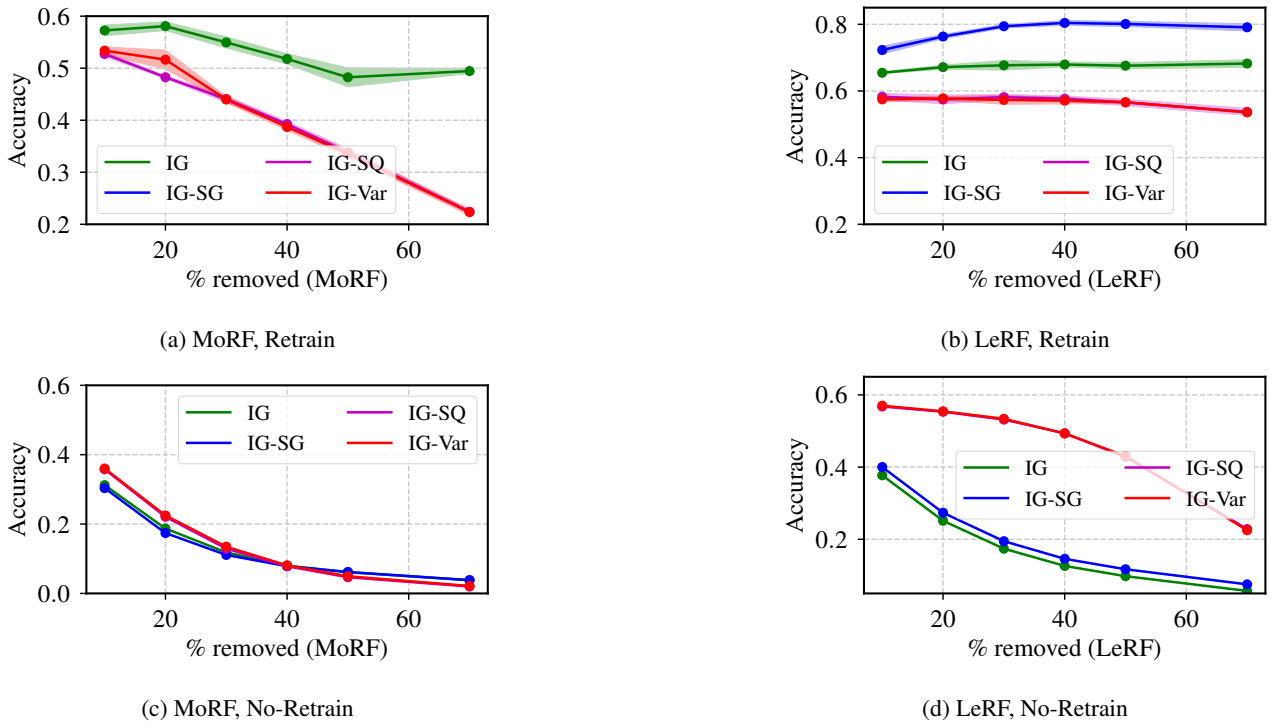


(d) LeRF, No-Retrain

Figure 18. Consistency comparison using **GAN** imputation on **GB**-based methods on CIFAR-10

		Retrain MoRF			No-Retrain MoRF			Retrain LeRF			No-Retrain LeRF		
		fixed†	lin	gan	fixed	lin*	gan	fixed	lin	gan	fixed	lin	gan
Retrain MoRF	fixed†	1.00 ±0.00											
	lin	0.48 ±0.03	1.00 ±0.00										
	gan	0.50 ±0.04	0.79 ±0.03	1.00 ±0.00									
No-Retrain MoRF	fixed	**0.12** ±0.01	0.57 ±0.02	0.50 ±0.01	1.00 ±0.00								
	lin*	0.61 ±0.01	**0.81** ±0.02	0.67 ±0.04	0.31 ±0.01	1.00 ±0.00							
	gan	0.74 ±0.01	0.79 ±0.02	**0.67** ±0.04	0.35 ±0.01	0.86 ±0.00	1.00 ±0.00						
Retrain LeRF	fixed	**-0.26** ±0.02	0.41 ±0.02	0.30 ±0.02	0.53 ±0.01	0.10 ±0.01	0.11 ±0.01	1.00 ±0.00					
	lin	-0.40 ±0.02	**0.26** ±0.04	0.19 ±0.04	0.30 ±0.03	-0.05 ±0.01	0.09 ±0.01	0.83 ±0.01	1.00 ±0.00				
	gan	-0.18 ±0.01	0.46 ±0.04	**0.32** ±0.04	0.50 ±0.03	0.13 ±0.02	0.14 ±0.03	0.89 ±0.02	0.83 ±0.01	1.00 ±0.00			
No-Retrain LeRF	fixed	0.79 ±0.02	0.79 ±0.03	0.63 ±0.05	**0.32** ±0.01	0.85 ±0.00	0.89 ±0.00	**0.02** ±0.01	-0.15 ±0.02	0.10 ±0.03	1.00 ±0.00		
	lin	-0.28 ±0.02	0.35 ±0.02	0.28 ±0.04	0.46 ±0.00	**-0.03** ±0.00	-0.06 ±0.00	0.89 ±0.01	**0.81** ±0.02	0.87 ±0.01	-0.11 ±0.00	1.00 ±0.00	
	gan	-0.45 ±0.02	-0.08 ±0.03	-0.04 ±0.04	0.23 ±0.00	-0.37 ±0.00	**-0.44** ±0.00	0.58 ±0.01	0.61 ±0.01	**0.54** ±0.00	-0.41 ±0.00	0.70 ±0.00	1.00 ±0.00

Table 10. **Food-10**: Rank Correlations between all evaluation strategies used with standard deviations computed by considering the rankings obtained through five consecutive runs as independent. The ROAR benchmark is marked by † and our ROAD by *. Bold results highlight the consistency between Retrain and No-Retrain (still very high) as well as MoRF and LeRF evaluation strategies using different imputation operators (fair increase when using Noisy Linear and GAN imputations instead of fixed imputation in "Retrain", decrease in "No-Retrain").



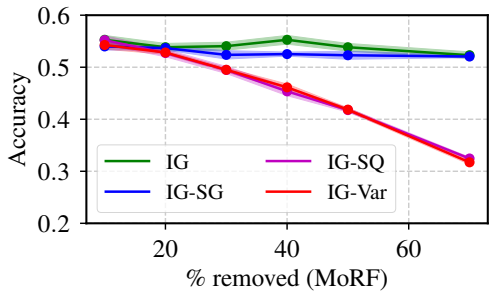
(a) MoRF, Retrain

(b) LeRF, Retrain

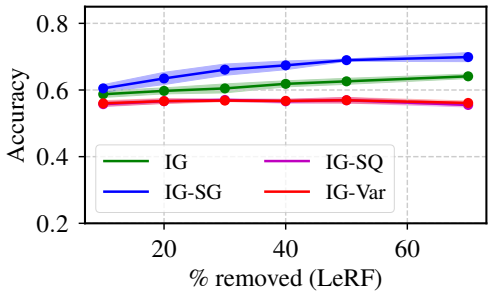
(c) MoRF, No-Retrain

(d) LeRF, No-Retrain

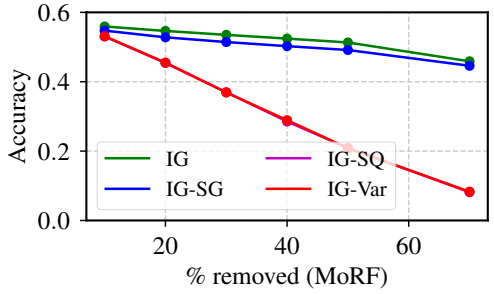
Figure 19. Consistency comparison using **Fixed Value** imputation on **IG**-based methods on Food-101.



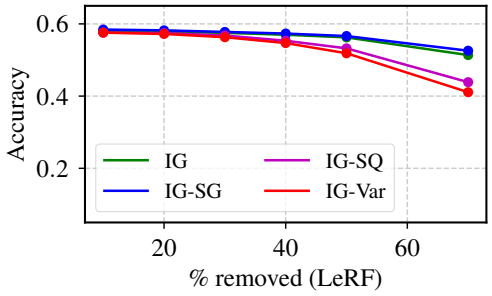
(a) MoRF, Retrain



(b) LeRF, Retrain

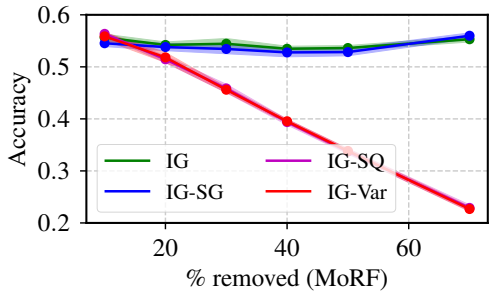


(c) MoRF, No-Retrain

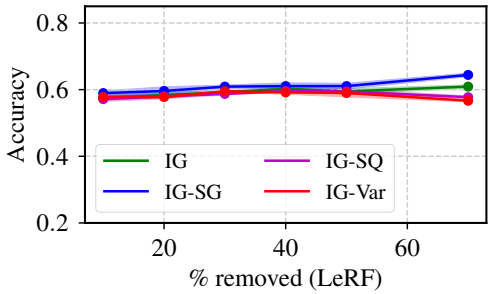


(d) LeRF, No-Retrain

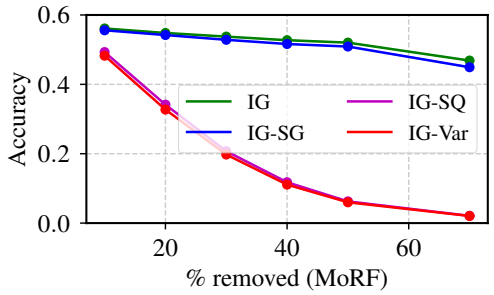
Figure 20. Consistency comparison using **Noisy Linear** imputation on **IG**-based methods on Food-101.



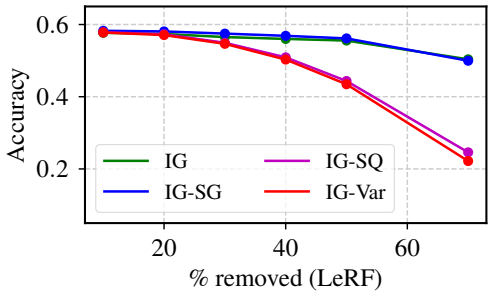
(a) MoRF, Retrain



(b) LeRF, Retrain

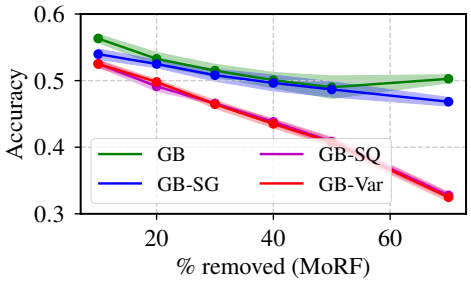


(c) MoRF, No-Retrain

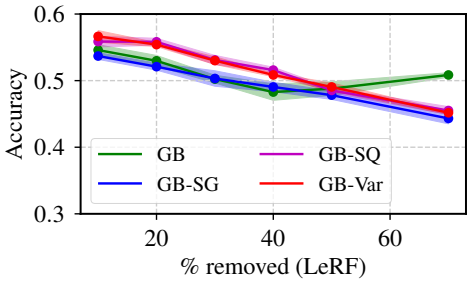


(d) LeRF, No-Retrain

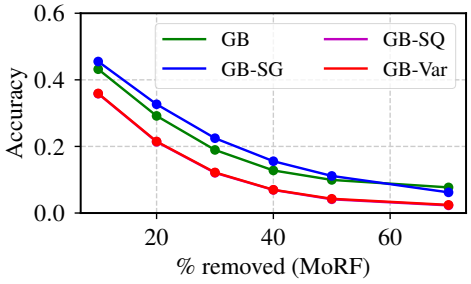
Figure 21. Consistency comparison using **GAN** imputation on **IG**-based methods on Food-101.



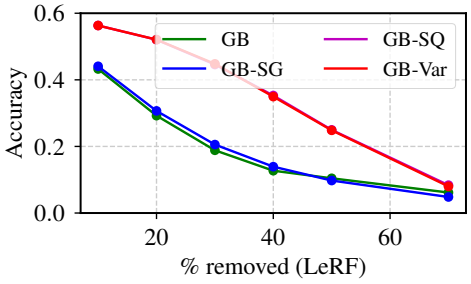
(a) MoRF, Retrain



(b) LeRF, Retrain

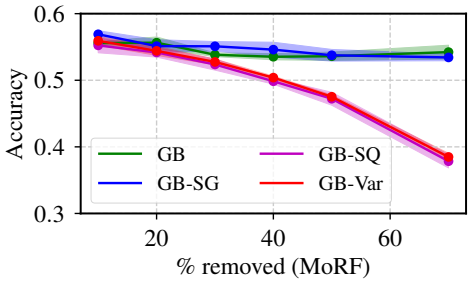


(c) MoRF, No-Retrain

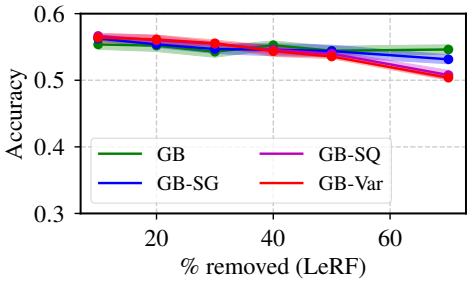


(d) LeRF, No-Retrain

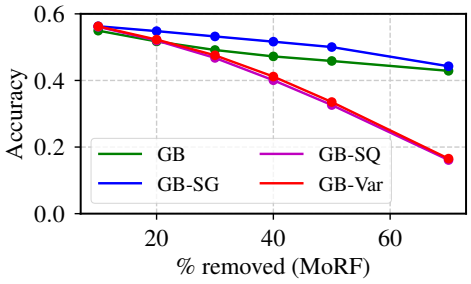
Figure 22. Consistency comparison using **Fixed Value** imputation on **GB**-based methods on Food-101.



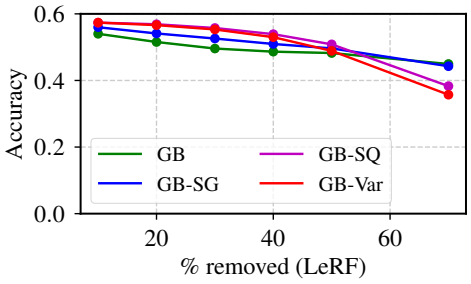
(a) MoRF, Retrain



(b) LeRF, Retrain

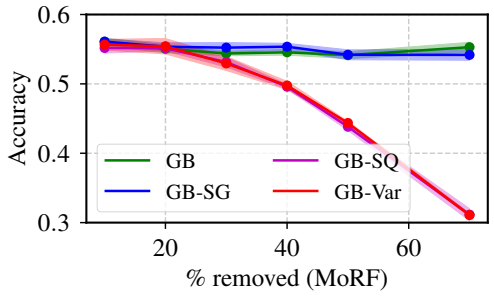


(c) MoRF, No-Retrain

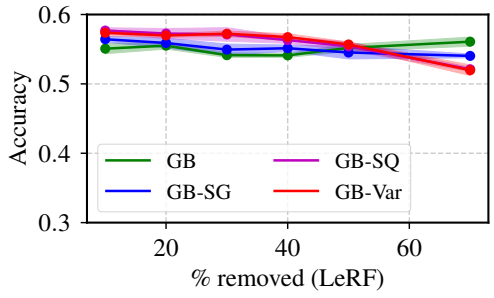


(d) LeRF, No-Retrain

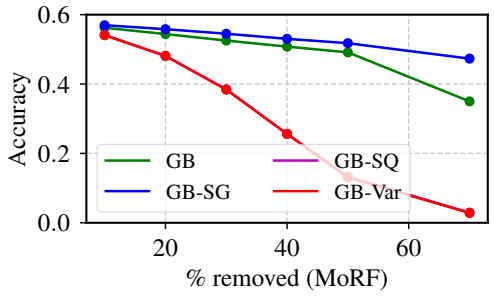
Figure 23. Consistency comparison using **Noisy Linear** imputation on **GB**-based methods on Food-101.



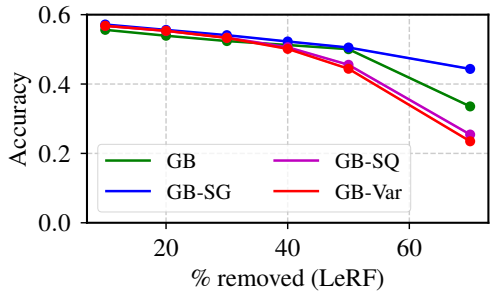
(a) MoRF, Retrain



(b) LeRF, Retrain



(c) MoRF, No-Retrain



(d) LeRF, No-Retrain

Figure 24. Consistency comparison using **GAN** imputation on **GB**-based methods on Food-101.