

# Beyond Explaining: Opportunities and Challenges of XAI-Based Model Improvement

Leander Weber<sup>1</sup>, Sebastian Lapuschkin<sup>\*1</sup>, Alexander Binder<sup>2, 4</sup>, and Wojciech Samek<sup>\*1, 3</sup>

<sup>1</sup>Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, 10587 Berlin, Germany

<sup>2</sup>ICT Cluster, Singapore Institute of Technology, 138683 Singapore, Singapore

<sup>3</sup>BIFOLD – Berlin Institute for the Foundations of Learning and Data, Berlin, Germany

<sup>4</sup>Department of Informatics, University of Oslo, 0373 Oslo, Norway

March 16, 2022

## Abstract

Explainable Artificial Intelligence (XAI) is an emerging research field bringing transparency to highly complex and opaque machine learning (ML) models. Despite the development of a multitude of methods to explain the decisions of black-box classifiers in recent years, these tools are seldomly used beyond visualization purposes. Only recently, researchers have started to employ explanations in practice to actually improve models. This paper offers a comprehensive overview over techniques that apply XAI practically for improving various properties of ML models, and systematically categorizes these approaches, comparing their respective strengths and weaknesses. We provide a theoretical perspective on these methods, and show empirically through experiments on toy and realistic settings how explanations can help improve properties such as model generalization ability or reasoning, among others. We further discuss potential caveats and drawbacks of these methods. We conclude that while model improvement based on XAI can have significant beneficial effects even on complex and not easily quantifiable model properties, these methods need to be applied carefully, since their success can vary depending on a multitude of factors, such as the model and dataset used, or the employed explanation method.

## 1 Introduction

In recent years, great advances have been made in the field of artificial intelligence, with especially deep neural networks (DNNs) achieving impressive performances in a multitude of domains, from image classification ([1, 2]) and the diagnosis of medical conditions ([3, 4]), to understanding chemical systems ([5, 6]), playing video games on a competitive level ([7, 8]), predicting the weather ([9, 10]) or the spread of infectious diseases ([11, 12]). And yet, the widespread application of these techniques in real-world scenarios has been hindered immensely by the black-box nature of DNNs. The reasoning behind their decisions is generally not obvious, and as such, they are simply not trustworthy enough, as their decisions may be (and often are) biased, as shown in, e.g., ([13–15]).

In order to alleviate this problem, the field of eXplainable Artificial Intelligence (XAI) has recently shifted into focus, which aims to open the black box of deep learning models. This increases transparency and trust, by better understanding the reasoning of these models. In fact, a multitude of explanation methods has been developed, that are able to visualize the basis for a model’s decision, all approaching the subject from different angles. Roughly, *global* (e.g., [16–18]) and *local* (e.g., [19–21]) XAI methods can be distinguished, with the former focusing on providing a general understanding of a model’s learned concepts and internal representations, and the latter aiming to explain the reasoning behind specific, singular decisions. However, while all of the above methods offer various insights into a model’s reasoning, the majority of research on the subject seems to stop here: Decisions are explained, and problems may be discovered, but the obtained insights are rarely applied to actually achieve *more* trustworthy, fairer, or simply better performing models.

In contrast, the very similar idea of incorporating human knowledge into a machine learning (ML) model’s training for the purpose of correcting its reasoning is relatively old. By using expert knowledge and feedback to regularize the model’s learning process, approaches such as [22] in the context of Support Vector Machines managed to improve learning speed and reasoning as early as 2007. Very recently, there have been growing efforts to include explanations in a similar manner, with the aim of improving various properties of current ML models. The techniques developed in this novel area of research are, however, conceptually extremely heterogeneous and their benefits and drawbacks are not well-studied. Also, to the best of our knowledge, so far there is no systematic review of XAI-based model improvement (although many reviews on explanation techniques have been recently published [23–25]).

In this paper, we aim to close this gap by deriving an **unifying theoretical framework** for XAI-based model improvement. We provide an exhaustive overview of the current state of this emerging research field by **reviewing existing approaches** in a structured manner and **evaluating them** systematically.

---

<sup>\*</sup>corresponding: {sebastian.lapuschkin,wojciech.samek}@hhi.fraunhofer.de

Here, we consider factors such as the improved model property (performance, reasoning, equality, etc.), the augmented component, the degree of human involvement, and the model agnosticity. **Showcasing various examples**, we further demonstrate the power of XAI for the purpose of model improvement in several applications experimentally, as well as discuss in detail how these effects are achieved. We conclude the paper by providing **practical recommendations** and discussing the opportunities and pitfalls in using explanations to improve deep learning models.

## 2 From Explaining Predictions to Improving Models

This section offers an overview over existing explanation methods, and discusses how they can help improve machine learning models. Through various toy examples, this effect is showcased in an exemplary fashion, and finally formalized to build a basis for categorizing the various approaches presented in Section 3.

### 2.1 Global vs. Local XAI

DNNs are highly complex models with an enormous number of parameters that interact in a nonlinear fashion. While their learned representations perform potentially better than hand-crafted ones, the underlying decision-making cannot always be easily interpreted. To cope with this problem the field of explainable AI has recently developed a multitude of methods, which elucidate on and visualize a model’s reasoning. Generally, two ends of the spectrum of XAI approaches can be distinguished:

*Global* XAI focuses on interpreting a model’s behavior and the features it has learned or is sensitive to in a general manner, e.g., by identifying and visualizing encoded concepts, as well as learned representations or sensitivities. Some methods aim to link information about specific, human-interpretable semantic concepts, e.g., labels, to single neurons [26] and subnets [18], or to vectors in feature space [17]. In contrast to finding representations (in terms of neurons or filters) for pre-defined concepts, methods such as SUMMIT [27] instead generate neuron importance maps from which encoded concepts can be inferred, or hierarchical relationships between concepts visualized.

*Local* XAI instead aims to explain the individual predictions of a model. That is, for specific samples, local explanation techniques seek to determine the parts of the input that have the most influence on a model’s decision for one particular sample. Methods that modify the backward pass, such as [19, 20, 28–30], explain efficiently in terms of computation time, but require grey-box access to a model’s internal parameters. Sampling-based techniques leverage local surrogate models [21], input perturbations [31, 32] or even game theory [33, 34] to interpret complete black-box models. In turn, however, they consume far more computational resources, only arrive at estimated explanations (due to sampling), and do not offer intermediate attributions as part of the explanation process. For more information about the explanation methods, we refer the reader to the excellent review papers [23–25].

In the literature, explanation methods are frequently used for visualization purposes and to *identify* problems with the model — but not to alleviate them. Nevertheless, a variety of approaches following above goal exists, that we aim to review and systematize in this work. Most of these techniques focus exclusively on employing local XAI, due to the detailed and sample-specific nature of these explanations.

### 2.2 Improving Different Model Properties with XAI

When ML models are deployed to real-world scenarios, they should not only achieve a good prediction accuracy, but also be trustworthy and reliable in their decision-making. However, traditional optimization metrics such as the test accuracy on a benchmark dataset do not guarantee that trustworthiness or reliability are fulfilled. As we will demonstrate on toy experiments (and also discuss theoretically in Section 2.3), explanations may help to improve on a variety of model properties, the most prominent of which we discuss in the following:

- **Performance:** ML models are usually trained in order to maximize their ability to predict correctly and with a high generalization ability. However, metrics solely focusing on the model’s performance, such as test accuracy on benchmark datasets, may be difficult to maximize, as many tasks and datasets are extremely complex. Existing models often have difficulties to thoroughly understand such datasets, and to identify the most descriptive — and at the same time domain-relevant — input features. Instead, detrimental effects such as overfitting on well correlating but comparatively domain-irrelevant input features in the training set may occur, and affect the test performance (and generalization beyond that) negatively. In this paper, we refer to test performance as *Performance*. Note, however, that this property may not always reflect the model’s actual generalization ability, e.g., if the test domain is too similar to the training domain (see *Reasoning* property).
- **Convergence:** Due to their enormous number of trainable parameters, modern ML architectures usually take significant effort and time to train. Faster convergence is thus desirable, but often difficult to balance with converging to an optimum that achieves state-of-the-art performance.

- **Robustness:** DNNs are often overly sensitive to changes in the input — even if those changes are imperceptible by humans. This effect can lead to adversarial cases where slight perturbations to input samples cause vastly different predictions [35, 36], or make models untrustworthy, as the explanations for their decisions can be manipulated arbitrarily by perturbing inputs without affecting the prediction [37, 38]. Both effects can be mitigated by increasing model robustness against slight alterations of the input.
- **Efficiency:** Current DNN-architectures usually require enormous amounts of data to learn from in order to achieve state-of-the-art performance. Depending on the data domain, it either needs to be labeled manually by experts, incurring extremely high effort and cost, or it can be annotated using automation or crowdsourcing, with the danger of including wrong labels. Neither high labeling cost nor a model learning the wrong information can be considered efficient. Furthermore, a large number of parameters — far more than are required in theory to solve the task at hand — is beneficial during training, as it leads to alternative subnets, helps represent complex functions and makes models easier to optimize [2, 39–41]. However, after a model is trained, a majority of these parameters does not effectively contribute much to solving the designated task accurately [42–44], but nevertheless requires a lot of storage resources and increases time and energy cost during inference. Therefore, making models more efficient by reducing the amount of data or features required during training, as well as the number or storage cost of parameters after training — while keeping performance intact — is extremely desirable, as the increased efficiency eases many future applications, such as employing DNNs on mobile devices.
- **Reasoning.** Due to their strongly data-dependent nature, modern ML models are prone to reflect any regularities present in the training data — whether these generalize to real-world circumstances or not. Therefore, even if (seemingly) performing well with a high test accuracy, the decisions of ML models are often based on spurious correlations, biases, and Clever Hans features occurring in the training dataset [13–15]. This is because all of these are features may appear helpful in solving a given task, without needing to understand the desired, potentially more complex (but often more valid) connections. Therefore, *Reasoning* is connected to the *Performance* property discussed above: If these confounding (or simply undesired) features are only present in the training set, such behavior can negatively affect test performance, and thus can be easy to identify. In this case, improving *Reasoning* would simultaneously increase *Performance*. On the other hand, if an undesired feature is present in both training and test sets, as is often the case, it may be helpful to the model and even increase test performance. The latter effect leads to test performance vastly overestimating a model’s actual generalization ability and obscures the undesired model behavior. Improving *Reasoning* in this case may impact *Performance* negatively [15], even though the resulting model is able to generalize better. As such, better *Reasoning* is extremely difficult to measure reliably, and often depends on (human-defined) ground truths of feature desirability and importance.
- **Equality:** Many real-world datasets contain inherent imbalances, e.g., between classes. When optimized to accurately predict such imbalanced datasets, machine learning models are prone to ignore the minority classes in favor of majority classes, since majority classes affect the average performance on the whole dataset the most and contribute significantly to the loss signal steering the training process. To achieve a good performance in the general case, equal treatment of all populations and sources of data has to be ensured. Note that this property is different from the notion of fairness, which is a widely researched topic on its own that we do not touch upon in this work.

Improvements w.r.t. the discussed model properties are crucial for the widespread and successful application of complex ML models. However, this is often far from trivial, due to the black-box nature of large models, and the resulting lack of information about the model and its decision-making. If exploited successfully, explanations can provide the additional information needed to improve above properties (at least partially). For instance, in the case of performance and convergence, knowledge about relevant and irrelevant feature representations can help focus on the most important ones and therefore reduce training time and increase accuracy. Similarly, identifying the most important neurons and filters is key to improving model efficiency. In order to gain an intuition of this phenomenon, we demonstrate empirically how XAI — under the right conditions — can help improve specific model properties through a series of simple toy experiments, with three different scenarios. Details about the experimental setup of the toy experiments can be found in A, and the datasets used in these experiments are visualized in C.

### Toy Experiment 1 (Model Performance)

The first experiment demonstrates that XAI can help improve prediction accuracy by XAI-based weighting of feature representations during the forward pass. More precisely, we use explanations to identify and increase intermediate features, which correspond to positive evidence for a specific prediction, and decrease the ones corresponding to negative evidence, similar to the approaches discussed in Section 3.2. In short, we trained a neural network to solve a binary classification task on a dataset with two informative and three

random input dimensions (Dataset visualized in Figure 1). We weighted intermediate features during the forward pass (similar to [45]) using a feature-wise (e.g., intermediate input features to layer  $l$ ) and sample-wise (e.g., obtained for sample  $x_i$  with the network parameters  $\theta^t$ ) attention mask  $M_{\text{feat}}^{i,l,t} \in [0.5, 1.5]$ . The weighted features are obtained as

$$f_{\theta^t}^l(x_i)' = M_{\text{feat}}^{i,l,t} \odot f_{\theta^t}^l(x_i), \quad (1)$$

with  $\odot$  denoting the element-wise product. Thus, local explanations are used to construct  $M_{\text{feat}}^{i,l,t}$ , which acts as an attention filter to enhance or inhibit the relevant or irrelevant feature representations, respectively. In the experiment we compare the attention-augmented models with an unaugmented baseline.

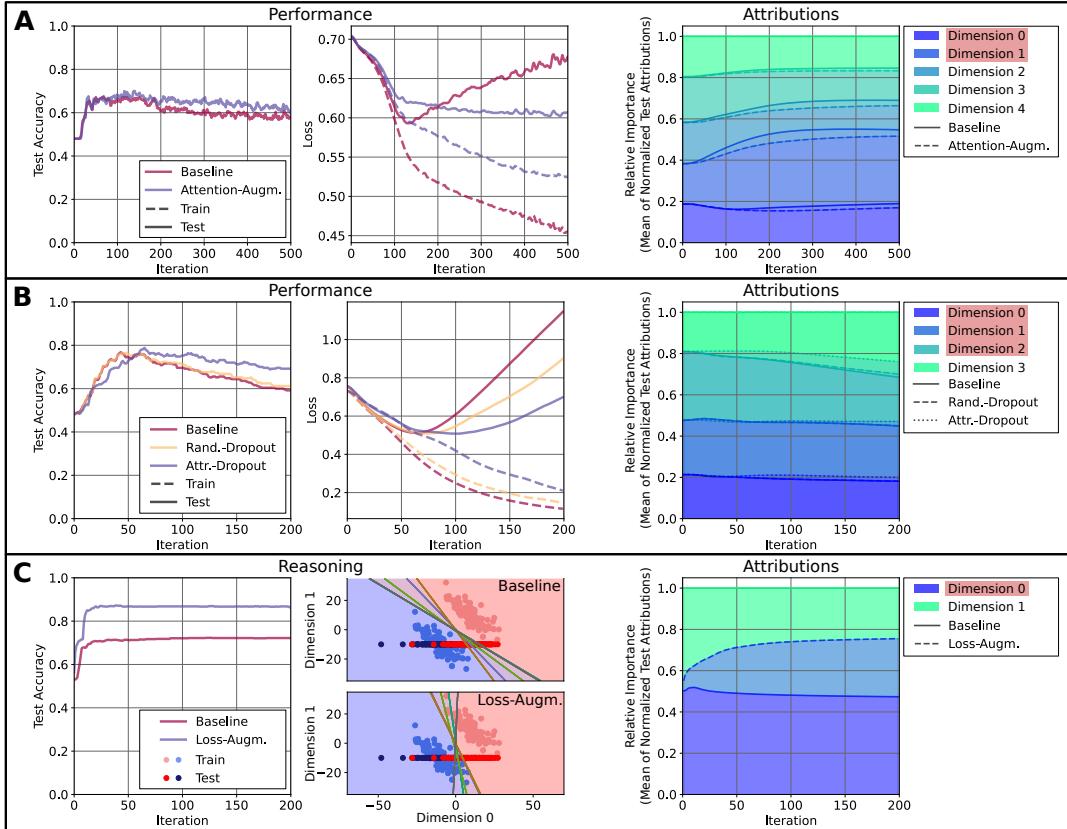


Figure 1: Toy experiments demonstrating the opportunities of XAI for model improvement. **A:** Improving performance by masking intermediate features during the forward pass (attention-augmentation) based on attributions, i.e., local explanations. The respective informative or desired input dimensions are highlighted in red. *Left:* Attention-augmentation improves test accuracy over baseline. *Center:* Attention-augmented models demonstrate a significantly reduced tendency to overfit. *Right:* The augmented model puts a more equal relative importance on all input dimensions. **B:** Reducing overfitting by introducing explanation-guided dropout. *Left:* Explanation-guided dropout improves test accuracy over baseline and random dropout models. *Center:* Strong overfitting tendencies for the baseline models, slightly less for the random dropout models, but far less for the explanation-guided dropout models. *Right:* Random dropout models attribute less importance to the distractor dimension 3 than the baseline. The explanation-guided dropout models exhibit this effect far more pronounced. **C:** Altering reasoning by using XAI to regularize the loss function. *Left:* The loss-augmented models significantly outperform the baseline models. *Center:* Visualization of the train (pastel colored points) and test (saturated colored points) datasets, along with the achieved decision boundaries by the five repetitions of the experiment. The loss-augmented models barely use the undesired dimension 1 for their predictions. *Right:* By regularizing the attributions in the loss function, the relative importance scores of dimension 1 are in fact reduced enormously.

The results (based on 5 repetitions) of this experiment are depicted in Figure 1A. The attention-augmented models are on average able to generalize considerably better than the baseline models that were trained without a feature re-weighting (see *top left*). The trajectory of the test accuracy is also far more stable for the augmented models, and decreases less over training iterations. This finding is strongly supported by the associated training and test losses (*center*). The baseline models achieve a lower training loss than the augmented models, but their test loss starts increasing from around iteration 120 — a clear sign of overfitting. In contrast, the test loss of the augmented models stays comparatively constant from iteration 120 onward, even decreasing slightly but steadily. The normalized test set attribution values as shown on the *right* explain the reduced overfitting of the augmented models. The relative

importance of the three uninformative input dimensions (2–4) decreases steadily for both baseline and attention-augmented models. Between the two informative dimensions (0 and 1), the baseline models seem to strongly favor dimension 1. While this is still the case for the attention-augmented models, they attribute less importance to dimension 1 in comparison, using both informative dimensions more equally. However, the relative importance of dimension 0 also decreases compared to the baseline, while the relative importance of the uninformative dimensions increases. It seems that by quickly identifying the informative dimensions, and overfitting on them, the baselines simultaneously lose generalization performance. In contrast, the augmented models learn more cautiously but with increased stability, and thus are able to generalize better (note that considering all informative dimensions is a property of large margin classifiers).

### Toy Experiment 2 (Model Performance)

The second toy experiment also aims at improving model performance, this time by explicitly reducing overfitting on singular distractive input dimensions. We train a neural network to solve a binary classification task and generate the data in a way that the dimensions 0–2 are truly informative about the class, while dimension 3 indicates the correct class via the sign for the training samples, but is randomized for the test samples (Dataset visualized in Figure 2). A generalizing model would not only rely on the distractor dimension 3 for its predictions, but instead leverage information from all informative input dimensions (large margin property). In order to ensure this property, we use an explanation-guided dropout method, similar to [46], which temporarily turns off the features that the model uses most to make its predictions. We compare this XAI-supported approach with the unaugmented baseline models and models that employ random (i.e., the standard) dropout.

Figure 1B visualizes the results of this series of experiments. Due to how we designed our test set, the test performance shown to the *left* is related to how much the three informative input dimensions, rather than the distractive fourth one are used. The random dropout improves slightly upon the baseline, indicating that the resulting models are less reliant on the fourth input dimension. While the explanation-guided dropout performs worse than both baseline and random dropout for the first few iterations — as is to be expected, since learning would slow down when the currently important intermediate features are dropped out — from around iteration 60, it outperforms even the random dropout increasingly. Concurrently with these observations, the training and test loss curves depicted in the *center* indicate strong overfitting for the baseline models, since test losses start rising again around iteration 75. The same effect occurs for the random dropout and XAI-dropout, but far later (iteration 80 and 125, respectively), and less rapidly. In order to gain an intuition about why above effects occur, we investigate the corresponding attributions in further detail. Specifically, the mean over samples and iterations of input dimension-wise test set attributions is shown to the *right* of Figure 1B. Compared to the baseline, the random dropout model attributes slightly less relevance to the distractor dimension relative to the informative dimensions. Again, this effect is even more pronounced for the explanation-guided dropout models.

Dropout is a useful tool in order to decrease overfitting, since it encourages the model to find alternative solutions to a given task [47]. Usually, the dropped out neurons are chosen at random, which only stochastically encourages alternative solutions. In contrast, XAI allows for a smarter choice. By dropping out the most important neurons, the most relevant path through the network is always disrupted, maximizing the resulting increase in generalization. Note, however, that this effect may potentially be detrimental to learning on more complex tasks, since the model may never get a chance to converge properly. Making the dropout criterion increasingly random over the course of training may therefore be beneficial.

### Toy Experiment 3 (Model Reasoning)

For the last toy experiment, we aim at aligning model reasoning to a ground truth provided by a human expert, similar to the approaches discussed in Section 3.3. We train models on two-dimensional data, and alter their decision-making to ignore one of the two dimensions. This selection seems arbitrary in a toy setting, however, even on complex datasets, the decision whether an input feature is desired or not often requires a human expert rather than being made automatically based on the available data alone. The data is visualized in Figure 1C (*center*), with the training set consisting of the pastel colored points and the test set in saturated colors, and in Figure 3. While the training set varies in the direction of dimension 1, the test set does not. After computing the explanations, we augment the loss function similarly to [48] as

$$\mathcal{L}_{\text{loss-aug}}^{l,t}(x_i) = \mathcal{L}_{\text{pred}}(f_{\theta^t}(x_i), y_i) + \mathcal{L}_{\text{reason}}(r_i, r_A), \quad (2)$$

$$\text{with } \mathcal{L}_{\text{reason}}(r_i, r_A) = \|(1 - r_A) \odot (r'_i)^{l,t}\|_2^2, \quad (3)$$

where  $\mathcal{L}_{\text{pred}}$  is the standard classification loss, and  $r_A$  a binary ground truth mask (same one for the whole dataset). Through the regularization term, the model is rewarded for aligning its explanations ( $(r'_i)^{l,t}$  before layer  $l$  at training iteration  $t$ ) with the ground truth explanations ( $r_A$ ). More precisely, it is penalized for using input dimensions marked as not desirable (i.e.,  $r_A = 0$ ).

The results of this experiment are visualized in Figure 1C. Due to the design of the test set, which only varies in the direction of dimension 0, the test accuracy over iterations depicted on the *left* directly indicates how well a model can classify without relying on the undesired dimension 1. Here, the loss-augmented models quickly and consistently outperform the baseline models, implying that the loss-augmented models rely less on input dimension 1. This interpretation is confirmed by the corresponding decision boundaries (*center*): The boundaries of the loss-augmented models are shifted to be nearly completely orthogonal to dimension 0, demonstrating that the models’ decision-making is changed to exhibit the desired behavior. While the baseline models attribute almost equal importances to both input dimensions (*right*), the loss regularization leads to a significant increase for the importance of dimension 0, while the opposite is the case for the undesired dimension 1.

In this experiment, XAI provides a measurement for model behavior, which can be compared against a ground truth, and in turn employed to alter model behavior as desired. But for more complex tasks, even if the explanations change through loss regularization, there is no guarantee that the decision behavior is altered in the desired manner, as indicated by [37, 38]. Due to the simple setting, however, this effect cannot occur in the above toy experiment. Ground truth explanation(s)  $r_A$  are required for adapting reasoning through loss regularization, but obtaining them requires involvement from human experts and may be infeasible for large datasets and complex tasks in practice.

As demonstrated through the three example scenarios, XAI can be employed in various ways to improve different model properties, since it is able to measure and identify the relevance of input dimensions, features, and neurons wrt. the model’s decision-making. Therefore, it is not only useful for informing human experts about a model’s behavior, but can be utilized practically to obtain better models.

### 2.3 Theoretical Formalization

After showcasing the ability of XAI to improve ML-models empirically through different toy examples, we will formalize those insights in order to derive a theoretical framework for categorizing existing approaches in a systematic manner.

Assume we have a model  $f_{\theta^t}$ , parametrized by parameters  $\theta^t$  after training iteration  $t \in \{1, \dots, T\}$ , and some data  $X = \{x_1, \dots, x_N\}^*$  with ground-truth labels  $Y = \{y_1, \dots, y_N\}$ . The model  $f_{\theta^t}$  consists of  $L$  layers  $l \in \{0, \dots, L-1\}$ , where the parameters of each singular layer after  $t$  training iterations are denoted by  $\theta^{l,t}$ . The *input features* to layer  $l$  are given by  $f_{\theta^t}^l(X)$ , so that, e.g., the input can be written as  $X = f_{\theta^t}^0(X)$ , and the model’s output as  $f_{\theta^t}(X) = f_{\theta^t}^L(X)$ . We further assume that a (local) XAI technique is available, that is able to provide explanations  $R^{l,t} = \{r_1^{l,t}, \dots, r_N^{l,t}\}$  for the model’s decisions at each intermediate layer  $l$  and iteration  $t$ , corresponding to intermediate features  $f_{\theta^t}^l(X)$ . These explanations can be leveraged to augment each component — i.e., Data → Feature Representations → Loss Function → Gradient → Trained Model — separately. Some examples of this approach were discussed in the toy experiments in Section 2.2. The different types of augmentation at each component of the training loop are depicted in Figure 2.

**Augmenting the Data.** XAI-dependent data augmentation leverages explanations in order to alter the structure of the data. We describe this alteration via the general function  $\Theta(X, R^{l,t})$ , which takes the original data  $X$ , and corresponding attributions  $R^{l,t}$  as inputs, in order to generate augmented data as

$$(X')^{l,t} = \Theta(X, R^{l,t}). \quad (4)$$

In practice,  $\Theta$  either generates new samples depending on  $R^{l,t}$  [49, 50], or alters the distribution of existing data [51]. As shown in Figure 3 (*top left*), this type of augmentation is applied during the first component of the forward-backward training loop, and therefore affects all other components.

**Augmenting the Intermediate Features.** As explanations can provide a measurement of feature importance, this information can be leveraged to scale, mask, or transform intermediate features, as shown by the toy experiments in Section 2.2 and the approaches discussed in Section 3.2. We generalize these augmentations using the function  $\Xi(f_{\theta^t}^l(X), R^{l,t})$ , which takes the input features  $f_{\theta^t}^l(X)$  to layer  $l$  and the corresponding attributions  $R^{l,t}$  as inputs, in order to generate augmented feature representations as

$$f_{\theta^t}^l(X)' = \Xi(f_{\theta^t}^l(X), R^{l,t}). \quad (5)$$

As shown in Figure 3 (*center left*), augmenting intermediate features does not affect the model’s inputs, but all subsequent training process components. Note that the input space can technically be viewed as an extension of the feature space. However, as will be shown in further detail in Section 3.1, the relevant data augmentation approaches focus on altering the distribution of the data *as a whole* — as opposed to most (intermediate) feature augmentation approaches, that focus on masking or transforming feature

---

\*Note that we switch to dataset-wise notation in this Section, since it allows for more concise and readable expressions, as opposed to the sample-wise notation in other Section that is required due to details of some augmentation methods being sample-specific.

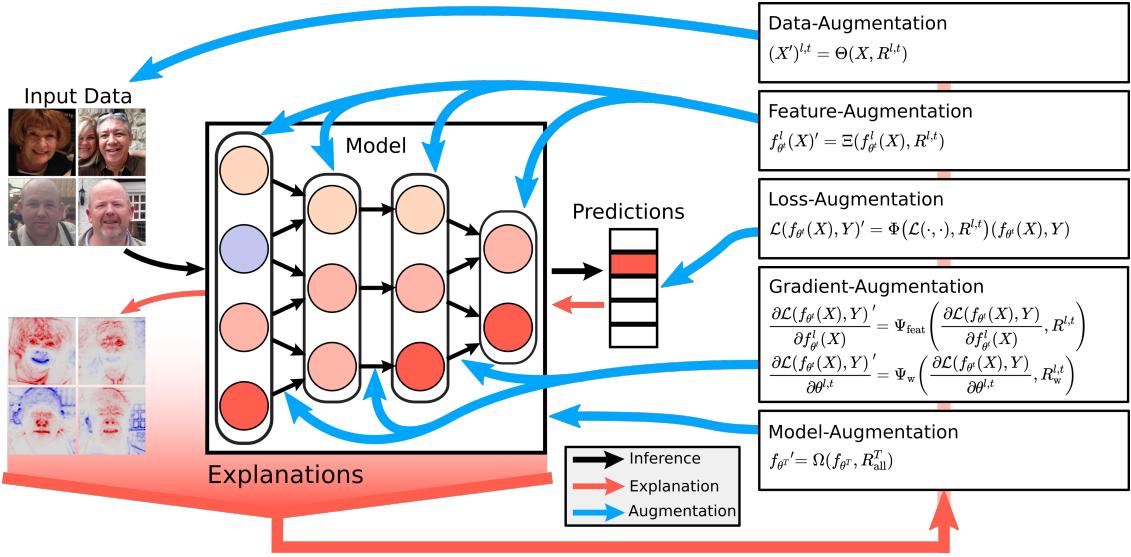


Figure 2: Model improvement with XAI. Explanations offer information about the model decision-making and behavior, which may in turn be leveraged to improve various model properties by augmenting different components of the training process or by adapting the trained model.

representations on a *per-sample* basis.

**Augmenting the Loss Function.** The loss function determines the behavior of a model. Thus, augmenting the loss function based on explanations can help specify which behavior is desired, using explanations as a feedback. Based on the standard loss function  $\mathcal{L}(f_{\theta^t}(X), Y)$ , measuring a model’s predictive error, and explanations  $R^l$ , the general augmented loss function is then given as the output of augmentation  $\Phi$ , i.e.

$$\mathcal{L}(f_{\theta^t}(X), Y)' = \Phi \left( \mathcal{L}(\cdot, \cdot), R^{l,t} \right) (f_{\theta^t}(X), Y). \quad (6)$$

For instance,  $\Phi$  can either add an XAI-based regularization term or scale class-wise losses, as discussed in Section 3.3. Augmenting the loss function in this manner only affects the backward pass (see Figure 3 (*top right*)).

**Augmenting the Gradients.** Similarly to how explanations can be leveraged in order to augment feature representations during the forward pass, the information about feature importance that they offer is applicable to the backward pass as well. Here, however, two sub-types of augmentation can be distinguished, due to how parameter updates are derived using the chain rule.

Firstly, as depicted at the *top* of the *bottom right* panel of Figure 3, the intermediate feature gradients at layer  $l$  can be scaled, masked, or transformed, mirroring the previously discussed feature augmentations during the forward pass. Based on the corresponding attributions  $R^{l,t}$ , we formulate this augmentation generally as

$$\frac{\partial \mathcal{L}(f_{\theta^t}(X), Y)}{\partial f_{\theta^t}^l(X)}' = \Psi_{\text{feat}} \left( \frac{\partial \mathcal{L}(f_{\theta^t}(X), Y)}{\partial f_{\theta^t}^l(X)}, R^{l,t} \right). \quad (7)$$

When applied to an intermediate layer  $l$ , this technique additionally alters feature gradients and parameter updates of all layers lower than  $l$ .

Alternatively, the parameter gradients can be augmented directly by computing parameter-wise importance scores  $R_w^l$ . These can either be obtained from the intermediate feature explanations  $R^{l,t}$  and  $R^{l+1,t}$  [52], or, in the case of many modified backpropagation XAI approaches, such as Layer-wise Relevance Propagation (LRP) [20], by simply computing explanations w.r.t. the parameters [53]. The corresponding augmentation can then be generalized as

$$\frac{\partial \mathcal{L}(f_{\theta^t}(X), Y)}{\partial \theta^{l,t}}' = \Psi_w \left( \frac{\partial \mathcal{L}(f_{\theta^t}(X), Y)}{\partial \theta^{l,t}}, R_w^{l,t} \right). \quad (8)$$

In contrast to Equation (7), this only leads to the parameter updates at layer  $l$  being altered, as visualized at the *bottom* of the *bottom right* panel of Figure 3. Approaches that augment the gradients during the backward pass are further discussed in Section 3.4.

**Augmenting the Model.** Even after a model is trained, the information w.r.t. intermediate feature importance offered by XAI may still be leveraged in order to augment the whole model, e.g., in order to alter its structure or reduce the amount of storage space required by the parameters. More concisely,

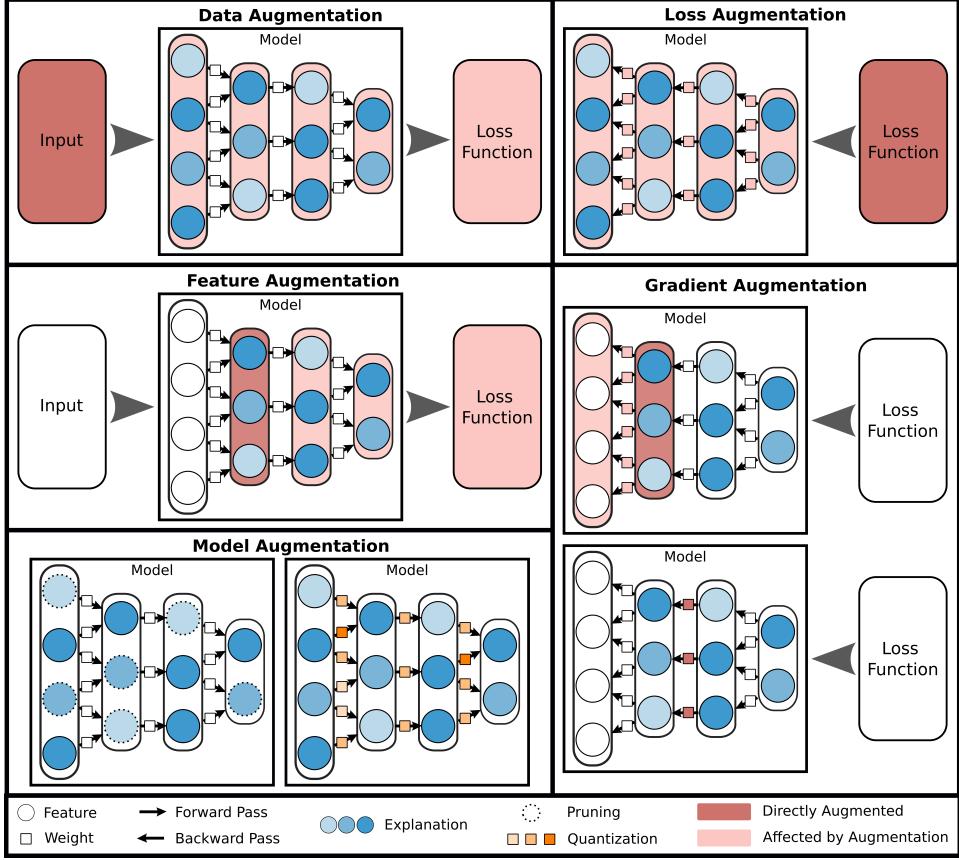


Figure 3: Types of XAI-based augmentation. *Top left*: Data augmentation alters the input distribution by utilizing input layer explanations. It therefore affects all components of the training loop as well as the final model. *Center left*: Feature augmentation leverages intermediate explanations at a specific layer to mask or transform the corresponding intermediate features, indirectly affecting all higher feature representations and subsequent components of training, such as gradient updates, and the resulting model. *Top right*: Loss augmentation uses the performance measuring capabilities of the loss function and affects all components of the backward pass and the final model indirectly. *Bottom right*: Two sub-types of gradient augmentation exist: Methods that augment the feature gradient at a specific layer (*top*) by masking or transforming it affect all parameter updates and feature gradients of lower layers in addition to the final model. In contrast, approaches that augment the parameter gradient at a specific layer (*bottom*) instead only affect the parameter updates of this layer. However, parameter-wise explanations are required for this purpose, which not all XAI techniques are able to easily provide. *Bottom left*: Augmentation of the model after training. Explanations at intermediate layers are leveraged to estimate neuron, filter, or parameter importance, and employ it as a criterion for, e.g., pruning (*left*) or quantization (*right*).

using explanations  $R_{\text{all}}^T = \{R^{0,T}, \dots, R^{L-1,T}\}$  before each layer of the model and after the last iteration  $T$ , the model  $f_{\theta T}$  can generally be augmented as

$$f_{\theta T}' = \Omega(f_{\theta T}, R_{\text{all}}^T). \quad (9)$$

In practice, XAI is either employed here to prune (Figure 3, *left* of the *bottom left* panel) or quantize the model (Figure 3, *right* of the *bottom left* panel), as discussed further in Section 3.5. Note that typically in literature, the above categories of XAI-based augmentation are only applied one at a time. However, due to each category altering a separate component of the training process, in theory, multiple augmentations, e.g., targeting the same model property, could be applied at the same time, altering different components of the training process simultaneously.

### 3 Review of Methods for XAI-Based Model Improvement

This section reviews different types of methods proposed for XAI-based model improvement. A main conceptual difference, which allows to distinguish these methods, is the component of the training loop at which augmentations are performed in order to achieve the improvement. More precisely, changes can be applied at the dataset-level, during the forward pass to the feature representations, within the loss function, to the gradients during optimization, or after training to the model structure and parameters, as

discussed in the previous section. Generally, each of these augmentations comes with different implications in terms of the improved model properties, computational costs, time requirements, and model agnosticity.

### 3.1 Augmenting the Data

The fitting and training processes in ML are generally highly data-dependent, especially for the widely employed DNNs, with the performance of resulting models being strongly affected by the available input features, imbalances, biases, and spurious correlations present within the dataset. Additionally, (local) XAI methods primarily (although not exclusively) focus on providing explanations in input space, as they target a human audience. When employing XAI to improve ML models, augmentations of the data as described in (4) thus come naturally to mind. In principle, methods not discussed in this Section, particularly some feature augmentation methods (Section 3.2) could be applied to the input space as well (as is done, for instance, in [15]). However, here we distinguish data augmentation approaches as those that focus on changing the distribution of the data as a whole — as opposed to altering singular samples.

By changing the distribution of samples that are fed to the model, the approaches in this category aim to mitigate biased, unfair, or simply wrong decision-making behavior that is caused by the vanilla data structure. For this purpose, explanations can be leveraged in order to generate artificial samples that provide exemplary information against undesired behavior [49, 50], or to simply re-sample the existing data to achieve fairer predictions. More precisely, eXplanatory Interactive Learning (XIL) [49, 50] enables human users to correct a model’s decision-making instance-wise during an active learning setting. Here, the learner is able to periodically query a user for the correction of its prediction. In order to improve model reasoning and human trust in a model’s decisions, XIL leverages local explanations: After selecting an informative instance, the subsequent query from model to user here includes the instance, corresponding prediction, as well as an explanation for this decision in form of a heatmap. In turn, the user may not only correct the prediction, but the explanation as well, and provide this feedback to the model. Data augmentation is then performed based on the corrected explanation by generating counter examples to discourage the model from depending on irrelevant input features. XIL thereby corrects the model’s reasoning, avoids Clever Hans (CH) [14, 54] moments, and increases user’s trust into the model’s decision-making. The work in [50] further extends XIL to alternatively augment the loss function using the Right for the Right Reasons (RRR) loss proposed by [48], which will be further discussed in Section 3.3. The authors of [55] employ this loss-augmenting variant of XIL to Neuro-Symbolic concept learners, allowing for concept-based, semantic corrections to be made to the model.

The authors of [56] also aim at improving model reasoning, specifically w.r.t. artifacts such as CH effects or Backdoors [57], by discovering and removing artifactual samples from the dataset. To obtain explanations, they employ the self-interpretable ProtoPNet [58] architecture, and extend this method to Prototypical Relevance Propagation (PRP) using concepts from LRP [20]. Multiple explanations for each input image w.r.t. its true class can thus be obtained. Various multi-view clustering approaches, such as [59, 60], are subsequently employed to detect artifact samples and separate them from clean samples effectively. By training a new model on the clean data, the effects of specific artifacts on the model’s decision-making can be mitigated.

In order to improve model performance on visual data, specifically on fine-grained classification tasks (i.e., tasks with classes that have similar features, such as distinguishing different dog breeds, and thus require attention to detail), the authors of [61, 62] propose Guided Zoom. This method aims at refining model predictions through a comparison with evidence used at training time that led to correct decisions. First, an evidence pool is generated from the training data, by selecting the top-l salient patches of samples correctly classified by the original model  $f$ , based on an evidence grounding method. Specifically, explanations, i.e., contrastive Excitation Backprop [63] are employed as a grounding method here. An evidence model  $f^e$  is then trained on the obtained evidence pool to solve the same task as  $f$ . During test time, for each test sample  $x$ , the original model’s decision is refined by selecting the top-l salient regions w.r.t. the top-k predicted classes according to  $f(x)$ , and taking the prediction of  $f^e$  on these regions into account through a weighted combination. Predictions are therefore refined by an additional comparison of evidence used by  $f$  for a preliminary prediction to known class-specific evidence encoded via  $f^e$ . The authors of [61, 62] show that Guided Zoom improves model performance, and, potentially, model reasoning, as the refined decisions are less based on biases — although confounders that are strong enough to be selected as positive evidence towards a specific class might still affect the final decisions.

In the context of imbalanced data, [51] introduce XAI-guided imbalance mitigation, aiming towards an equal classification performance between classes. For this purpose, scalar metrics are first derived from a few representative attribution maps (here LRP) in order to extract descriptive information, such as the attribution map entropy, and the pairwise distance between attribution maps for the same sample at different times during training. These are observed to correlate strongly to test accuracy and F1-Score, and thus demonstrate that attributions allow for an estimation of a model’s generalization performance and convergence, even if there is no labeled data available, and can distinguish well between accurately and inaccurately predicted classes. This insight is then exploited to achieve more balanced class-wise performances when training a model on imbalanced datasets, by augmenting the learning process based on representative attributions. In a second step, class-wise factors are computed from above metrics

repeatedly during training. These are able to react to the model’s decision-making in order to alter the distribution of the input data that is fed to the model, leading to faster convergence and more equalized class-wise performances. Similarly, class-wise factors are also used to augment the loss by scaling its class-wise contributions, which will further be discussed in Section 3.3.

In the original works, Local Interpretable Model-Agnostic Explanations (LIME) [21] is used as means to obtain XAI attributions for XIL, although the framework is formulated in a general manner and considers any technique that offers local explanations. Similarly, [51] employ LRP to obtain explanations. Other (local) explanation methods would be applicable as well, as demonstrated in an exemplary fashion in Section 4. The authors of [61, 62] also formulate their method generally enough to account for explanation techniques beyond the employed Excitation Backprop. In contrast, the technique proposed by [56] relies on PRP, a specific explanation method which makes a model self-explainable, instead of providing post-hoc attributions, but in turn requires the model to be altered with a specialized additional layer and subsequent adaptation of parameters. While XIL allows for extremely thorough and instance-wise corrections to be made to the model, the required human feedback, especially the explanation corrections, in turn make it not only extremely slow in practice, but also subject to human error. The technique of [51] increases the computational load during model training, due to the repeated calculation of attributions, yet the additional (real) time cost is negligible, since no human interaction is required. The technique of [56] uses aggregated explanations and clustering algorithms to automatically separate artifactual and clean samples, drastically limiting the amount of required human involvement. However, they propose to remove the affected samples from the dataset, thus eliminating potentially valuable information in addition to the undesired artifacts. Guided Zoom requires no human involvement, but requires the training of an additional evidence model. Furthermore, inference time drastically increases due to each prediction requiring multiple inference steps, one for the original image and each selected salient region.

Due to the data-dependent nature of ML models, the impact of data augmentation methods on the model — and therefore their improvement potential — is extremely promising. As they only require access to the input-space, approaches that augment the data are generally able to retain a relatively high degree of model agnosticity.

### 3.2 Augmenting the Intermediate Features

While a model’s parameterization directly depends on the given (training) data, the feature space is often able to encode this data in a more concise form, so that some problems, e.g., with biases or reasoning, may be easier and more effectively (compared to input space) mitigated there via augmentation, as described in Equation (5). Two different types of methods exist in this category.

**Attention and Intermediate Feature Masking.** This type of XAI-guided feature augmentation techniques seek to boost a model’s performance by using explanations to distinguish relevant intermediate features from irrelevant ones. For this purpose, XAI methods that can provide intermediate explanations are required. As the shape of these intermediate explanations matches the shape of the features they explain, they can be directly employed to obtain a mask that expresses feature importance and weights them during the forward pass, similar to an attention mechanism. As a consequence, the features of the augmented layer, as well as the features of all higher layers are affected. In the context of image recognition, Attention Branch Network (ABN) [64] interprets the local explanations obtained by extending Class Activation Mapping (CAM) [65] as an attention map. ABN is comprised of three modules, a feature extractor, an attention branch, and a perception branch. While the perception branch is a standard classifier on the feature extractor’s output, the attention branch uses the output of the feature extractor to compute an attention map. In its original formulation, CAM uses a triple of convolutional layer, Global Average Pooling (GAP), and a fully-connected classification layer to return pixel-wise importance scores. These are obtained as the average activations over the channels of the convolutional layer, weighted by the impact of each channel on the classification score of a target class. Since this requires a trained model, the attention branch of ABN omits the fully-connected classification layer, and instead uses a convolutional layer with  $K$  channels that each correspond to a category. It can thus learn the attention map by interpreting the GAP result for each convolutional channel (i.e., one channel per class) as a logit and applying softmax to obtain class-wise probability scores. Utilizing this attention map, the perception branch input (i.e., the input of layer  $l$ ) can then be masked in order for the model to focus on the most important parts of a given sample. Since both branches output separate probability scores, the resulting ABN is trained using a loss function  $\mathcal{L}_{\text{abn}}$  that simply sums up the branch-wise losses:

$$\mathcal{L}_{\text{abn}}(f_{\theta^t}(x_i), y_i) = \mathcal{L}_{\text{att}}(f_{\theta^t}(x_i), y_i) + \mathcal{L}_{\text{per}}(f_{\theta^t}(x_i), y_i), \quad (10)$$

where  $\mathcal{L}_{\text{att}}$  and  $\mathcal{L}_{\text{per}}$  are the attention and perception branch losses, respectively. With this method, ABN is able to simultaneously explain decisions and exploit this knowledge for improved performance.

The authors of [66] extend ABN by additionally correcting attention maps by exploiting human knowledge, and thereby improve upon the visual explanation and classification performance of the original approach, while also altering model reasoning. For this purpose, an ABN is first trained, and the attention maps corresponding to each sample are stored. A human expert then edits and corrects these attention

maps, and the model is finetuned using the edited attention maps. In the last step, *loss augmentation* (see Section 3.3) is performed additionally, regularizing Equation (10) to produce attention maps that are similar to the corrected ones:

$$\mathcal{L}_{\text{abn}}(f_{\theta^t}(x_i), y_i) = \mathcal{L}_{\text{att}}(f_{\theta^t}(x_i), y_i) + \mathcal{L}_{\text{per}}(f_{\theta^t}(x_i), y_i) + \gamma \mathcal{L}_{\text{reason}}(r_i, a_i), \quad (11)$$

$$\text{with } \mathcal{L}_{\text{reason}}(r_i, a_i) = \|a_i^{l,t} - r_i^{l,t}\|_2, \quad (12)$$

where  $r_i^{l,t}$  is the ABN attention map,  $a_i^{l,t}$  the corrected attention map,  $\|\cdot\|_2$  the  $\ell_2$ -norm, and  $\gamma$  a scale factor.

Similar to ABN, [67] also aim at improving the image classification performance — specifically whale sound spectrogram classification — of Convolutional Neural Network (CNN)-type architectures by masking the feature extractor output (and classifier input) based on local XAI. More precisely, [67] first feeds the whole dataset to a CNN pre-trained on orca sound spectrograms. The intermediate activations after the feature extractor are then explained using the Deep Taylor Decomposition (DTD) method [28]. From these attributions, binary masks are then computed with two different goals, setting either the least relevant or the most relevant feature representations to zero. New classifiers are trained on the masked features, or on varying concatenations of masked features across different models and masking methods, resulting in a significantly increased classification performance compared to the base models, especially for the concatenation of both masked feature representations. Furthermore, by reducing the amount of used features — the authors employ a *binary* mask which turns off a subset of features, in contrast to other approaches — efficiency is increased while preserving performance.

The authors of [45] apply a similar method in a few-shot setting. In few-shot classification, the goal is for a (pre-trained) model to generalize to new classes, using only a small number of examples. However, this generalization can be difficult, if the source and target domains are very different. To mitigate that problem, the explanation-guided training proposed by [45] leverages XAI, more specifically LRP: Assuming a model that can be split into feature processing and classifier, during a forward pass, the feature processor outputs feature representations  $f_{\theta^t}^l(x_i)$  for each sample  $x_i$ , which are then used by the classifier to arrive at prediction  $f_{\theta^t}(x_i)$ . Using LRP,  $f_{\theta^t}(x_i)$  is explained in terms of the feature processing output  $f_{\theta^t}^l(x_i)$ , as  $r_i^{l,t}$ . Similarly to the approaches of [64] and [67],  $r_i^{l,t}$  can be employed — after normalizing it to the interval  $[-1, 1]$  to obtain  $(r')_i^{l,t}$  — as a mask, weighing the intermediate features  $f_{\theta^t}^l(x_i)$  as

$$f_{\theta^t}^l(x_i)' = (1 + (r')_i^{l,t}) \odot f_{\theta^t}^l(x_i), \quad (13)$$

with  $\odot$  denoting the element-wise product. When feeding  $f_{\theta^t}^l(x_i)'$  into the classifier, the LRP-weighted prediction  $f_{\theta^t}(x_i)'$  is obtained. The model can then be trained using an objective function that considers both predictions, i.e.,

$$\mathcal{L}(f_{\theta^t}(x_i), f_{\theta^t}(x_i)', y_i) = \alpha \mathcal{L}(f_{\theta^t}(x_i), y_i) + \beta \mathcal{L}(f_{\theta^t}(x_i)', y_i), \quad (14)$$

where  $\alpha, \beta$  are positive scalars. By applying above method, [45] are able to significantly improve model performance in cross-domain few-shot classification.

Similarly, [68] improve domain generalization performance by using XAI-feedback (specifically, Gradient-weighted Class Activation Mapping (Grad-CAM) [69] is employed during their experiments) to force the model to exhibit the correct reasoning and focus on the correct objects, rather than contextual information. For this purpose, they require binary ground-truth annotation masks for each sample that localize these objects, which is rescaled accordingly if explanations are computed at an intermediate layer. Periodically during training, each training sample is explained, and the explanation is compared to the ground-truth annotation. For each sample, if the peak, i.e., the maximum value of the corresponding explanation lies within the object area marked by the annotation, a binary mask  $M_i^{l,t}$  is defined, which is set to 1 where the explanation is larger than zero, and to 0 otherwise. Otherwise  $M_i^{l,t}$  is equal to the (potentially rescaled) ground truth annotation.  $M_i^{l,t}$  is then employed to select intermediate features as

$$f_{\theta^t}^l(x_i)' = M_i^{l,t} \odot f_{\theta^t}^l(x_i). \quad (15)$$

Intuitively, the approach of [68] can be understood as “dropping in” features that are seemingly correctly used by the model. In contrast, dropout aims at avoiding overfitting by periodically setting a subset of features to zero during training. For this purpose, random dropout [47] is usually employed, which simply decides randomly which features to mask at a given iteration. The authors of [46] improve upon this technique by instead employing XAI, more specifically Excitation Backpropagation [63], to drop out more important neurons with a higher probability in order to speed up the overfitting avoidance effect. During training, each sample is explained and a sample-specific dropout mask is computed accordingly, leading to not only an improved generalization ability, but also less degradation when neurons are removed, compared to random dropout.

All of the above methods — with the exception of [66] — are applicable in a relatively efficient manner and without human interaction during learning, so that no additional time investment is required on top of the standard model training time — although this time may be slightly increased due to the need for computing explanations for each sample during training, which is nevertheless comparatively

insignificant when using, e.g., modified backpropagation approaches to explain. In contrast, [68] include human knowledge through the necessary ground-truth object segmentation masks for each sample, which can be extremely tedious and infeasible to obtain. The approach of [66] instead requires manual editing of a multitude of attention maps by a human expert, which does not scale well to large datasets, but improves upon the original approach of [64] in terms of explanation, reasoning, and performance. The inclusion of human knowledge by [66] further mitigates the issue that the attention may be inconsistent with the ground truth, e.g., by including multiple objects, since ABN learns the attention map instead of computing it w.r.t. a target class from a trained model as in the original formulation of CAM. While the model structure of ABN as proposed by [64, 66] is extensible to multiple image recognition tasks, multi-task learning problems, and various CNN models, it is not general enough to include XAI methods other than CAM. Even though [45, 46, 67, 68] employ specific XAI methods in their experiments, any other (local) explanation technique is applicable in theory, as long as it is able to produce intermediate explanations.

**Intermediate Feature Transformation.** While above attention and feature masking methods directly utilized the feature-wise information provided by XAI to determine the importance of intermediate features and to scale them accordingly, the approaches listed here exploit explanations more indirectly and rely on more complex feature transformations such as translation and projection to, e.g., correct a model’s reasoning. With this goal in mind, the Class Artifact Compensation (ClArC) framework [15] is formulated in a general manner and specifically aims at the identification and removal of biases, artifacts, and CH behavior. For this purpose, the ClArC framework consists of three steps: Identifying artifacts, estimating an artifact model for each artifact, and updating the predictor model to reduce the artifact’s impact. More precisely, [15] proposes an extension of the Spectral Relevance Analysis (SpRAY) [14] algorithm for the artifact identification, which leverages large sets of local explanations to identify a model’s behavioral patterns. This condensed information can then be exploited by a human observer to pinpoint undesired biases and artifacts. Once an artifact is found and subsequently estimated, e.g., using Concept Activation Vectors (CAVs) [17], two distinct variants of ClArC exist for removal: Augmentative Class Artifact Compensation (A-ClArC), where the artifact is added to all samples regardless of class membership (with a certain probability) and the model is finetuned in order to be desensitized to the now class-unspecific artifact feature, and Projective Class Artifact Compensation (P-ClArC), where the artifact is suppressed during inference via projection, without additionally finetuning the model. Both approaches can be applied in feature space as well as to the input space, since inputs can be viewed as an extension of intermediate features. In this context, the authors demonstrate that the layer where a CH artifact can be mitigated most effectively strongly depends on the artifact’s complexity. ClArC thus manages to not only identify CH strategies, but successfully “un-Hanses” even large and complex datasets implicitly, such as ILSVCR2012 [70], ISIC 2019 [71–73], or the Adience benchmark dataset [74], without sacrificing samples which might otherwise contain valuable information next to artifactual features. Since human interaction is only required in the identification step, to interpret the model’s behavioral patterns aggregated via SpRAY, this method is comparatively fast and relatively independent of human errors. ClArC is further extended in [75], by estimating artifact directions as signals, with more robustness against noise.

As demonstrated above, XAI-dependent techniques that augment intermediate features can change a model’s internal feature representations in order to improve performance or achieve better reasoning. While most of these methods can boast no or only minimal requirement of human involvement — which strongly reduces the required time and effort for their application — they in turn cannot be model-agnostic, since they require access to a model’s internal feature representations (i.e., layer-wise explanations).

### 3.3 Augmenting the Loss

The loss function measures how wrong a model’s predictions are. Thus, it strongly influences the decision-making that a model will learn during training, as it controls the nature and shape of the error signal to be minimized via parameter updates. As such, augmenting the loss — as shown in Equation (6) — with additional regularization terms or applying a XAI-dependent scaling can naturally correct a model’s reasoning, robustness, and performance, or lead to faster convergence.

**Loss Regularization.** By adding a regularization term to the loss function, a model’s learning behavior can easily be nudged towards a variety of effects. For instance, by comparing explanations to some ground truth that encodes human expectations, reasoning can be enforced to align with expert knowledge [48, 76–79]. Alternatively, simply imposing some human-independent constraint on the explanations may provide various positive effects, such as improved reasoning, robustness, or performance [78–81].

Here, the authors of [48] introduce Right for the Right Reasons (RRR) which aims at optimizing a model’s reasoning. They assume a dataset  $X$  which offers — in addition to ground truth class labels — a binary annotation mask  $a_i^l$  for each sample  $x_i \in X$  that denotes for each input dimension  $\delta \in \{1, \dots, D\}$ , whether it should be irrelevant ( $a_i^l[\delta] = 1$ ) to the model’s decision. Note that in the original approach,  $l = 0$ , so that only annotation masks  $a_i^0$  in the input space are considered. The loss function can then simply be augmented by adding an additional regularization term that aims to align the explanation of

each prediction with the corresponding annotation mask:

$$\mathcal{L}_{\text{rrr}}(f_{\theta^t}(x_i), y_i) = \mathcal{L}_{\text{pred}}(f_{\theta^t}(x_i), y_i) + \lambda \mathcal{L}_{\text{reason}}(r_i^{l,t}, a_i^l), \quad (16)$$

where  $\lambda$  is a regularization parameter,  $\mathcal{L}_{\text{pred}}$  denotes the standard prediction loss term between the true and predicted class probabilities including any non-XAI-based regularization terms. In addition to learning to predict accurately through  $\mathcal{L}_{\text{pred}}$ , the reasoning loss term  $\mathcal{L}_{\text{reason}}$  enforces correct reasoning. In their approach, the authors employ explanations  $r_i^{l,t}$  at layer  $l = 0$ , i.e., input gradient w.r.t. log outputs  $r_i^{0,t} = \partial \sum_{c=1}^C \log(f_{\theta^t}(x_i)) / \partial x_i$  as their explanation method of choice, where large values indicate input elements that strongly affect the prediction when changed. Here,  $C$  denotes the number of classes. Therefore, in order to align the model’s decision-making to the ground truth input irrelevancies  $a_i^l$  of each sample, input gradients should be close to zero where  $a_i^l[\delta] = 1$ , i.e., for irrelevant parts. This goal is captured by the reasoning loss term  $\mathcal{L}_{\text{reason}}$ :

$$\mathcal{L}_{\text{reason}}(r_i^{l,t}, a_i^l) = \|a_i^l \odot r_i^{l,t}\|_2^2, \quad (17)$$

where  $\|\cdot\|_2$  denotes the  $\ell_2$ -norm and  $\odot$  the element-wise product. We employed the same loss term during Toy Experiment 3 in Section 2.2 with LRP as the explanation method instead of log probabilities input gradients. Note that the  $r_A$  we defined for Toy Experiment 3 defines which features should be *relevant* to the model’s decision and is therefore complementary to the  $a_i^l$  employed here. Using this method, the authors of [48] are able to align model reasoning with human expectations, and even have models generalize on ambiguous datasets. Although, for this purpose, annotation masks have to be provided by a human expert time-consuming. An alternative is discussed for the case if these annotations are not provided, where multiple models are trained sequentially, each constrained to learn qualitatively different reasoning compared to the previous ones. However, this is not only quite costly due to the large amount of training processes, but a human expert is still required to select the model with a desired reasoning.

In [80], the human involvement is completely left out, by simply removing  $a_i^l$  from the equation. Consequently, instead of reasoning, the robustness of models against adversarial perturbations is improved here, although the authors note that the model’s decision boundaries change as a side-effect, implying a different — although not necessarily better — reasoning. The approaches of [48, 80] are largely model-agnostic (as long as the models and explanations are differentiable) and, even though input gradients are used in the original formulation, they are easily generalizable and adaptable to other local XAI techniques.

For instance, to reduce model bias and boost performance in scarce data settings for text classification [76] employ Integrated Gradients (IG) [30] to incorporate priors into the loss function via an XAI-based regularization term. Similarly to [48], an attribution target, i.e., a ground truth attribution is required for each sample (and also w.r.t. each class, which is different to [48]). However, this target annotation is not necessarily binary here, and only needs to be manually specified for samples and w.r.t. classes of interest. The loss function is simply regularized using the squared distance between each attribution and attribution target. Based on Contextual Decomposition [82], the authors of [77] introduce Contextual Decomposition Explanation Penalization (CDEP), where explanations are utilized in a similar manner, regularizing by using the absolute difference between explanations and ground truth attributions. They note that obtaining ground truth explanations from human experts may be too costly, and thus propose using programmatic rules to identify important regions. The authors of [77] demonstrate that their method can successfully correct model reasoning on various real-world datasets, such as the ISIC skin cancer classification dataset.

For Visual Question Answering (VQA), [83] assign scalar importance scores to region proposals (i.e., image regions that the model proposes to utilize for answering a given question) by summing over the gradients of the ground truth output w.r.t. proposal features using a modified version of the feature map importance computation proposed for Grad-CAM [69]. The region proposal importance scores are then compared to human attention scores. For this purpose, the ranking of region proposal importances is enforced to be similar to the human baseline through a ranking loss term, in addition to the standard loss term promoting high task performance. In its original formulation, the approach of [83] is restricted to their modified version of Grad-CAM to obtain network importance scores, however, given a method to reduce arbitrary explanations to scalar importance scores (e.g., through the ratio of importance inside the proposal region and total importance, in the same manner that the authors reduce human attention maps to scalar values), any local explanation method would be applicable.

In the context of text classification, [78] further pursue the idea of XAI-dependent loss regularization by introducing three additional regularization terms, with the respective aims of reducing the impact of irrelevant input features, making explanations uncertain if the input contains no important features, and offering sparse explanations. Here, a representation erasure based method [84] is applied to obtain explanations. While the first two regularization terms again require (binary) human ground truth annotations for each sample, the last regularization term is only applied to samples where this annotation is not available. Using this method, [78] achieve models that not only have less biased reasoning, but are also able to generalize better.

By regularizing the loss function so that the attributions themselves become more robust, [81] are simultaneously able to increase the robustness of the predictions a model makes, as well as align the

model’s reasoning more with human perception. Their experiments are based on the IG attribution method, which is computed between an input and a reference input. By minimizing the IG attributions between an input and references close to it in the loss function, similar attributions for similar inputs are enforced in an automated manner, thereby increasing model robustness. However, the approach of [81] is relatively constrained to IG attributions, due to the specificity of the proposed loss regularization to this XAI-method. Similarly, [85] obtain higher-quality (local) explanations by reducing low gradient noise. For this purpose, they add a regularization term to the loss function that encourages similar predictions for the original inputs and inputs where the parts with low gradient values are masked. The achieved models are able to distinguish better between informative and uninformative features, resulting in clearer explanations. In contrast to many other approaches that augment the loss function, the technique proposed in [85] does not require any ground truth explanations to be provided.

The authors of [79] formalize the introduction of attribution priors into the loss function, thereby providing a generalized framework for XAI-based loss regularization approaches, including the ones mentioned above, and explore this framework in various concrete settings. They summarize a XAI-regularized loss function in general as

$$\mathcal{L}_{\text{xai-reg}}(f_{\theta^t}(x_i), y_i) = \mathcal{L}_{\text{pred}}(f_{\theta^t}(x_i), y_i) + \lambda \zeta(r_i^{l,t}), \quad (18)$$

where  $\mathcal{L}_{\text{pred}}$  is the prediction loss between ground truth and predictions,  $\lambda$  is a scalar factor, and  $\zeta(r_i^{l,t})$  denotes an arbitrary scalar-valued penalty function on the attributions. This formulation is general enough to include cases where human ground truth attribution annotations are available, and cases where these annotations are not available. The authors of [79] demonstrate how this approach can be employed to improve model robustness, efficiency through sparsity, and performance, although their formulation is general enough to allow for improving even more properties.

In order to improve models based on XAI, approaches that regularize the loss function are the most researched, as these types of methods are not only extremely versatile in the improvement goals that can be accomplished and the XAI-techniques that can be utilized, but also comparatively easy to implement and inherently model-agnostic — apart from the specific XAI-methods potentially requiring access to internal model parameters. In terms of efficiency, there is a high variance between approaches: When gradient-based attributions are used during training in the regularization term, including modified back-propagation approaches, second-order gradients need to be computed to optimize the loss [77], which moderately increases computational costs. The inclusion of attributions into the loss function further imposes the restrictions that only explanation methods that are differentiable w.r.t. the model weights can be employed, and that the model needs to be twice differentiable (a condition which most current DNN architectures fulfill). Some approaches require ground truth attribution annotations by human experts, often for each single sample, which are time-consuming to obtain, or even infeasible for large datasets. However, some human guidance may be required for improvement goals such as reasoning, since valid and invalid input features are generally impossible to determine based on the available training data only. Note also, that [37, 38] showed that explanations can potentially be manipulated so that they do not necessarily reflect the actual classifier behavior. Thus, a model that is trained via an XAI-regularized loss in order to correct its reasoning could theoretically simply learn to emulate the correct explanations while still basing its predictions on invalid input features.

**Loss Scaling.** While above methods introduce a regularization term into the loss function in order to achieve various effects, in the case of imbalanced data, the class-wise losses can simply be scaled based on information offered by XAI. The approach of [51] — which is already discussed in-depth in Section 3.1 — employs class-wise factors based on LRP (although other local XAI methods can in theory be substituted) for this very purpose. As a result, the model’s convergence speed is improved in an automated manner. Similarly to regularizing the loss, scaling losses does not require any access to the model’s internal parameters. This approach is, however, much less general, since a variety of model properties can be controlled via an added regularization term, but simple scaling factors on the loss are only able to change training behavior in a comparatively limited fashion.

### 3.4 Augmenting the Gradient

The gradient determines the direction and speed with which a model’s parameters are updated during the backward pass. By modifying either the intermediate feature gradients, or the parameter gradients directly, as described in Equations (7) and (8), the backward flow of weight updates can be controlled, improving convergence behavior and performance. Compared to the intermediate feature masking approaches presented in Section 3.2, where a mask is obtained from intermediate explanations in order to weigh *features* during the *forward pass*, a similar mask can be computed based on XAI, denoting the importance of *gradients* during the *backward pass*. While all gradient transformation approaches seek to alter the ratio with which model parameters are updated, two distinct types of methods can be distinguished here:

Due to their shape being the same as the explained intermediate features, explanations can directly be employed to obtain importance scores of the feature gradients  $\frac{\partial \mathcal{L}(f_{\theta^t}(X), Y)}{\partial f_{\theta^t}^l(X)}$ , see Equation (7) (*top*),

and augment them accordingly. By doing this, parameter updates of all layers below the augmentation are affected, as is the case for, e.g., [86]. Alternatively, parameter gradients  $\frac{\partial \mathcal{L}(f_{\theta^t}(X), Y)}{\partial \theta^{l,t}}$ , see Equation (8) (*bottom*), can be augmented directly based on XAI. In order to do this, weight-wise importance scores first need to be obtained from intermediate explanations, which is not always trivial due to the shape mismatch, but nevertheless employed by, e.g., [52]. In contrast to augmenting the feature gradients, only the weight update at the augmented layer is affected here.

In the context of Generative Adversarial Networks (GANs), [86] propose XAI-GAN, which improves upon the generator optimization by augmenting the feature gradients at a singular intermediate layer. In standard GANs, the generator  $g$  is tasked with fooling the discriminator  $d$ , so that, given a noise sample  $z_i, i \in \{1, \dots, N\}$ , a generated sample  $g_{\theta_g^t}(z_i)$  cannot be distinguished by  $d$  from real samples  $X$ . Usually, the generator and discriminator are trained in an alternating fashion, with the parameters  $\theta_g^t$  of  $g$  being updated by backpropagating the loss of the discriminator prediction  $d_{\theta_d^t}(g_{\theta_g^t}(z_i))$  through the discriminator, yielding the gradient of the generated example  $\nabla_{g(z_i)} = \frac{\partial \mathcal{L}(d_{\theta_d^t}(g_{\theta_g^t}(z_i)), y_i)}{\partial g_{\theta_g^t}(z_i)}$ . The generator gradients  $\nabla_g$  are then simply computed from  $\nabla_{g_{\theta_g^t}(z_i)}$ , in order to update the generator weights. However, [86] suggest to first augment  $\nabla_{g_{\theta_g^t}(z_i)}$  with a importance mask  $M^t$ , which is obtained from the explanation of the discriminators prediction:

$$\nabla'_{g_{\theta_g^t}(z_i)} = \nabla_{g_{\theta_g^t}(z_i)} + \lambda \nabla_{g_{\theta_g^t}(z_i)} M^t, \quad (19)$$

where  $M^t$  denotes the importance of each feature of  $g_{\theta_g^t}(z_i)$  towards the discriminator's decision, and  $\lambda$  is a scalar hyperparameter.

In contrast, [52] propose various variants of parameter gradient augmentation. Based on LRP, the authors compute an attribution  $r_i^{l,t}$  for each layer  $l$  in a model. For two layers  $l$  and  $l+1$ , with  $M$  and  $N$  neurons, respectively, and corresponding attribution maps  $r_i^{l,t} \in \mathbb{R}^M$  and  $r_i^{l+1,t} \in \mathbb{R}^N$ , weight-wise attribution scores are then obtained via a simple dot product, solving the shape mismatch problem discussed above as follows (note that this equation would require  $M = N$ ):

$$r_{w,i}^{l,t} = r_i^{l,t} \cdot r_i^{l+1,t}. \quad (20)$$

Then,  $r_{w,i}^{l,t}$  is simply normalized into the interval  $[0, 1]$  by division through its sum, yielding  $(r_{w,i}^{l,t})^{l,t}$ , and can be employed to augment the gradient, e.g., by using weight-wise learning rates for a simple multiplication:

$$(\theta')^{l,t} = \theta^{l,t} - \eta \frac{\partial \mathcal{L}(f_{\theta^t}(x_i), y_i)}{\partial \theta^{l,t}} r_{w,i}^{l,t}, \quad (21)$$

where  $\eta$  denotes the scalar learning rate. The authors of [86] demonstrate the variability w.r.t. the employed XAI technique by using their method of feature gradient augmentation with LIME, DeepSHAP [34], or DeepLIFT [29]. They obtain generator models that do not only produce higher quality samples, but do so much more efficiently, and can therefore be trained using only a fraction of the data compared to standard GANs. The authors of [52] suggest multiple variations of their technique based on the weight-wise importance mask. They test their method on various image classification datasets, and report similar or better performance and convergence — although not necessarily faster convergence, as well as higher confidence of the resulting models, compared to some state-of-the-art optimization schemes such as Stochastic Gradient Descent (SGD) [87] and Adadelta [88]. However, their solution to obtaining weight-wise importance scores is based on a dot product between the explanations of consecutive layers — and as such is not applicable if these consecutive layers are not fully-connected or do not contain the same number of neurons. Using the outer product instead would allow for applying this method to dense layers of arbitrary neuron numbers, although other layer types are still not considered. Both methods can in theory be applied using any XAI technique that is able to provide layer-wise explanations.

By not considering all gradients equally during backpropagation and instead using explanations to select the most important ones, XAI-based gradient augmentation methods are able to affect the direction of each learning step, leading to distinct effects, such as better performance and convergence, or even increased data efficiency. Since most of these methods do not require any human involvement, they can easily be employed in order to achieve better models without requiring significantly more training time. However, these methods are generally not model-agnostic, simply because access to a model's internal parameters is required by definition.

### 3.5 Augmenting the Model

Most of the above augmentation categories are employed during training in order to improve the model. However, even after obtaining a good model, some undesirable properties may persist, such as a large number of parameters and thus the large required disc space and high computational effort. These properties depend on the model definition itself and can thus best be improved upon by augmenting the model, as described by Equation (9).

Algorithms for pruning or quantization rely on an estimation of each parameter’s importance to the model’s decision-making and performance. As such, the information offered by XAI naturally presents such a criterion, and can therefore be leveraged in order to increase model efficiency in terms of required computational cost of inference or storage space. For this purpose, XAI-methods that can provide layer-wise explanations are required. Based on these, the connectivity and parameters of a model are altered. To prune a trained model, existing approaches [42, 43] compute intermediate attributions, e.g., for a small number of reference samples, and average these to obtain a pruning criterion in the form of importance scores. Consequently, the neurons or filters with the lowest importance are pruned first, yielding a more storage-efficient model. Similarly, the same importance scores can be leveraged in order to quantize the model’s weights, and thus reduce their memory requirements. With this methodology, [42] propose a pruning criterion based on LRP, which offers intermediate explanations at any layer of a given model. They compute attributions w.r.t. the respective true class of each sample, and obtain their importance-score based pruning criterion by averaging the (absolute) relevances over the reference samples. Employing this method, the authors of [42] are able to successfully remove unimportant units while preserving performance, both when fine-tuning and not fine-tuning the model after pruning, and, in the latter case, vastly outperform other state-of-the-art approaches [89–91].

Similarly, [43] utilize DeepLIFT to obtain importance scores for neural network pruning. After first normalizing these scores using the  $l_1$ -norm for each layer to be considered, the neurons with the lowest importance scores can then be pruned accordingly. Additionally, [43] propose to employ the DeepLIFT scores in order to quantize a model. Here, either weight sharing is applied by k-means clustering weights [92] using a DeepLIFT-weighted mean squared error, where all weights within a cluster then share the same value for quantization, or mixed-precision integer quantization is employed, where the bit-precision of weights is reduced depending on their DeepLIFT scores, leading to a significant boost to storage efficiency.

The authors of [53] employ explainability in order to improve upon Entropy-constrained Quantization (ECQ) (a generalization of EC2T [93]). As a clustering-based quantization algorithm, ECQ not only considers the distance of each weight to each centroid, but additionally promotes sparse assignments due to an entropy term based on the fractions of weights close to each centroid. However, this may also lead to significant model degradation if important weights are assigned to zero, as weight magnitude does not always indicate importance. By additionally considering weight-wise importance scores obtained through LRP for the zero-centroid, [53] are able to increase model storage efficiency by generating low bit width and sparse networks, and simultaneously preserving or even improving performance.

In contrast to the above approaches, the authors of [94] utilize XAI for transferring knowledge, building a completely different model with beneficial properties and similar behavior instead of altering a singular model. More specifically, their technique, Adaptive Wavelet Distillation (AWD), transfers knowledge from a pre-trained DNN into a learnable wavelet transform. This is achieved by integrating an interpretation loss term into the optimization problem for obtaining the wavelet functions, based on Transformation Importance (TRIM) [95] and saliency [96], that ensures sparse attributions in the space of wavelet coefficients, i.e., forcing the wavelet transformation to encode model predictions as concisely as possible.

Since computing LRP and DeepLIFT only requires a modified backward pass while saliency simply requires computing gradients, and since no human interaction is required, the approaches of [42], [43], and [53] are computationally extremely efficient. However, the inherent optimization problem for AWD [94] requires evaluation over multiple data points and coefficients in the wavelet representation, making it comparatively expensive. As the authors of [42] note, LRP-based pruning can improve upon both computational inference cost and model storage requirements, depending on whether the pruning focuses on convolutional or dense layers, respectively. Using their proposed methods for XAI-based pruning and quantization, the authors of [43] are able to optimize memory usage and latency in an automated fashion, and apply their XAI-based criterion to a wide range of pruning and quantization variations. By including XAI into existing quantization techniques, the authors of [53] are able to significantly reduce the trade-off between efficiency and performance. The authors of [94] show that their resulting wavelet transformations lead to far smaller, simplified, more computationally efficient, and more inherently interpretable models, while simultaneously preserving performance. While above approaches employ one specific XAI method each, any explanation technique that is able to provide intermediate attributions can be substituted in theory, although this may affect computational efficiency and quality of the resulting pruning or quantization criterion.

By changing a model’s parameters based on the information offered by XAI, model efficiency in terms of storage cost and inference speed is easily increased. Similarly to XAI-dependent intermediate feature augmentation methods, above model augmentation techniques require layer-wise explanations, and are by definition not model-agnostic, as access to the model’s parameters is required. However, they are applicable in an automated fashion, and often do not even need any finetuning, making their usage effortless and time-efficient.

### 3.6 Approaches not Considered

In this review, we only focus on model improvement methods where XAI is employed to achieve that goal. Of course the research community investigating ways to design and train more performant, reliable and

efficient models is much broader. For instance, approaches such as [22, 97–100] utilize human knowledge to providing better annotations, and thus achieve better performing models that make decisions for better reasons. While not necessarily providing additional information beyond labels, techniques from the field of human-machine interaction [101–104] also leverage external knowledge, but expect a (potentially human) expert to actively interact with the model. This may slow down the learning process but allows for specific, isolated, and more targeted labels or corrections to be given in order to learn with a lower sample complexity [105]. These approaches which rely on human knowledge only (i.e., do not employ XAI) are not considered here. Also the whole research field aiming at training models that inherently explain themselves [58, 106–109] is out of scope of this review. This can also be considered an improvement, since explainable models do not necessarily trade-off in terms of performance and may even have more desirable robustness and reasoning properties [106]. Nevertheless, these approaches tend to make significant restrictions to the model architecture, and they do not employ XAI to improve models, but instead improve the model’s explainability itself.

### 3.7 Discussion

The various previously described approaches for XAI-based model improvement not only differ in terms of the model property that is improved, but also w.r.t. what exactly is augmented, i.e., the data distribution, the intermediate features during the forward pass, the loss function, the gradients during the backward pass, or the model after training. An overview over this two-dimensional categorization is shown in Table 1. Note that this table is quite sparse, suggesting that each type of augmentation is only suited for improving specific model properties. XAI-based loss augmentation seems to be extremely versatile, being able to improve model performance, robustness, efficiency, and reasoning. This is not surprising, since many of the above properties can be simply expressed as an additional regularization term. On the other hand, XAI-based model augmentation approaches, which are generally applied after training, and include short finetuning at most, only seem to improve model efficiency — albeit with remarkable success [42, 53]. Other properties, which are generally decided during training, therefore do not seem influenceable through model augmentation.

Table 1: Overview and Categorization of Approaches that aim to improve ML-models using XAI.

	[49]	[50]	[56]	[61]	[51]	[64]	[66]	[67]	[45]	[68]	[46]	[15]	[48]	[80]	[76]	[77]	[83]	[78]	[81]	[85]	[79]	[86]	[52]	[42]	[43]	[53]	[94]	
Performance				X		X	X	X	X	X	X				X	X	X	X		X	X	X	X					
Convergence					X																				X			
Robustness															X				X		X							
Efficiency							X													X	X				X	X	X	X
Reasoning	X	X	X	X			X		X	X	X		X	X	X	X	X	X	X									
Equality					X																							



As suggested by Table 1, XAI can be employed to improve upon a multitude of model properties. Explanations are able to measure the importance of parts of the input or intermediate features towards a model’s decision — and can therefore be viewed as an additional and high-dimensional measurement for the discussed properties, depending on the application. They thus allow for a better (compared to just relying on the prediction error) control of the model behaviour. However, while the chances in leveraging XAI to improve ML-models seem extremely promising, there also exist a number of pitfalls and limitations: The quality of the desired improvement to the model directly depends on the quality of the explanation itself. If the explainer does not provide the expected information about the model in a sufficient manner, improving model properties may be impossible. Furthermore, if augmentations are applied during training, there is a danger of the model overfitting on uninformative explanations. More concisely, some augmentations (e.g., intermediate feature or gradient augmentations) rely on explanations as a feature importance measure during training. However, if the model is untrained and makes random decisions, the explanations may not be meaningful in terms of important intermediate features, and thus throw off the model’s learning trajectory [45]. Some model properties, e.g., correct reasoning, cannot be improved via XAI alone, because valid input features and unwanted biases can often not be distinguished from the limited domain of the dataset alone and may be subjective. In this case, the ground truth needs to be provided by humans, to be compared to the XAI-“measurements”, making such methods often extremely time-consuming and costly, up to the point of infeasibility for large datasets. Moreover, [37, 38] showed that explanations can be manipulated while keeping predictions intact. This is especially dangerous for loss regularization

approaches, where a minimization is directly performed using the explanation, potentially enabling the model to learn to emulate explanations while still predicting for the wrong reasons.

Nevertheless, if above limitations are kept in mind, XAI can be leveraged to improve current ML-models significantly.

## 4 Demonstrative Examples

The approaches discussed in Section 3 are quite heterogeneous in terms of improved property, augmented location, human involvement, and degree of improvement. In the following section, we aim to investigate how and under which conditions XAI can be employed to improve specific properties of models trained on highly complex datasets, as well as potential caveats and drawbacks. For this purpose, we focus — as an example — on the methods introduced in [51] (data augmentation) and [45] (intermediate feature augmentation). Finally, we derive recommendations and lessons learned when employing XAI to improve ML-models.

### 4.1 Example 1 (Model Performance)

In the following, we consider and evaluate the usage of methods from XAI during the training step of few-shot classifiers with the aim to improve the model accuracy at test time. For this purpose, we employ the method of [45] (described in Section 3.2) and extend it by using several local XAI-methods, in addition to LRP. The few-shot learning setup was chosen because, firstly, it emulates the human ability to transfer learned knowledge to unseen tasks, and, secondly, it is usually employed with a small number of samples for each seen class. It is of interest for applications in problem settings with a large number of classes with only few samples available, such as tagging of images or text streams, where it could be applied to rare single tags or sets of tags, and, generally, problems with a large number of distinct configurations.

The goal of this experiment is to evaluate the general suitability of XAI to improve the accuracy of models when faced with few sample settings and out-of-domain data. We took two models from [45], the RelationNet [110] and a graph-based few-shot classification model [111], and extended them with several approaches from explainable AI, namely Guided Backpropagation [112], Gradient $\times$ Input, Grad-CAM and Guided Grad-CAM for the RelationNet, and Guided Gradient $\times$ Input for the model from [111], LRP and the natively trained base models. We applied each explanation method at the same place, as with [45], that is from the classifier output to the input to the relation module for the RelationNet and from the classifier output to the input to the graph classification module for the model from [111]. When speaking of applying Gradient $\times$ Input and Grad-CAM, we used as input the input features for the aforementioned modules and the gradient until these input features. We computed explanations at training time for each class which has a prediction probability above the inverse of the number of classes which is the threshold for guessing in a few-shot problem. This was done to be consistent with the backpropagation of LRP in [45]. At test time we used the saved prediction models and computed predictions without explanations. We trained each model on the mini-ImageNet dataset [113] and evaluated its few-shot accuracy at test time on mini-ImageNet and four out-of-domain datasets, Stanford Cars [114], CUB-2011 birds [115], Places [116] and Plantae [117]. We used the same training parameters as in [45], however, repeated training with five different random seeds and report the average over those, as well as the standard deviation over the five runs.

As shown in Table 2, model performance in the few-shot context can be reliably improved through application of XAI. Although the increase in accuracy is often only marginal (i.e., < 1%), this observation consistently holds with only minor variations across datasets, models, and seeds (e.g., cf. results for LRP). However, there are considerable differences between explanation techniques. For instance, LRP and Gradient $\times$ Input (if the diverged seeds are disregarded) lead to significant improvements in the majority of settings, i.e., in 9/10 and 8/10 cases, respectively. On the other hand, e.g., Guided Backpropagation only leads to a better performance in 3/10 of the considered settings. This may be due to the fact that Guided Backpropagation is the only of the applied XAI-methods that does not take intermediate features into account, and instead only expresses a modified sensitivity (which — except for the influence of ReLUs being turned on or off — exclusively relies on the model parameters). In any case, the reasons behind the observed differences in suitability of XAI methods for model improvement are subject to further research.

### 4.2 Example 2 (Model Equality)

Datasets that reflect realistic circumstances — such as the Adience [74] benchmark dataset of unfiltered faces or the Pascal VOC 2012 challenge dataset [118] — often do not contain the same amount of examples for each class. If this imbalance is too significant, it can lead to models trained on those datasets overfitting on majority classes while largely ignoring minority classes, since modern ML-systems tend to minimize error criteria estimated over *all* available data, without class-wise distinctions. For reference, the extremely imbalanced distribution of samples per class for Adience and Pascal-VOC is visualized in Figure 4 (*top*), which easily results in a skewed class-wise performance over training of a naively trained model (*bottom*).

Table 2: Performance of RelationNet (*Top*) and Graph Classification-based model (*Bottom*) trained for few-shot classification with and without XAI-based support. In each cell, the mean and standard deviation of the accuracy over five seeds is reported. For Gradient $\times$ Input and RelationNet two of the seeds diverged, likely due to too high learning rate. The average and standard deviation of the three undiverged seeds are reported in brackets. The **bold** numbers indicate the highest mean measurement of the row, the underlined numbers the second highest. The last row counts the number of settings in which the respective XAI-methods improved upon the baseline.

Dataset	Baseline	LRP	GBP	Grad $\times$ Inp	GradCam	GuiGrCAM
Cars	<u>39.10</u> $\pm$ 0.69	<b>39.56</b> $\pm$ 0.86	37.31 $\pm$ 1.15	31.35 $\pm$ 10.38 (38.92 $\pm$ 0.79)	37.96 $\pm$ 1.17	37.96 $\pm$ 1.17
CUB	57.69 $\pm$ 1.15	<b>58.27</b> $\pm$ 0.44	56.49 $\pm$ 1.37	42.88 $\pm$ 20.88 (58.13 $\pm$ 0.40)	57.08 $\pm$ 1.66	<u>58.00</u> $\pm$ 0.71
MiniImg	72.25 $\pm$ 0.23	<b>72.65</b> $\pm$ 0.55	71.69 $\pm$ 0.72	51.66 $\pm$ 28.90 (72.76 $\pm$ 0.70)	72.38 $\pm$ 0.76	<u>72.63</u> $\pm$ 0.51
Places	64.75 $\pm$ 0.36	64.74 $\pm$ 0.12	63.95 $\pm$ 0.74	47.03 $\pm$ 24.68 (65.05 $\pm$ 0.24)	<u>64.80</u> $\pm$ 1.02	<b>64.98</b> $\pm$ 0.70
Plantae	<u>47.15</u> $\pm$ 1.01	<b>48.07</b> $\pm$ 0.84	45.26 $\pm$ 0.75	35.77 $\pm$ 14.39 (46.28 $\pm$ 0.27)	46.22 $\pm$ 1.14	46.77 $\pm$ 0.64
Better than Base	–	4	0	0 (3)	3	3

Dataset	Baseline	LRP	GBP	Grad $\times$ Inp	GuiG $\times$ Inp
Cars	44.95 $\pm$ 1.16	<u>45.50</u> $\pm$ 0.70	44.28 $\pm$ 0.61	<b>45.56</b> $\pm$ 0.82	45.36 $\pm$ 0.64
CUB	63.86 $\pm$ 1.74	64.96 $\pm$ 1.46	64.65 $\pm$ 1.19	<b>65.19</b> $\pm$ 0.35	64.88 $\pm$ 1.00
MiniImg	80.33 $\pm$ 0.83	80.40 $\pm$ 0.29	80.02 $\pm$ 0.92	<b>81.30</b> $\pm$ 0.92	<u>80.71</u> $\pm$ 0.34
Places	70.38 $\pm$ 1.17	72.09 $\pm$ 0.51	71.06 $\pm$ 0.51	<b>72.63</b> $\pm$ 1.07	71.59 $\pm$ 0.60
Plantae	55.60 $\pm$ 1.02	<b>57.30</b> $\pm$ 0.96	55.86 $\pm$ 0.94	<u>56.04</u> $\pm$ 1.37	55.80 $\pm$ 0.72
Better than Base	–	5	3	5	5

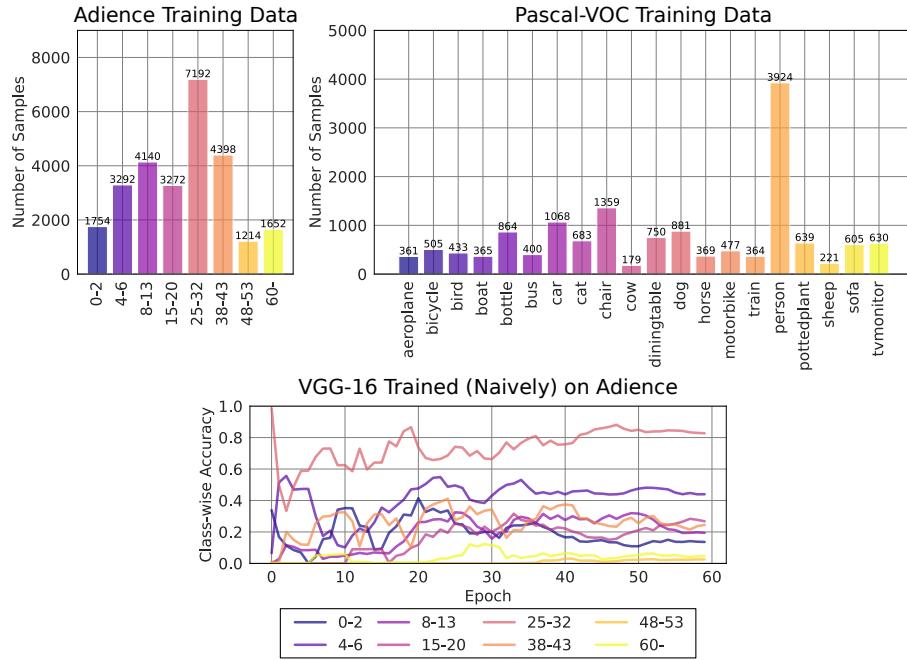


Figure 4: Visualization of class-wise (training set) distributions of the Adience [74] benchmark dataset of unfiltered faces (*top left*) and the Pascal VOC 2012 challenge dataset [118] (*top right*). If the available data is this imbalanced, models trained naively on it easily overfit on majority classes while ignoring minority classes (*bottom*).

However, as shown by [51], the class-wise performance of a model can be estimated through explanations (here specifically LRP), even before any changes in class-wise performance metrics are detectable. In the following, we employ their method of XAI-guided imbalance mitigation to re-balance the distribution of input data based on the entropy of and MSE-distance between explanations. Details on the data and models used, on the employed explanation and augmentation methods, and about the training scheme and evaluation metrics can be found in B.2.

Results are shown in Figures 5 and 6. In each of these figures, the *top* panel depicts results for VGG-16 [39], while the *bottom* panel shows results for ResNet-50 [119]. The balance scores  $b_p$  over mini-epochs (refer to B.2) are shown to the *left* of each panel. Here, results are averaged over the employed resampling criteria (Entropy and MSE-distance of attributions, refer to [51]), and a sliding window of 10 mini-epochs was applied to reduce noisiness of the lines. Standard deviation and mean of class-wise performances of the final models are depicted in the *center* of each panel. Here, a lower standard deviation and a higher mean performance are desirable, that is, models that score more towards the top right of the plot have better and more similar class-wise performance, since the x-axis is inverted. To the *right* of each panel, the predicted probabilities for the true class label are shown for the final models of all investigated settings, averaged over the respective test set. Here, minimum and maximum average probabilities are shown in black (a higher minimum and lower maximum are better), and the mean over all classes in red (higher is better). The green number denotes the average difference in predicted probability between classes (lower is better).

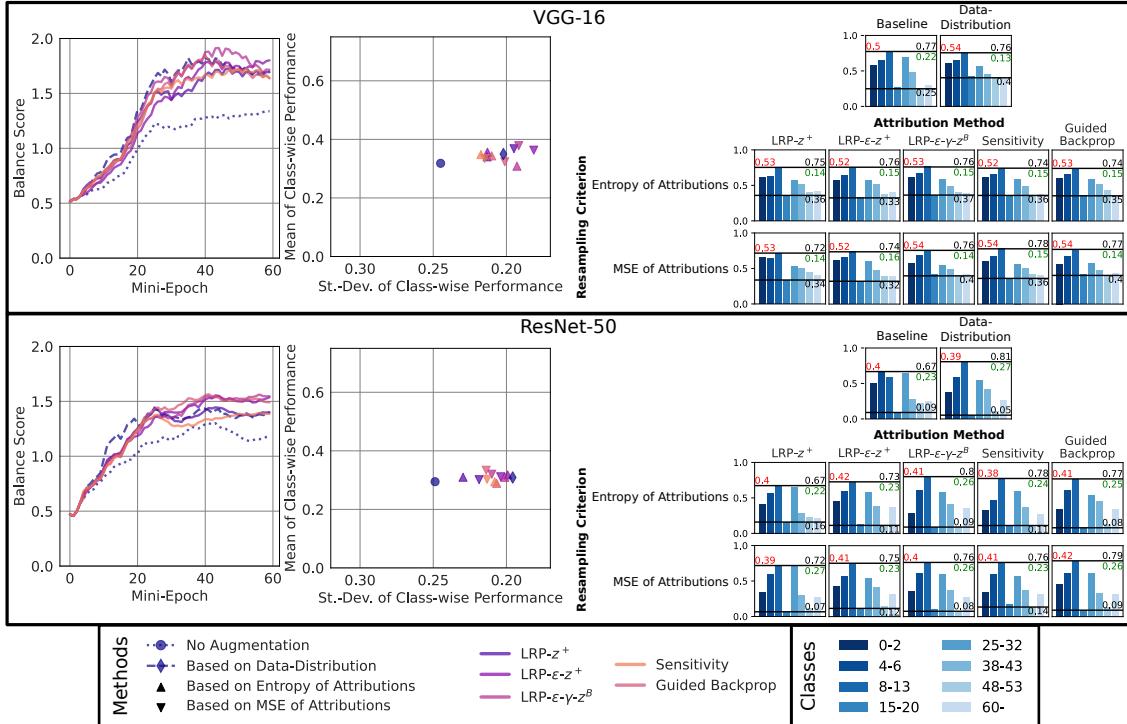


Figure 5: Redistribution of samples per mini-epoch based on attribution, compared to an unaugmented baseline and a control setting (based on class-wise datadistribution) on the Adience dataset. Results are shown for VGG-16 (*top*) and ResNet-50 (*bottom*). In terms of performance, all rebalancing techniques achieve a higher balance score (*left* of each panel), as well as a lower standard deviation (*center* of each panel, note that smaller values are to the right on the standard deviation axis), implying a more equalized class-wise performance. The predicted probabilities for the true class label are shown to the *right*. Here, minimum and maximum values are indicated by the *black* lines and numbers. The mean over all classes is shown in *red*, and the average difference between classes in *green*. For VGG-16, the predicted probability of the true class becomes more balanced in all settings compared to the baseline, however, this is not the case as reliably for ResNet-50.

For the Adience dataset (see Figure 5), all augmented models achieve a higher balance score over the course of training than the unaugmented baseline (*left*), both for VGG-16 and ResNet-50, although the effect is far more significant for the former. For the final models, the augmented models achieve a similar or slightly higher mean class-wise performance compared to the baseline, but a far lower standard deviation, implying increased equality (*center*). Again, this effect is more pronounced for VGG-16 than for ResNet-50. Looking at the average predicted true class probabilities of the final models (*left*), the baseline has the lowest minimum predicted true class probability for VGG-16. This is strongly improved upon by the redistribution based on data distribution, and by the explanation-based variants, especially  $\text{LRP-}\varepsilon\text{-}\gamma\text{-}z^B$  and Guided Backpropagation. There is, however, a notable trade-off between increasing the minimum and decreasing the maximum predicted probability. While all augmentations lead to increased minimum probabilities, most of them also reduce the maximum probability. However, all augmentation methods seem to increase the average predicted probability, while lowering the average distance between class-wise probabilities for VGG-16, showing an increase in model equality. The predicted probabilities are far more imbalanced for the ResNet-50 baseline model than for VGG-16, indicating that this model tends to overfit

more in the given setting. While most augmentations increase the minimum predicted probability here, the average across classes is not increased as consistently as for VGG-16. Interestingly, the redistribution based on the data-distribution performs far worse than the baseline, as opposed to the VGG-16 setting, showing inconsistent behavior. Moreover, the average distance between class-wise predicted probabilities never decreases for ResNet-50, as opposed to VGG-16. Investigating the explanations of ResNet-50 more closely, we find that the downsampling shortcuts within ResNet-50, specifically those with kernel-size  $1 \times 1$  and stride 2, lead to a significant artifact for some explanation methods. Some examples for this effect (from the Pascal-VOC dataset) can be found in Figure 7. While computing attributions without considering those shortcuts removes the downsampling artifact, the resulting attributions would also only partially explain the model decisions. However, this artifact may affect the employed attribution-based augmentation methods, making them perform less reliably on ResNet-50 than on VGG-16, although the performance equality is still visibly (albeit less significantly) improved (*center and left*).

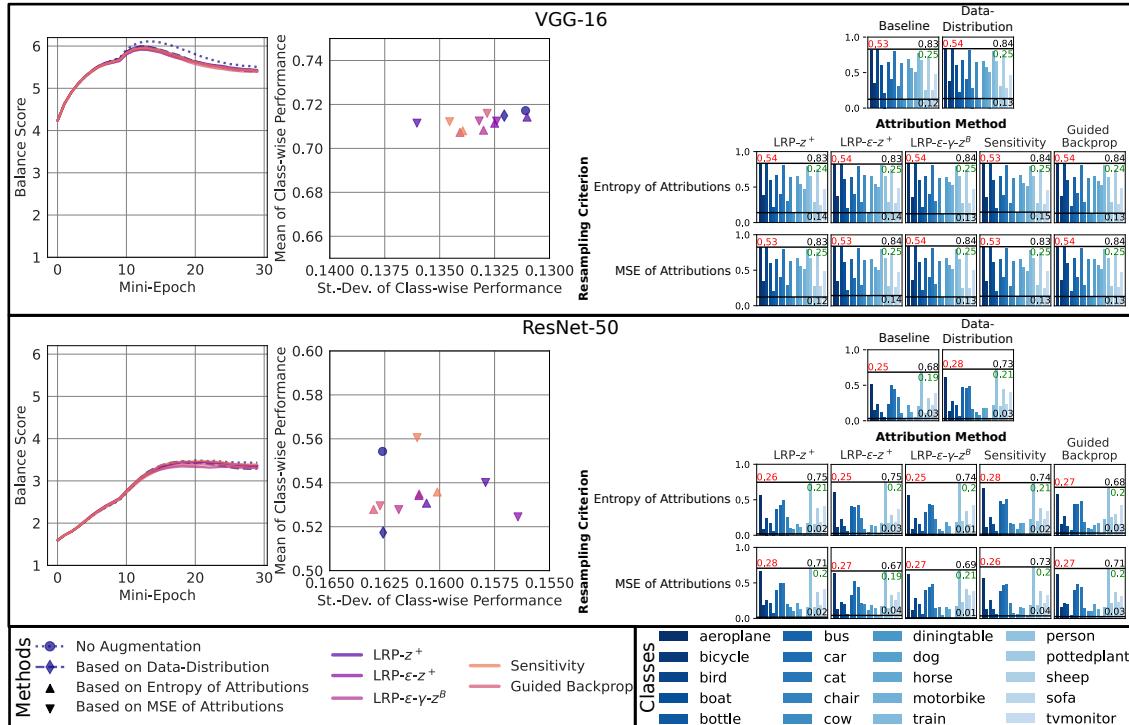


Figure 6: Redistribution of samples per mini-epoch based on attribution, compared to an unaugmented baseline and a control setting (based on class-wise datadistribution) on the Pascal-VOC dataset. Results are shown for VGG-16 (*top*) and ResNet-50 (*bottom*). Here, model balance score (*left* of each panel), and class-wise performance mean and standard deviation (*center* of each panel, note that smaller values are to the right on the standard deviation axis) do not vary much between settings. In fact, the baseline seems to have a slightly more balanced performance for VGG-16 and ResNet-50. The predicted probabilities for the true class label are shown to the *right*. Here, minimum and maximum values are indicated by the *black* lines and numbers. The mean over all classes is shown in *red*, and the average difference between classes in *green*. However, while for VGG-16 a slight improvement in equality is indicated by the predicted true class probabilities, barely any consistent effect is visible for ResNet-50.

For the Pascal-VOC dataset (see Figure 6) the balance score is lower than baseline in all settings. Both for VGG-16 and ResNet-50, rebalancing seems to slightly lower the balance score over mini-epochs (*left*). However, the differences between models are extremely small, and the balance scores are generally far higher than for the Adience dataset in Figure 5) — even for the baseline models. Compared to Adience, the difference between classes is significantly lower for Pascal-VOC, as indicated by the generally lower standard deviations (*center*) compared to Figure 5, both for VGG-16 and ResNet-50. This may be the case because the domain of Pascal-VOC is semantically closer to ILSVCR2012 [70] (since all models used here were pre-trained on ILSVCR2012) than the Adience dataset is. The models therefore perform comparatively well on Pascal-VOC from the start, and do not change during training as significantly as for Adience, an effect also increased by the comparatively small learning rate (0.05). The final augmented models have similar or slightly lower mean class-wise performances for VGG-16 and ResNet-50, a slightly lower standard deviation than the baseline model for VGG-16, and a slightly higher one for ResNet-50 (albeit the considered value range is extremely small here, compared to the experiments on Adience) (*center*). Because samples from Pascal-VOC can contain more than one true class label due to the multi-label setting, the rebalancing of mini-batches employed here may not be as impactful as for Adience, since

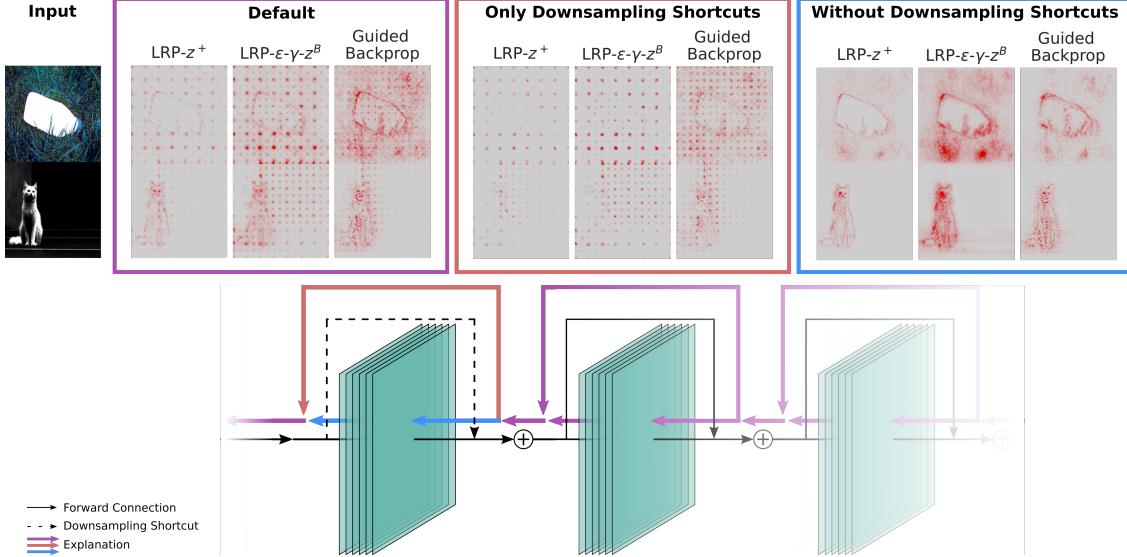


Figure 7: Examples for the downsampling artifact present when explaining decisions of ResNet-50 with various explanation methods. A visualization of the explanation flow through a segment of ResNet-50 is visualized at the bottom. Left to Right: 1. Example input images from the Pascal-VOC Dataset. 2. Default explanations of these samples for  $\text{LRP-}z^+$ ,  $\text{LRP-}\varepsilon\text{-}z^B$ , and Guided Backpropagation without excluding any paths through the model; the downsampling artifact (evenly spaced dots) is clearly visible here. 3. Partial explanations where only the downsampling shortcut paths are used for the modified backpropagation; almost only the artifact is visible here. 4. Partial explanations where the downsampling shortcut paths are excluded from the explanation computation; the downsampling artifact is gone.

when adding or removing samples for one class, other classes may also be affected. However, similar to Figure 5, all augmentations seem to improve upon the baseline in terms of predicted true class probabilities for VGG-16 (higher minimum and average predicted probabilities, with lower or similar average distance between probabilities, compared to the baseline), albeit far less significantly. Interestingly, while minimum probabilities increase, maximum probabilities do not seem to decrease, which may be related either to the reduced impact of the augmentation compared to Adience or to the large amount of classes and the multilabel setting. Again, ResNet-50 seems to overfit more, and augmentations barely improve upon this, assumedly due to the attribution artifacts caused by the downsampling shortcut connections (Figure 7). While the average predicted probabilities (*red*) mostly increase, the average distances between probabilities also mostly increase with augmentation (*green*) while the minimum probabilities mostly decrease. It seems that especially for ResNet-50, the augmentations barely have any effect on class equality in this setting.

### 4.3 Lessons Learned

The above experiments demonstrate that augmentations based on XAI are able to improve model properties such as performance or equality even in highly complex settings. In Section 4.1, XAI was used to achieve small improvements to the test accuracy of few-shot classifiers in a consistent manner. In Section 4.2, explanations were employed to achieve more equal class-wise performances in imbalanced settings, with varying success. We found that in complex settings, the achieved effects may not always be as significant as in toy settings (e.g., compare Toy Experiment 1 in Section 2.2 to Section 4.1). However, there are also often no notable drawbacks to employing XAI-based model improvement in order to, e.g., increase few-shot accuracy by a few percent (Section 4.1).

We further noted that XAI-based techniques can be better than the alternatives, i.e., approaches not based on XAI, but this is not always the case, as demonstrated in Section 4.2, and should therefore not be blindly applied. Explainability methods and model improvement techniques derived from them are highly dependent on various hyperparameters, such as the employed task, dataset, model (see, e.g., Figure 7), and XAI-method. These parameters therefore need to be carefully selected based on the specific setting. For instance, which XAI-method is used to obtain explanations can significantly impact the success of a model improvement technique (Sections 4.1 and 4.2). As such, while XAI-based model improvement techniques can be useful even in complex settings, e.g., to simply achieve higher performance or to affect model properties that are non-trivial and difficult to quantify, since XAI offers detailed information about the model behavior, these methods in turn need to be applied with caution, since they are often sensitive to various hyperparameters and do not always outperform alternatives not based on XAI.

## 5 Conclusion

With the advance of explainability methods that are able to inform about the decision-making behavior of modern ML-architectures, especially DNNs, these methods could serve as tools for improving models beyond simple statistics such as test performance, but have barely been applied beyond explaining decisions and discovering problems in existing models. Only recently, more and more research has been published that employs XAI in practice, augmenting models in order to improve various beneficial (and often complex or intangible) properties such as model reasoning and efficiency.

In this paper, we illustrated how XAI can be used to improve models through various toy examples, and that the resulting effects can be significant and beneficial. We further introduced a formalized categorization of XAI-dependent model augmentations based on the component of the model training process they are applied at, in order to systematically review and compare approaches from existing research. Finally, we investigated the effect of XAI-based augmentations in complex settings, as well as the drawbacks, limitations, and caveats of their application.

XAI can be practically applied to achieve various beneficial effects. In contrast to the generally utilized test performance statistics such as accuracy, precision, or recall, it offers comprehensive feedback on complex properties that can be used to diagnose and significantly improve models. Depending on how the model or the training process are altered, and what the desired goal is, various restrictions may apply. For instance, reasoning is a property that can ensure a (truly) general understanding of the data and task at hand — unaffected by any confounders or biases within the data. While XAI can offer information about which features a model uses to make its predictions, the decision whether these features should be used cannot be derived from data or explanations alone, and requires some additional ground truth, e.g., provided by a human expert. Augmenting the loss function based on XAI is by far the most researched type of augmentation, since it allows for various improvement goals to be easily expressed as regularization terms. However, when explanations are part of these regularization terms, this restricts the number of applicable methods to gradient-based and modified backpropagation techniques, since they need to be differentiable w.r.t. model weights. Furthermore, not every type of augmentation is applicable to directly improve any property. For instance, augmenting the model after training has finished — via pruning or quantization based on XAI — can provide significant improvements to model efficiency, but is not suitable to improve any other properties.

However, we found that despite the various benefits of utilizing the information gained through XAI to augment and improve ML-models, this application is not always trivial. XAI offers useful but complex information that is often extremely dependent on factors such as the employed explanation method, model, dataset, task, and hyperparameters such as normalization, and can therefore be difficult to exploit reliably. As shown by our experiments on relatively complex tasks in Section 4, effects of augmentations in these settings may be comparatively minimal or vary due to above factors. XAI-dependent augmentations may further invoke potentially unexpected side-effects [37, 38], and related techniques need to be applied carefully and with caution. Nevertheless, our experiments also showed that under the right conditions, XAI can provide significant, diverse, and reliable benefits. As such, practically employing explanations to improve ML-models is not only a promising concept, but extremely helpful in improving non-trivial model properties. Nevertheless, it requires careful consideration and further research. While most approaches focus on augmenting a single component of the training process, different augmentations (e.g., data and loss function) do not interfere with each other, and combining multiple ones could provide a significant improvement to a specific property, however, this is subject to future work. Moreover, it should be noted that XAI-based augmentations often aim at improving different, more complex, and less tangible properties than test performance, such as equality or reasoning. These goals are of a semantic nature, and strict test performance in fact often needs to be sacrificed in order to achieve them. For instance, in terms of reasoning, a model that overfits on Clever Hans type of features may achieve a high test performance, but classify based on the wrong evidence. When improving reasoning, these (apparently useful) features are forbidden to the model, and test performance usually decreases as a result. However, since a model that relies on the wrong reasons cannot easily generalize to new data, it is not practically useful in real-world applications. Established test performance metrics may therefore be outdated, as different properties become more and more important, and new metrics may be required. For this purpose, XAI could be a valuable asset, as it is able to provide semantic information about model behavior, and its usefulness for improving complex model properties should therefore be explored further.

## Acknowledgments

This work was supported by the German Ministry for Education and Research (BMBF) [grant numbers 01IS14013A-E, 01GQ1115, 01GQ0850, 01IS18025A and 01IS18037A]; the iToBoS (Intelligent Total Body Scanner for Early Detection of Melanoma) project funded by the European Union’s Horizon 2020 research and innovation programme [grant agreement No 965221]; the Research Council of Norway, via the SFI Visual Intelligence grant [project grant number 309439], and UiO dScience – Centre for Computational and Data Science.

## References

- [1] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nat.*, 521(7553):436–444, 2015.
- [2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90, 2017.
- [3] Mohammad Ali Kadampur and Sulaiman Al Riyae. Skin cancer detection: Applying a deep learning based model driven architecture in the cloud for classifying dermal cell images. *Informatics in Medicine Unlocked*, 18:100282, 2020. ISSN 2352-9148.
- [4] Farman Ali, Shaker El-Sappagh, S.M. Riazul Islam, Daehan Kwak, Amjad Ali, Muhammad Imran, and Kyung-Sup Kwak. A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. *Information Fusion*, 63:208 – 222, 2020. ISSN 1566-2535.
- [5] Kristof T. Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus-Robert Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature Communications*, 8(1):13890, Jan 2017. ISSN 2041-1723.
- [6] Yalin Wang, Zhuofu Pan, Xiaofeng Yuan, Chunhua Yang, and Weihua Gui. A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network. *ISA Transactions*, 96:457 – 467, 2020. ISSN 0019-0578.
- [7] Max Jaderberg, Wojciech M. Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio García Castañeda, Charles Beattie, Neil C. Rabinowitz, Ari S. Morcos, Avraham Ruderman, Nicolas Sonnerat, Tim Green, Louise Deason, Joel Z. Leibo, David Silver, Demis Hassabis, Koray Kavukcuoglu, and Thore Graepel. Human-level performance in first-person multiplayer games with population-based deep reinforcement learning. *arXiv preprint arXiv:1807.01281*, 2018.
- [8] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander Sasha Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Çağlar Gülcöhre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy P. Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in starcraft II using multi-agent reinforcement learning. *Nat.*, 575(7782):350–354, 2019.
- [9] Snehlata Shakya, Sanjeev Kumar, and Mayank Goswami. Deep learning algorithm for satellite imaging based cyclone detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, 13:827–839, 2020.
- [10] Pradeep Hewage, Marcello Trovati, Ella Pereira, and Ardhendu Behera. Deep learning-based effective fine-grained weather forecasting model. *Pattern Analysis and Applications*, 24(1):343–366, Feb 2021. ISSN 1433-755X.
- [11] Farah Shahid, Aneela Zameer, and Muhammad Muneeb. Predictions for covid-19 with deep learning models of lstm, gru and bi-lstm. *Chaos, Solitons & Fractals*, 140:110212, 2020. ISSN 0960-0779.
- [12] Parul Arora, Himanshu Kumar, and Bijaya Ketan Panigrahi. Prediction and analysis of covid-19 positive cases using deep learning models: A descriptive case study of india. *Chaos, Solitons & Fractals*, 139:110017, 2020. ISSN 0960-0779.
- [13] Pierre Stock and Moustapha Cissé. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VI*, volume 11210 of *Lecture Notes in Computer Science*, pages 504–519. Springer, 2018.
- [14] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, 10(1):1096, Mar 2019. ISSN 2041-1723.
- [15] Christopher J. Anders, Leander Weber, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. Finding and removing clever hans: Using explanation methods to debug and improve deep models. *Inf. Fusion*, 77:261–295, 2022.
- [16] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, 2003.

- [17] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie J. Cai, James Wexler, Fernanda B. Viégas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2673–2682. PMLR, 2018.
- [18] Jie Hu, Rongrong Ji, Qixiang Ye, Tong Tong, Shengchuan Zhang, Ke Li, Feiyue Huang, and Ling Shao. Architecture disentanglement for deep neural networks. *arXiv preprint arXiv:2003.13268*, 2020.
- [19] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *J. Mach. Learn. Res.*, 11:1803–1831, 2010.
- [20] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7):1–46, 07 2015.
- [21] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "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, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144. ACM, 2016.
- [22] Omar Zaidan, Jason Eisner, and Christine D. Piatko. Using "annotator rationales" to improve machine learning for text categorization. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA*, pages 260–267. The Association for Computational Linguistics, 2007.
- [23] Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J. Anders, and Klaus-Robert Müller. Explaining deep neural networks and beyond: A review of methods and applications. *Proc. IEEE*, 109(3):247–278, 2021.
- [24] Erico Tjoa and Cuntai Guan. A survey on explainable artificial intelligence (XAI): towards medical XAI. *arXiv preprint arXiv:1907.07374*, 2019.
- [25] Giulia Vilone and Luca Longo. Explainable artificial intelligence: a systematic review. *arXiv preprint arXiv:2006.00093*, 2020.
- [26] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3319–3327. IEEE Computer Society, 2017.
- [27] Fred Hohman, Haekyu Park, Caleb Robinson, and Duen Horng (Polo) Chau. Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations. *IEEE Trans. Vis. Comput. Graph.*, 26(1):1096–1106, 2020.
- [28] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognit.*, 65:211–222, 2017.
- [29] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3145–3153. PMLR, 2017.
- [30] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 2017.
- [31] Luisa M. Zintgraf, Taco S. Cohen, Tameem Adel, and Max Welling. Visualizing deep neural network decisions: Prediction difference analysis. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.
- [32] Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 3449–3457, 2017.

- [33] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*, 41(3):647–665, 2014.
- [34] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774, 2017.
- [35] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.
- [36] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, pages 372–387. IEEE, 2016.
- [37] Ann-Kathrin Dombrowski, Maximilian Alber, Christopher J. Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13567–13578, 2019.
- [38] Christopher J. Anders, Plamen Pasliev, Ann-Kathrin Dombrowski, Klaus-Robert Müller, and Pan Kessel. Fairwashing explanations with off-manifold detergent. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 314–323. PMLR, 2020.
- [39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [40] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6155–6166, 2019.
- [41] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Mach. Learn.*, 109(3):467–492, 2020.
- [42] Seul-Ki Yeom, Philipp Seegerer, Sebastian Lapuschkin, Alexander Binder, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Pruning by explaining: A novel criterion for deep neural network pruning. *Pattern Recognition*, 115:107899, 2021. ISSN 0031-3203.
- [43] Muhammad Sabih, Frank Hannig, and Jürgen Teich. Utilizing explainable AI for quantization and pruning of deep neural networks. *arXiv preprint arXiv:2008.09072*, 2020.
- [44] Huan Wang, Can Qin, Yulun Zhang, and Yun Fu. Emerging paradigms of neural network pruning. *arXiv preprint arXiv:2103.06460*, 2021.
- [45] Jiamei Sun, Sebastian Lapuschkin, Wojciech Samek, Yunqing Zhao, Ngai-Man Cheung, and Alexander Binder. Explanation-guided training for cross-domain few-shot classification. In *Proceedings of the 25th International Conference on Pattern Recognition*, pages 7609–7616, 2020.
- [46] Andrea Zunino, Sarah Adel Bargal, Pietro Morerio, Jianming Zhang, Stan Sclaroff, and Vittorio Murino. Excitation dropout: Encouraging plasticity in deep neural networks. *Int. J. Comput. Vis.*, 129(4):1139–1152, 2021.
- [47] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014.
- [48] Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 2662–2670. ijcai.org, 2017.
- [49] Stefano Teso and Kristian Kersting. Explanatory interactive machine learning. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2019, Honolulu, HI, USA, January 27-28, 2019*, pages 239–245. ACM, 2019.

- [50] Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8):476–486, Aug 2020. ISSN 2522-5839.
- [51] Leander Weber. Towards a more refined training process for neural networks: Applying layer-wise relevance propagation to understand and improve classification performance on imbalanced datasets. Master’s thesis, Technische Universität Berlin, 2020.
- [52] Jin Ha Lee, Ik hee Shin, Sang gu Jeong, Seung-Ik Lee, Muhamamad Zaigham Zaheer, and Beom-Su Seo. Improvement in deep networks for optimization using explainable artificial intelligence. In *2019 International Conference on Information and Communication Technology Convergence, ICTC 2019, Jeju Island, Korea (South), October 16-18, 2019*, pages 525–530. IEEE, 2019.
- [53] Daniel Becking, Maximilian Dreyer, Wojciech Samek, Karsten Müller, and Sebastian Lapuschkin. Ecq<sup>X</sup>: Explainability-driven quantization for low-bit and sparse dnns. In *xxAI Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 13200 of *Lecture Notes in Computer Science*. Springer, 2022.
- [54] Ian Goodfellow, Nicolas Papernot, and Patrick McDaniel. cleverhans v0.1: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 2016.
- [55] Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. Right for the right concept: Revising neuro-symbolic concepts by interacting with their explanations. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 3619–3629. Computer Vision Foundation / IEEE, 2021.
- [56] Srishti Gautam, Marina M.-C. Höhne, Stine Hansen, Robert Jenssen, and Michael Kampffmeyer. This looks more like that: Enhancing self-explaining models by prototypical relevance propagation. *arXiv preprint arXiv:2108.12204*, 2021.
- [57] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian M. Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. In *Workshop on Artificial Intelligence Safety 2019 co-located with the Thirty-Third AAAI Conference on Artificial Intelligence 2019 (AAAI-19), Honolulu, Hawaii, January 27, 2019*, volume 2301 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.
- [58] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan Su. This looks like that: Deep learning for interpretable image recognition. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8928–8939, 2019.
- [59] Abhishek Kumar, Piyush Rai, and Hal Daumé III. Co-regularized multi-view spectral clustering. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 1413–1421, 2011.
- [60] Kun Zhan, Feiping Nie, Jing Wang, and Yi Yang. Multiview consensus graph clustering. *IEEE Trans. Image Process.*, 28(3):1261–1270, 2019.
- [61] Sarah Adel Bargal, Andrea Zunino, Vitali Petsiuk, Jianming Zhang, Kate Saenko, Vittorio Murino, and Stan Sclaroff. Guided zoom: Questioning network evidence for fine-grained classification. In *30th British Machine Vision Conference 2019, BMVC 2019, Cardiff, UK, September 9-12, 2019*, page 17. BMVA Press, 2019.
- [62] Sarah Adel Bargal, Andrea Zunino, Vitali Petsiuk, Jianming Zhang, Kate Saenko, Vittorio Murino, and Stan Sclaroff. Guided zoom: Zooming into network evidence to refine fine-grained model decisions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(11):4196–4202, 2021.
- [63] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *Int. J. Comput. Vis.*, 126(10):1084–1102, 2018.
- [64] Hiroshi Fukui, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Attention branch network: Learning of attention mechanism for visual explanation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10705–10714. Computer Vision Foundation / IEEE, 2019.
- [65] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2921–2929. IEEE Computer Society, 2016.

- [66] Masahiro Mitsuhashi, Hiroshi Fukui, Yusuke Sakashita, Takanori Ogata, Tsubasa Hirakawa, Takayoshi Yamashita, and Hironobu Fujiyoshi. Embedding human knowledge in deep neural network via attention map. *arXiv preprint arXiv:1905.03540*, 2019.
- [67] Dominik Schiller, Tobias Huber, Florian Lingenfelser, Michael Dietz, Andreas Seiderer, and Elisabeth André. Relevance-based feature masking: Improving neural network based whale classification through explainable artificial intelligence. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 2423–2427. ISCA, 2019.
- [68] Andrea Zunino, Sarah Adel Bargal, Riccardo Volpi, Mehrnoosh Sameki, Jianming Zhang, Stan Sclaroff, Vittorio Murino, and Kate Saenko. Explainable deep classification models for domain generalization. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021, virtual, June 19-25, 2021*, pages 3233–3242. Computer Vision Foundation / IEEE, 2021.
- [69] Ramprasaath R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 2016.
- [70] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.
- [71] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5(1):180161, Aug 2018. ISSN 2052-4463.
- [72] Noel C. F. Codella, Veronica Rotemberg, Philipp Tschandl, M. Emre Celebi, Stephen W. Dusza, David A. Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael A. Marchetti, Harald Kittler, and Allan Halpern. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (ISIC). *arXiv preprint arXiv:1902.03368*, 2019.
- [73] Marc Combalia, Noel C. F. Codella, Veronica Rotemberg, Brian Helba, Verónica Vilaplana, Ofer Reiter, Allan C. Halpern, Susana Puig, and Josep Malvehy. BCN20000: dermoscopic lesions in the wild. *arXiv preprint arXiv:1908.02288*, 2019.
- [74] Eran Eidinger, Roee Enbar, and Tal Hassner. Age and gender estimation of unfiltered faces. *Proc. of the IEEE Transactions on Information Forensics Security*, 9(12):2170–2179, 2014.
- [75] Frederik Pahde, Leander Weber, Christopher J. Anders, Wojciech Samek, and Sebastian Lapuschkin. PatCIArC: Using pattern concept activation vectors for noise-robust model debugging. *arXiv preprint arXiv:2202.03482*, 2022.
- [76] Frederick Liu and Besim Avci. Incorporating priors with feature attribution on text classification. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6274–6283. Association for Computational Linguistics, 2019.
- [77] Laura Rieger, Chandan Singh, W. James Murdoch, and Bin Yu. Interpretations are useful: Penalizing explanations to align neural networks with prior knowledge. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8116–8126. PMLR, 2020.
- [78] Mengnan Du, Ninghao Liu, Fan Yang, and Xia Hu. Learning credible deep neural networks with rationale regularization. In *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*, pages 150–159. IEEE, 2019.
- [79] Gabriel Erion, Joseph D. Janizek, Pascal Sturmels, Scott Lundberg, and Su-In Lee. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *arXiv preprint arXiv:1906.10670*, 2020.
- [80] Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1660–1669. AAAI Press, 2018.

- [81] Jiefeng Chen, Xi Wu, Vaibhav Rastogi, Yingyu Liang, and Somesh Jha. Robust attribution regularization. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14300–14310, 2019.
- [82] W. James Murdoch, Peter J. Liu, and Bin Yu. Beyond word importance: Contextual decomposition to extract interactions from lstms. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.
- [83] Ramprasaath Ramasamy Selvaraju, Stefan Lee, Yilin Shen, Hongxia Jin, Shalini Ghosh, Larry P. Heck, Dhruv Batra, and Devi Parikh. Taking a HINT: leveraging explanations to make vision and language models more grounded. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 2591–2600. IEEE, 2019.
- [84] Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016.
- [85] Aya Abdelsalam Ismail, Hector Corrada Bravo, and Soheil Feizi. Improving deep learning interpretability by saliency guided training. volume 34, 2021.
- [86] Vineel Nagisetty, Laura Graves, Joseph Scott, and Vijay Ganesh. xAI-GAN: Enhancing Generative Adversarial Networks via Explainable AI Systems. *arXiv preprint arXiv:2002.10438*, 2020.
- [87] Léon Bottou. Stochastic gradient descent tricks. In *Neural Networks: Tricks of the Trade - Second Edition*, volume 7700 of *Lecture Notes in Computer Science*, pages 421–436. Springer, 2012.
- [88] Matthew D. Zeiler. ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.
- [89] Xu Sun, Xuancheng Ren, Shuming Ma, and Houfeng Wang. meprop: Sparsified back propagation for accelerated deep learning with reduced overfitting. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3299–3308. PMLR, 2017.
- [90] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient transfer learning. *arXiv preprint arXiv:1611.06440*, 2016.
- [91] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.
- [92] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):100–108, 1979.
- [93] Arturo Marban, Daniel Becking, Simon Wiedemann, and Wojciech Samek. Learning sparse & ternary neural networks with entropy-constrained trained ternarization (ec2t). In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3105–3113. IEEE, 2020.
- [94] Wooseok Ha, Chandan Singh, Francois Lanusse, Eli Song, Song Dang, Kangmin He, Srigokul Upadhyayula, and Bin Yu. Adaptive wavelet distillation from neural networks through interpretations. *arXiv preprint arXiv:2107.09145*, 2021.
- [95] Chandan Singh, Wooseok Ha, Francois Lanusse, Vanessa Böhm, Jia Liu, and Bin Yu. Transformation importance with applications to cosmology. *arXiv preprint arXiv:2003.01926*, 2020.
- [96] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014.
- [97] Omar Zaidan and Jason Eisner. Modeling annotators: A generative approach to learning from annotator rationales. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 31–40. ACL, 2008.
- [98] Lisa Anne Hendricks, Kaylee Burns, Kate Saenko, Trevor Darrell, and Anna Rohrbach. Women also snowboard: Overcoming bias in captioning models. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*, volume 11207 of *Lecture Notes in Computer Science*, pages 793–811. Springer, 2018.

- [99] Ye Zhang, Iain James Marshall, and Byron C. Wallace. Rationale-augmented convolutional neural networks for text classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 795–804. The Association for Computational Linguistics, 2016.
- [100] Tyler McDonnell, Matthew Lease, Mücahid Kutlu, and Tamer Elsayed. Why is that relevant? collecting annotator rationales for relevance judgments. In *Proceedings of the Fourth AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2016, 30 October - 3 November, 2016, Austin, Texas, USA*, pages 139–148. AAAI Press, 2016.
- [101] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, 2001.
- [102] Kshitij Judah, Alan Paul Fern, and Thomas Glenn Dietterich. Active imitation learning via reduction to I.I.D. active learning. In *Robots Learning Interactively from Human Teachers, Papers from the 2012 AAAI Fall Symposium, Arlington, Virginia, USA, November 2-4, 2012*, volume FS-12-07 of *AAAI Technical Report*. AAAI, 2012.
- [103] Pannaga Shivaswamy and Thorsten Joachims. Coactive learning. *J. Artif. Intell. Res.*, 53:1–40, 2015.
- [104] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192. PMLR, 2017.
- [105] Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. The true sample complexity of active learning. *Mach. Learn.*, 80(2-3):111–139, 2010.
- [106] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, May 2019. ISSN 2522-5839.
- [107] Tianfu Wu and Xi Song. Towards interpretable object detection by unfolding latent structures. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6032–6042. IEEE, 2019.
- [108] Zhi Chen, Yijie Bei, and Cynthia Rudin. Concept whitening for interpretable image recognition. *arXiv preprint arXiv:2002.01650*, 2020.
- [109] Alina Jade Barnett, Fides Regina Schwartz, Chaofan Tao, Chaofan Chen, Yinhao Ren, Joseph Y. Lo, and Cynthia Rudin. Interpretable mammographic image classification using cased-based reasoning and deep learning. *arXiv preprint arXiv:2107.05605*, 2021.
- [110] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE CVPR*, pages 1199–1208, 2018.
- [111] Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *ICLR*, 2018.
- [112] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015.
- [113] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NIPS*, pages 3630–3638, 2016.
- [114] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2013, Sydney, Australia, December 1-8, 2013*, pages 554–561. IEEE Computer Society, 2013.
- [115] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.
- [116] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE TPAMI*, 40(6):1452–1464, 2017.

- [117] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE CVPR*, pages 8769–8778, 2018.
- [118] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>, 2012.
- [119] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.
- [120] Christopher J. Anders, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. Software for dataset-wide xai: From local explanations to global insights with Zennit, CoRelAy, and ViRelAy. *arXiv preprint arXiv:2106.13200*, 2021.
- [121] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8024–8035, 2019.
- [122] Gil Levi and Tal Hassner. Age and gender classification using convolutional neural networks. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2015, Boston, MA, USA, June 7-12, 2015*, pages 34–42. IEEE Computer Society, 2015.
- [123] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: An overview. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700 of *Lecture Notes in Computer Science*, pages 193–209. Springer, 2019.

## A Details about Toy Experiments

LRP [20] with the  $\varepsilon$ - $z^+$ -composite (i.e.,  $\varepsilon$ -rule for fully-connected layers and  $z^+$ -rule for convolutional layers), as implemented in the Zennit library [120], is employed to compute explanations w.r.t. the true class label of each sample. Each experiment was repeated five times with randomly drawn seeds and averaged over these variations in order to obtain the results shown and discussed below.

### Toy Experiment 1 (Model Performance)

We first generate a five-dimensional dataset describing a binary classification problem consisting of 400 samples, and two gaussian clusters of samples per class. The first two of these input dimensions are independent and informative while the last three consist of random noise. To make classification more difficult, 10% of the class labels are assigned randomly. 50 samples are retained as a test set. The data for this toy experiment is visualized in Figure 1. A four layer fully-connected model (64, 32, 16, and 2 neurons), with ReLU activations after the first three layers, and softmax for the last one, is trained on this data for 500 iterations with batch-size 32, employing cross-entropy loss and an SGD [87] optimizer with learning rate 0.01 and a momentum of 0.9. In addition to the unaugmented baseline models, XAI-based feature augmentation is performed as follows:

Intermediate features are weighted during the forward pass. This is similar to [45], although there are some differences, e.g., in order to keep the experiment simple, we compute attributions w.r.t. the true class. Given samples  $x_i, i \in \{1, \dots, N\}$  in a data batch of size  $N$ , a model  $f_{\theta^t}$  with parameters  $\theta^t$ , and attributions  $r_i^{l,t}$  w.r.t. intermediate features  $f_{\theta^t}^l(x_i)$  at the input of layer  $l$  and iteration  $t \in \{1, \dots, T\}$  ( $T = 500$  in this experiment), we first normalize them as  $(r'_i)^{l,t} = \frac{r_i^{l,t}}{\max(|r_i^{l,t}|)}$ , and then obtain a feature-wise and sample-wise attention mask as

$$M_{\text{feat}}^{i,l,t} = 0.5 + \frac{(r'_i)^{l,t} + 1}{2}, \quad (22)$$

where  $M_{\text{feat}}^{i,l,t} \in [0.5, 1.5]$ . The reweighted features are then obtained as  $f_{\theta^t}^l(x_i)' = M_{\text{feat}}^{i,l,t} \odot f_{\theta^t}^l(x_i)$ , with  $\odot$  denoting the element-wise product. In this experiment, we set layer  $l = 1$  in order to make use of an internal representation, as opposed to the input, but at the same time affect a large amount of successive layers.

Each attribution is normalized by dividing it by its maximum absolute value. The input dimension-wise attribution values over iterations are depicted in Figure 1A (*right*). To reduce visual noisiness, the mean between iterations 0 and  $t$  is depicted at iteration  $t$  for each feature.

### Toy Experiment 2 (Model Performance)

We again generate a dataset with 400 samples (350 train samples, 50 test samples) describing a binary classification problem, similar to the one used in the Toy Experiment 1. Here, however, each sample only contains four dimensions, with dimensions 0–2 being (truly) informative, and dimension 3 indicating the correct class via its sign for the training samples, but being randomized for the test samples. The dataset used in this toy example is visualized in Figure 2. Here, we also only assign 5% of the class labels randomly, since we are using a smaller model than in the above experiment. More concisely, this architecture only consists of two fully-connected layers with five and two neurons, as well as ReLU and softmax activations, respectively. All models are trained for 200 iterations with batch-size 50, again using a cross-entropy loss function and SGD [87] optimizer with learning rate 0.01 and a momentum of 0.9 each.

A generalizing model would not only rely on the distractor dimension 3 for its predictions, but instead leverage information from all informative input dimensions — i.e., all four dimensions for this experiment. In order to improve upon this generalization, we propose an explanation-guided dropout method, similar to [46], which temporarily turns off the intermediate features that the model uses most to make its predictions: For this technique, we first compute attributions  $r_i^{l,t}$ , similar to the previous experiment. The absolute value of these is then taken in order to measure feature *importance* and is then normalized, so that  $(r'_i)^{l,t} = \frac{|r_i^{l,t}|}{\max(|r_i^{l,t}|)}$ . At iteration  $t$ , the  $p\%$  neurons of layer  $l - 1$  that correspond to the largest values of  $(r'_i)^{l,t}$  are dropped out, while the activations of all other neurons in layer  $l - 1$  are rescaled in order to preserve the total average activation.

To measure the effect of explanation-guided dropout on overfitting, we compare unaugmented baseline models against models that employ random (i.e., the standard) dropout and models that employ explanation-guided dropout instead. We choose a dropout rate of  $p = 25\%$ , since one fourth of input dimensions are distractors, and layer  $l = 1$ .

To obtain the graph in Figure 1B (*right*), the attribution of each sample is first normalized by its respective maximum absolute value, in order to portray the relative importance of each input dimension. Especially in the graph on the *right*, this becomes apparent, although the relative relevance of the

explanation-guided dropout models still grows over iterations, but much slower than for the other models, confirming the insights gained from the loss graphs.

### Toy Experiment 3 (Model Reasoning)

For this experiment, we first generate a dataset with 400 samples (200 in train and test set each) with two features, describing a binary classification problem. The data is visualized in Figures 3 and 1C (*middle*). In Figure 1C (*middle*), the training set consists of the pastel colored points, and the test set is shown in saturated colors. While the training set varies in the direction of feature 1, the test set does not. On this data, we train a model consisting of a single fully connected layer with two neurons and a softmax activation function, for 200 iterations with batch-size 50. We employ a crossentropy loss function to measure the prediction error, and an SGD [87] optimizer with learning rate 0.001 and momentum 0.9.

Given samples  $x_i$  and labels  $y_i$ , we first compute corresponding attributions  $r_i^{l,t}$ , and normalize them as  $(r'_i)^{l,t'} = \frac{|r_i^{l,t}|}{\max(|r_i^{l,t}|)}$ . we augment the loss function similar to [48] as

$$\mathcal{L}_{\text{loss-aug}}^{l,t}(x_i) = \mathcal{L}_{\text{pred}}(f_{\theta^t}(x_i), y_i) + \mathcal{L}_{\text{reason}}(r_i, r_A), \quad (23)$$

$$\text{with } \mathcal{L}_{\text{reason}}(r_i, r_A) = (\|(1 - r_A) \odot (r'_i)^{l,t'}\|_2)^2, \quad (24)$$

where  $\mathcal{L}_{\text{pred}}$  is the standard classification loss (here, cross-entropy),  $\|\cdot\|_1$  the  $\ell_1$ -norm, and  $r_A$  a binary ground truth mask (equal to 1 for important input dimensions).  $r_A$  is turned into the corresponding *irrelevancy* mask (which is employed, e.g., by the authors of [48]), by subtracting it from 1. In contrast to [48], due to the simplicity of the setting, we do not consider ground truth mask on a *per-sample* basis, but the same one for the whole dataset instead. Through the regularization term, the model is rewarded for aligning its explanations with the ground truth explanations. For this experiment, we computed input explanations before layer  $l = 0$  and chose  $r_A = (1, 0)^\top$  to focus on input dimension 0 and ignore dimension 1.

In Figure 1C (*right*), the absolute value of each attribution was first taken, and each attribution was then normalized by its respective maximum value, in order to compare the relative importance of each input dimension.

## B Details about Demonstrative Examples

### B.1 Example 1 (Model Performance)

These experiments extended upon [45]. Refer to this work for further details w.r.t. models, data, and implementation.

### B.2 Example 2 (Model Equality)

For these experiments, we trained the pretrained VGG-16 [39] (with batch normalization) and ResNet-50 [119] models available from the PyTorch [121] model zoo, on the Adience and Pascal-VOC datasets. Adience does not contain a dedicated train set, but five data folds instead, however, for our experiments we used fold 0 for testing, and folds 1–4 as a train set. For training, we utilized a crossentropy loss and SGD optimizer with a momentum of 0.9. We chose a learning rate of 0.05 for training on Pascal-VOC, while for Adience we used learning rates 0.1 (VGG-16) and 0.01 (ResNet-50). With Adience, we employed a learning rate decay by multiplying the learning rate with a factor of 0.3 every 5000 iterations, and a learning rate warmup over the first 1000 iterations. For both datasets, after each sample was first normalized using the ImageNet [70] mean and standard deviation, and resized to 256 x 256 pixels. After choosing a random crop of 227 x 227 pixel size and mirroring it randomly for each sample, models were trained using a batch-size of 32 for a certain number of *mini-epochs*. Where an epoch normally includes all samples in the training set, a mini-epoch only contains a subset of training samples, in order to compute current attributions and employ XAI-based changes with a controllable frequency. In our experiments, we trained 60 mini-epoch of 10000 samples each for Adience and 30 mini-epochs of 10000 samples each for Pascal-VOC.

Class-wise performance on the Adience dataset was evaluated using class-wise accuracy on a test set containing four corner-crops and one center-crop and their vertically mirrored versions (refer to [122] for this oversampling scheme). The predicted probabilities of all 10 crops per test sample were averaged to yield the final predicted probabilities of the sample. On Pascal-VOC, class-wise performances were evaluated by measuring the Average Precision (AP) per class. We evaluate the imbalanced performance of a model by computing the mean performance  $\mu_p$  over classes and seeds (3 randomly chosen seeds were evaluated for each experiment), as well as the mean standard deviation  $\sigma_p$  of class-wise performance over seeds. Here, a high  $\mu_p$  while maintaining a low  $\sigma_p$  indicates a more balanced class-wise performance. A balance score can be computed from these as  $b_p = \frac{\mu_p}{\sigma_p}$ . We furthermore investigate the mean predicted probabilities of the respective true class over all test set samples, and how their class-wise distributions change through the applied augmentations.

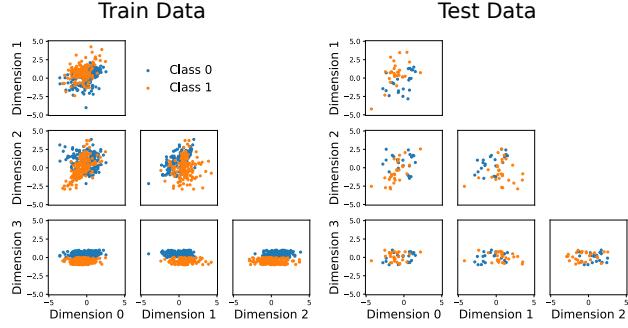
For computing attributions during training, we chose 20 representative samples per class randomly from the training set (160 total for Adience, 400 total for Pascal-VOC). These stayed constant over training. After each mini-epoch, we computed explanations from several local modified backpropagation techniques (since sampling-based techniques take too long to compute after each mini-epoch) for these representative samples using the Zennit library, i.e., (absolute) Sensitivity [19], (absolute) Guided Backpropagation [112], and LRP [20] with the following three composites: (1) The LRP- $z^+$  rule for all layers (called LRP- $z^+$  from here on), (2) the LRP- $z^+$  rule for all convolutional layers and the LRP- $\varepsilon$  rule for fully-connected layers (called LRP- $\varepsilon-z^+$  from here on), and (3) the LRP- $z^B$  rule for the first layer, the LRP- $\gamma$  rule for all convolutional layers, and the LRP- $\varepsilon$  rule for fully-connected layers (called LRP- $\varepsilon-\gamma-z^B$  from here on). Refer to [28, 123] regarding details on the mentioned LRP-rules. After computation, all attributions were normalized by their maximum absolute value. We did not consider attribution metrics based on sampling or surrogate models due to their low efficiency in terms of computation time, and the requirement of the employed method to compute attributions periodically.

During training, we applied the following augmentations with the aim of achieving a more balanced performance between classes: We re-distributed the samples for each mini-epoch, i.e., samples for one mini-epoch were chosen (randomly) from all available samples, such that a portion  $p_c$  of samples belong to class  $c$ . (I) Apart from an unaugmented baseline, (II)  $p_c$  was computed by applying a softmax to the (inverted) class distribution of the whole training set. (III) Furthermore, at each mini-epoch either the entropy (see [51] for details) of the attributions, or the MSE-Distance to the corresponding attributions from the previous mini-epoch (see [51] for details) was computed for each representative sample, and then averaged class-wise. To avoid large variations, attributions were averaged over the last 5 mini-epochs before computing above measurements on them. Afterwards,  $p_c$  was computed by applying softmax to these class-wise averaged measurements. Since we augmented the input data distribution, we employed input-space explanations here.

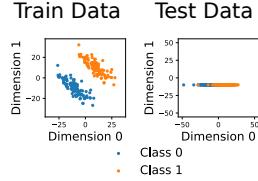
## C Supplementary Toy Experiment Figures



Supplementary Figure 1: Visualization of the dataset used in Toy Experiment 1. The vertical axis of each panel shows the data dimension corresponding to the row index of panels, the horizontal axis shows the data dimension corresponding to column index of the grid. Training data can be found to the *left*, test data to the *right*.



Supplementary Figure 2: Visualization of the dataset used in Toy Experiment 2. The vertical axis of each panel shows the data dimension corresponding to the row index of panels, the horizontal axis shows the data dimension corresponding to column index of the grid. Training data can be found to the *left*, test data to the *right*.



Supplementary Figure 3: Visualization of the dataset used in Toy Experiment 3. The vertical axis of each panel shows the data dimension corresponding to the row index of panels, the horizontal axis shows the data dimension corresponding to column index of the grid. Training data can be found to the *left*, test data to the *right*.