Bounded logit attention: Learning to explain image classifiers

Thomas Baumhauer **Djordje Slijepcevic**[*] **Matthias Zeppelzauer**
St. Pölten University of Applied Sciences
St. Pölten, Austria
{firstname.lastname}@fhstp.ac.at

Abstract

Explainable artificial intelligence attempts to elucidate the workings of systems too complex to be directly accessible to human cognition through suitable side-information referred to as "explanations". We present a trainable explanation module for convolutional image classifiers we call bounded logit attention (BLA). The BLA module learns to select a subset of the convolutional feature map for each input instance, which then serves as an explanation for the classifier's prediction. BLA overcomes several limitations of the instance-wise feature selection method L2X: 1) BLA scales to real-world sized image classification problems, and 2) BLA offers a canonical way to learn explanations of variable size. Due to its modularity BLA lends itself to transfer learning setups and can also be employed in a post-hoc fashion. Beyond explainability, BLA may serve as a general purpose method for differentiable approximation of subset selection. In a user study we find that BLA explanations are preferred over L2X and (Grad-)CAM explanations.

1 Introduction

A commonly quoted rule of thumb is that as machine learning systems increase in size and sophistication, it becomes increasingly hard to understand how they arrive at their predictions. Such uninterpretable "blackbox" systems are undesirable both from a usability and engineering perspective.

In this work, we deal with explanations of predictions made by image classification models based on deep convolutional neural networks (CNNs). Much work in this area is focused on generating explanations *post-hoc*. This means that after building and training a model, an additional mechanism is used to produce explanations for the model's behavior. An explanation could be any additional human-interpretable information suitable to improve the understanding of the workings of (some aspects of) a model. In the field of image classification such information could be visual, e.g. highlighting or masking certain parts of an input image.

Considerably less popular than post-hoc methods are trainable explanation mechanisms incorporated directly into a model. A drawback of this approach is that it cannot a priori be ruled out that such mechanisms could negatively affect the model's performance. However, building explainability mechanisms directly into a model as part of its architecture is appealing insofar as once trained such explanations provide by construction true, accurate insight into the workings of (individual aspects of) the model. This is in contrast to post-hoc explanations, where one always has to question how faithful these explanations are to actual calculations performed by the model (Hooker et al., 2019).

Contributions. We propose *bounded logit attention* (BLA), a trainable, modular explanation mechanism to be incorporated into convolutional image classification networks. The BLA module learns to select a subset of the convolutional feature map for each input instance, which then serves as an

[*] Corresponding author.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

explanation for the classifier's prediction. Relative to previous work on learned explanations by Chen et al. (2018) our key contributions are: **1)** BLA scales to real-world sized CNNs. **2)** BLA offers a canonical way to produce explanations of variable size. One of the merits of our method is that due to its modularity it can be used with typical transfer learning setups in computer vision, in particular pretrained feature extractors. Similarly, our method may even be employed as a purely post-hoc method. Beyond explainability, BLA can be used as a general purpose method for differentiable approximation of subset selection. In our experiments we obtain favorable results according to both quantitative metrics and human evaluation.

2 Related work

There is no consensus on the definitions of notions such as *interpretability, explainability*, etc. and we use them loosely throughout this work. Roughly by *interpretable* we mean "accessible to human understanding" and and by *explanation* we mean "any piece of information that facilitates interpretability". Lipton (2018) investigates notions related to interpretability in a principled way. In his taxonomy, the opposite of "blackbox-ness" is *transparency*. In this sense, a built-in explanation mechanism like ours makes a model (partially) transparent.

Explanations in computer vision. Linear models are easy to interpret, as long as they use a relatively small number of (interpretable) features. As a consequence, many approaches to interpretability involve the construction of a linear surrogate model of some sort. End-to-end differentiable models can be linearized around an input. In the field of computer vision this approach is known as *saliency maps* (Simonyan et al., 2013) which present gradient information graphically. Since the gradients of large image classification networks are noisy, averaging gradients over some neighborhood has been proposed by Sundararajan et al. (2017); Smilkov et al. (2017).

Another idea is decomposing input images into coarser, more interpretable features, such as superpixels (cohesive segments of similar pixels). Then, a neighborhood of this input is defined, consisting of all images with any of the interpretable features either "present" or "absent" (corresponding to occluding the "absent" superpixels with black color). LIME (Ribeiro et al., 2016) builds a local approximation of the model for this neighborhood. Similarly, one may use the Shapley value (Shapley, 1953) of interpretable features when constructing explanations. It turns out that Shapley values also fit into the LIME framework (Charnes et al., 1988) as pointed out by Lundberg and Lee (2017) (there called SHAP). Other perturbation-based methods, e.g. introduced by Fong and Vedaldi (2017); Qi et al. (2019), estimate the importance of input features by (partially) occluding the input and measuring the effect on the model output without employing a additional learning component.

Zhou et al. (2016) propose *class activation mappings* (CAM) which assigns saliency to convolutional feature maps based on the coefficients of a final dense linear layer. Selvaraju et al. (2017) generalize this approach to Grad-CAM where in the case that more than one layer is used to process the convolutional features they use a gradient-based linear approximation of this calculation. Example-based methods (Koh and Liang, 2017; Chen et al., 2019; Hase et al., 2019) provide explanations that relate predictions to the training data. User studies evaluating some of the explanation methods described in this section were conducted by Hase and Bansal (2020); Jeyakumar et al. (2020).

Learning to explain. Our work is inspired by the *learning to explain* (L2X) method by Chen et al. (2018), proposing "instancewise feature selection as a methodology for model interpretation". Given a d-dimensional input x first a binary mask $\delta \in 2^d$ is computed in an explanation network (e.g. a multilayer perceptron). Then, the masked input $x \odot \delta$ is used as the input to a second network solving the task. In order to make this setup end-to-end differentiable the computation of the discrete mask δ has to be approximated in a continuous fashion. To this end, the explanation network learns a distribution of inclusion probabilities of each input feature. Then, k features are drawn independently from this distribution, to approximate (coarsely) the sampling of a mask of size k (i.e. a mask with exactly k of the d entries of δ equal to 1). The discrete sampling itself is approximated by the *Gumbel-softmax trick* (Jang et al., 2016; Maddison et al., 2016). At test time no sampling is performed and instead the k features with highest inclusion probabilities are used. As a result, L2X is constrained to produce explanations of fixed size k, with k a hyperparameter. In this work, we propose an alternative way to compute masks that permits masks of different sizes for different inputs. This is crucial for vision tasks as the sizes of regions of interest in an image vary between images.

For image classification Chen et al. (2018) compute the mask δ through a convolutional network. As a consequence of pooling operations in the network, δ is of lower resolution than the input x and hence must be upsampled (with repetition) to $\tilde{\delta}$ before computing the masked input $x \odot \tilde{\delta}$. The masked input image is then classified by a second convolutional network. We find that this approach scales poorly to real-life sized image classification problems (Section 4.1). Furthermore, Jethani et al. (2021) identify that L2X can lead to explanations with limited fidelity in the case a prediction is encoded in only one image patch. They propose REAL-X, a method similar to L2X, except that the first convolutional network is forced to learn meaningful masks that are not subject to the aforementioned encoding. These approaches are computationally expensive, requiring two forward passes through convolution stacks. Our proposed architecture is designed to alleviate these issues. For further related work and the embedding of our approach into related approaches see Section 3.3.

3 Bounded logit attention

Image classification. We consider a standard supervised image classification task for some dataset consisting of pairs (x, y), with x an image annotated by some class-label y. A model h for this tasks predicts labels $\hat{y} = h(x)$. We assume the following standard architecture for h. Given an input image x a convolutional feature extractor $\mathcal{F}$ computes a list of feature vectors $\mathcal{F}(x) = \vec{f} = \langle f_i : i < n \rangle$, with n the size of the feature map. To avoid clutter, we use a single index $i < n$ for these features instead of the usual two indices indicating a feature's position in the two-dimensional feature map, but of course each feature still corresponds to a region in x. These features are then globally average-pooled to a feature vector

$$v = \frac{1}{n} \sum_{i<n} f_i.$$

Finally, a classifier $\mathcal{L}$ (e.g. a logistic regression head, or perhaps a multilayer perceptron) computes $\hat{y} = \mathcal{L}(v)$.

Explanation module. The features f_i learned by the model h may be considered an abstract representation of properties of the corresponding regions of the input image x. In this work, we are interested in selecting features that are "important" to the model h when predicting the label $\hat{y} = h(x)$. To this end, we propose to employ an explanation module $\mathcal{E}$ that outputs a subset of feature indices $\mathcal{E}(\vec{f}) = \vec{\delta} = \langle \delta_i : i < n \rangle \in 2^n$. A model h, as described above, may then be augmented by $\mathcal{E}$ by replacing the average-pooling operation by the reweighed pooling operation

$$v = \frac{1}{\|\vec{\delta}\|_1} \sum_{i<n} \delta_i f_i.$$

Then $\vec{\delta} = \mathcal{E}(\vec{f})$ acts as an explanation for the prediction $\hat{y} = \mathcal{L}(v)$. Indeed $\vec{\delta}$ is easily human-interpretable as each entry can be understood as a binary flag encoding the use of the corresponding feature in the prediction, and the size of $\vec{\delta}$ is typically small enough, e.g. $n = 49 = 7^2$. Our goal is to build the explanation module $\mathcal{E}$ in a way such that the model remains end-to-end differentiable, i.e. such that $\mathcal{E}$ may be learned.

3.1 Parameterizing the explanation module

Variational approximation. We identify $\vec{\delta} = \mathcal{E}(\vec{f})$ with the probability distribution $p(i|\vec{f}) = \frac{\delta_i}{\|\vec{\delta}\|_1}$. During training we approximate $p(\cdot|\vec{f})$ by some $q(\cdot|\vec{f}) = \mathcal{Q}(\vec{f})$ where $\mathcal{Q}$ is an element of some variational family

$$V = \{\mathcal{Q}_\phi \mid \mathcal{Q}_\phi : \vec{f} \mapsto q(\cdot|\vec{f}), \phi \in \Phi\}$$

parameterized by ϕ. Hence, we need to find a suitable variational family, which in deep learning terms means finding the appropriate neural architecture implementing $\mathcal{Q}$. We require $\mathcal{Q}$ to compute a probability distribution q approximating p.

Activation function. The key trick we propose is using the simple, yet uncommon activation function

$$\beta(x) = \min(x, 0)$$

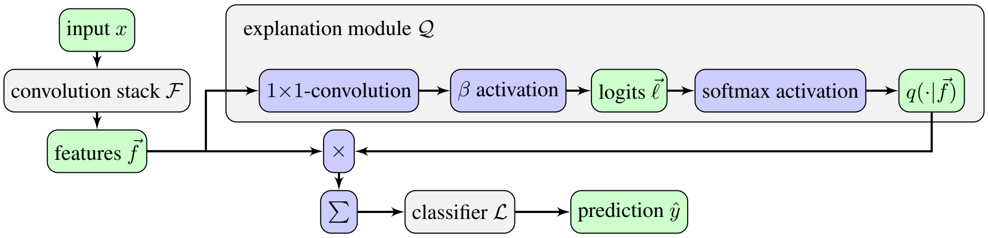


Figure 1: Image classifier augmented with a bounded logit attention explanation module.

in the approximation of the logits of such discrete distributions. The β-activation function is closely related to the ReLU function. What is uncommon is applying it directly before the softmax function.

Concretely, we build $\mathcal{Q}$ as follows (Figure 1). Given a feature map $\vec{f}$, we apply a 1×1-convolution with a single filter, mapping each f_i to a scalar value $g_i = u^\top f_i$, with u the weights of the convolution.[2] Then, we compute the logits $\vec{\ell} = \langle \ell_i : i < n \rangle$ where $\ell_i = \beta(g_i)$. Finally, we apply the softmax function σ to obtain $\vec{q} = \sigma(\theta \cdot \vec{\ell})$, with the temperature θ being a hyperparameter. During training we plug $\mathcal{Q}$ into the original model h by replacing the pooling operation with

$$v = \sum_{i<n} q(i|\vec{f}) f_i. \tag{1}$$

Discretization. After training we obtain the explanation module $\mathcal{E}$ from its variational approximation $\mathcal{Q}$ by defining $\delta_i = 1$ iff $\ell_i = 0$. The merit of the β-function now becomes clear: it allows for constructing explanations of variable size (the size of an explanation being $\|\vec{\delta}\|_1$) by giving us a canonical way of discretization. Importantly, it forces q to be uniform on indices for which $\ell_i = 0$, making the behavior of $\mathcal{Q}$ and its discretization $\mathcal{E}$ match more closely.

Thresholding. Why may we hope that $\mathcal{Q}$ might learn to approximate reasonable explanations? Remember that we are interested in finding “important” features. The explanation module $\mathcal{Q}$ computes weights $q(i|\vec{f})$ for each feature f_i to use in the pooling operation in formula (1). Thus during training $\mathcal{Q}$ should learn to assign more weight to features that are discriminative for the task, and to mostly ignore features that are not. To encourage this behavior and put additional learning pressure on $\mathcal{Q}$ we propose to threshold weights in a modified pooling operation

$$v = \sum_{i<n} \mathbb{1}[q(i|\vec{f}) > \gamma] f_i \tag{2}$$

where $\mathbb{1}[q(i|\vec{f}) > \gamma]$ is 1 if $q(i|\vec{f}) > \gamma$ and 0 otherwise. We propose $\gamma = \frac{1}{n}$ as the default threshold.

3.2 Application-specific aspects

Bounded logit attention (BLA). One way to think about the explanation module described above is to consider it an *attention mechanism*, in the sense that $\mathcal{Q}$ computes soft attention, approximating hard attention in $\mathcal{E}$. With the β-activation function bounding the logits of $q(\cdot|\vec{f})$ from above being the key property of this attention mechanism, we refer to our architecture as *bounded logit attention* (BLA). Independently of explainability, BLA can serve as a general purpose method to produce differentiable approximations for the selection of subsets.

Transfer learning. Note that the modularity of our method lends itself to transfer learning setups. Having trained a model $h = \mathcal{F} \circ \mathcal{L}$, we may for example hold both $\mathcal{F}, \mathcal{L}$ fixed and only train an explanation module $\mathcal{Q}$ as a post-hoc addition to h. We investigate this approach in Section 4.2 under the name BLA-PH. In this setup we could even make predictions using the original model h (without $\mathcal{Q}$ plugged in) and use the outputs of $\mathcal{Q}$ as a purely post-hoc explanation. Section 4.3 provides justification for this approach (that has the advantage that the outputs of $\mathcal{Q}$ are far cheaper to compute than the LIME scores used there). Finally, our method can readily be used with a pretrained feature extractor $\mathcal{F}$ while training both $\mathcal{L}$, $\mathcal{Q}$, an approach we take throughout Section 4.2.

[2] More complex mappings $f_i \mapsto g_i$ can be considered, however we found a 1×1-convolution to be sufficient.

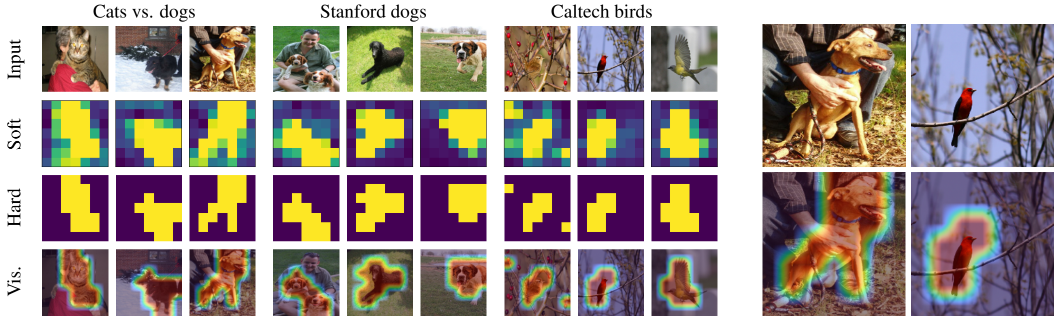


Figure 2: Left: Learned explanations by BLA on three datasets from the experiment in Section 4.2. Rows, from top to bottom: input image; soft explanation $q(i|\vec{f})$ computed by $\mathcal{Q}$; discretized hard test-time explanation $\vec{\delta}$ by $\mathcal{E}$; visualization of $\vec{\delta}$ as described in Section 3.2. Right: enlarged version.

Visualization. Like the (Grad-)CAM method (Zhou et al., 2016; Selvaraju et al., 2017), BLA produces explanations at the level of the convolutional feature map. In Section 1 we mentioned that a conceptual advantage of the learned explanations produced by $\mathcal{E}$ is that they faithfully reflect the workings of (some aspect of) the model by construction. However, we still need to make a choice how to present the outputs $\vec{\delta}$ of $\mathcal{E}$. Following the convention of the (Grad-)CAM literature, we visualize $\vec{\delta}$ using a colormap, which is upscaled and imposed on the input image (Figure 2). The experiment in Section 4.3 provides justification for our choice. Note that other choices are possible of course, e.g. to spread out each δ_i over the corresponding receptive field.

3.3 Alternative explanation modules

In Section 3.1 we proposed BLA as a possible architecture of the explanation module $\mathcal{Q}$. There exist several concepts in the literature one might consider for alternative explanation modules.

Fixed size explanations: L2X-F. When employing the transfer learning approach discussed above, keeping the feature extractor $\mathcal{F}$ fixed, and identifying each input x with its feature map $\vec{f} = \mathcal{F}(x)$, our approach resembles computing L2X (see Section 2) explanations for the classification head $\mathcal{L}$. The key differences are: L2X a priori fixes the size of the generated explanation to some $k < n$, while in our approach use of the β-function allows to generate explanations of a variable size. Like BLA, L2X computes a probability distribution over the input points (which here are the convolutional feature vectors), but unlike BLA the training of L2X has a non-deterministic element in the approximate sampling from this distribution. We refer to L2X at the feature level as *L2X-F* and use it as a baseline to compare our approach to.

Another way to think about the BLA explanation module is that $q(\cdot|f_i)$ computes an *objectness score* for the feature f_i. However, the prediction of the objectness of f_i is not made independently of the objectness of $f_j, j \neq i$. As a result of the softmax function there is global competition between these scores, in the sense that increasing $q(\cdot|f_i)$ necessarily means decreasing $\sum_{j \neq i} q(\cdot|f_j)$. Thus the softmax function provides a weak form of global context in the explanation module. This raises the question, which amount of context is most suitable.

More context: attention with global concept vector. Jetley et al. (2018) propose “an end-to-end differentiable attention module for CNN architectures built for image classification”. Like in our approach, the authors compute probability distributions to weight features (and they do this for three different levels of convolutional features, not just the outputs of the feature extractor as we do). The key differences to our approach is that a global concept vector g, computed as a learned linear transformation of the concatenation of all feature vectors is used in the computation of the logits of the probability distribution. To this end, g is added to each feature vector f_i, resulting in logits $\ell_i = u^\top(g + f_i)$. We attempted incorporating a global concept vector in the BLA module in this way (while keeping the β-activation for the logits before the softmax). Interestingly, the module learned to compute extremely negative logits (of magnitude ca. -10^3), essentially circumventing the β-activation function (Figure S19 in the supplemental material). As a result, only a single feature is chosen in the discrete attention module $\mathcal{E}$ (since there is now a unique maximum of the logits ℓ_i). We believe this is due to the concept vector enabling the explanation module to “cheat” in the sense that it can be very certain which feature vector it wants to choose. Since the initial results were not promising, we did not pursue this approach further.

Less context: pointwise attention. Park et al. (2018) and Woo et al. (2018) propose an attention module for CNNs they call CBAM. They do not use the softmax function when computing their attention maps, but instead use pointwise sigmoid activations, i.e. they use no global context at all. We found that for our purposes the lack of competition between features did not seem to put sufficient learning pressure on an explanation module to choose between features, resulting in poorly focused attention maps. Again, we did not pursue this approach further.

4 Experiments

4.1 Comparison to L2X

In this section we substantiate the claims that 1) BLA improves accuracy over L2X on small datasets and 2) unlike BLA, L2X scales poorly to larger datasets, hence a new method is indeed needed.

The MNIST dataset (LeCun et al., 2010) contains 28×28 grayscale images of handwritten digits. We subsample all images of 3 and 8, resulting in 11982 training and 1984 validation images. Chen et al. (2018) report validation accuracy of 0.958 (using hard explanations, there called post-hoc accuracy) on this dataset for their L2X method, with fixed explanation size $k = 4$. For their baseline model without learned explanations they report a validation accuracy of 0.997. We imitate their CNN architecture, building a model consisting of three 5×5 convolutions with ReLU activations with 8, 16, and 16 filters, respectively. The first two convolutions are followed by 2×2 maximum pooling, and the third convolution is followed by a dense linear layer with a sigmoid output unit. We augment this CNN by a BLA explanation module as described in Section 3.1, and train the model end-to-end for 3 epochs, using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 10^{-3} to minimize the cross-entropy loss. We use temperature $\theta = 0.1$ and thresholding with $\gamma = 49^{-1}$. Over 20 runs we obtain a mean baseline accuracy of $0.994\pm1.53\text{e-}3$, and a mean accuracy of $0.993\pm3.86\text{e-}4$ for the interpretable model, with no statistically significant difference (p=0.26 Mann-Whitney-U-test).

The cats vs. dogs (*CvsD*) dataset (Elson et al., 2007) consists of 23,262 color images, divided equally into two classes “cats” and “dogs”. We resize all images to 224×224. We use 18,609 images for training and 4653 images for validation. We adapt the setup of Chen et al. (2018) to the CvsD dataset. The explainer network uses an EfficientNet-B0 (Tan and Le, 2019) convolution stack (pretrained on ImageNet and frozen in the subsequent experiments), producing explanations of size $k = 8$ (8 selected 32×32 patches of the 224×224 input image), while the classification network uses the same convolution stack, followed by a dense layer with sigmoid activation. We train this setup for 3 epochs using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 10^{-3}. L2X obtains a validation accuracy of 0.96 using soft explanations, but a validation accuracy of 0.5 when using the desired hard explanations (i.e. the random baseline). According to our visual assessment, the produced explanations indeed failed to capture meaningful information. We take this failure as evidence that L2X is difficult to scale to deep CNNs (while also being less efficient than BLA).

4.2 Understanding the BLA explanation module

In this section we show that BLA improves accuracy and loss over the L2X-F baseline, implying that the improvements of BLA over L2X are not solely due to the switch to a single pass convolution architecture. We proceed to show that thresholding slightly increases accuracy further, while increasing loss. Finally, we show that employing BLA-T post-hoc does not decrease accuracy on average.

Datasets Additionally, we evaluate our method on the Stanford Dogs (*StanDogs*) dataset (Khosla et al., 2011; Deng et al., 2009), consisting of 20,580 color images of 120 dog breeds. We use 12,000 images for training and 8,580 for validation. Finally, the Caltech-UCSD Birds-200-2011 (*CUB-200*) dataset (Welinder et al., 2010), consists of 11,788 color images of 200 bird species. We use 5,994 images for training and 5,794 for validation. For each dataset images are resized to 224×224.

We use EfficientNet-B0 (Tan and Le, 2019) as our classification model $h = \mathcal{F} \circ \mathcal{L}$, i.e. $\mathcal{F}$ is the EfficientNet-B0 convolution stack and $\mathcal{L}$ consists of a dropout layer (Srivastava et al., 2014), dropping units with probability 0.2, followed by a dense linear layer with softmax activations. The parameters of $\mathcal{F}$ are pretrained on ImageNet and held fixed in our experiments. Throughout this section we

Table 1: Accuracy and loss for the experiments in Section 4.2. The symbol † denotes no statistically significant difference to the baseline (BL) column. The symbol $*$ denotes the same for the column to the left. The value after the ± symbol is the standard error multiplied by 100.

	Dataset	BL	L2X-F	BLA	BLA-T	BLA-PH
Accuracy	CvsD	0.9912 ± 0.01	0.9767 ± 0.12	0.9902 ± 0.02	$0.9905^{*} \pm 0.01$	$0.9902^{*} \pm 0.02$
	StanDogs	0.8357 ± 0.03	0.7730 ± 0.24	0.8234 ± 0.07	0.8269 ± 0.03	$0.8265^{*} \pm 0.07$
	CUB-200	0.6794 ± 0.07	0.6451 ± 0.12	0.6722 ± 0.09	$0.6801^{\dagger} \pm 0.06$	$0.6650^{\dagger *} \pm 0.62$
Loss	CvsD	0.0237 ± 0.03	0.0800 ± 0.33	0.0349 ± 0.09	0.0742 ± 0.15	0.0562 ± 0.40
	StanDogs	0.5313 ± 0.08	1.4042 ± 0.76	0.8357 ± 0.33	1.1319 ± 0.29	$1.0649^{*} \pm 2.53$
	CUB-200	1.2988 ± 0.09	2.3237 ± 1.37	1.7809 ± 0.59	2.1510 ± 1.09	$2.1714^{*} \pm 1.63$

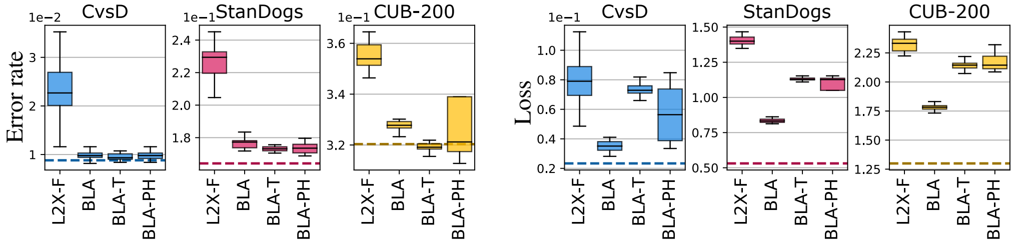


Figure 3: Graphical representation of Table 1. Left: error rate ($= 1-$accuracy), right loss (lower is better for either). The dashed line represents the median of the uninterpretable baseline models. We observe improvements of both error rate and loss for BLA (ours) over L2X-F.

use the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 10^{-3}, a batch size of 32, minimizing the cross-entropy loss. Non-pretrained parameters are set using the Glorot uniform initalizer (Glorot and Bengio, 2010).

Baseline (BL). We begin by training classification heads $\mathcal{L}$ for each dataset (training for 2 epochs for CvsD and 5 epochs for StanDogs, CUB-200), to obtain trained classification models $h = \mathcal{F} \circ \mathcal{L}$. These models will serve as the uninterpretable baseline. Note that our goal is not to reach state-of-the-art performance, but to investigate how the addition of explainability modules affects the performance, relative to the baseline. We conduct every experiment in this section 20 times for each dataset, e.g. we begin by training 20 baseline models for each dataset. To compare the results we use the Wilcoxon signed-rank test, setting the level of significance to 0.05. When talking about measurements (e.g. accuracy or loss) we always refer to the mean over all 20 runs. When comparing mean measurements, their differences are always implied to be statistically significant, unless stated otherwise.

In a next step, we augment our baseline models $h = \mathcal{F} \circ \mathcal{L}$ with explanation modules $\mathcal{Q}$. We train the resulting models for 2 epochs for dataset CvsD and 5 epochs for StanDogs, CUB-200. The parameters of $\mathcal{F}$ and $\mathcal{L}$ serve as a starting point for training, with $\mathcal{Q}$ randomly initialized as described above. The convolution stack $\mathcal{F}$ continues to be held fixed, while $\mathcal{L}$, $\mathcal{Q}$ are trained, except in the last experiment where we hold the head $\mathcal{L}$ fixed as well. Validation accuracies and losses are always reported for the hard explanations generated by $\mathcal{E}$ (which are approximated during training by soft explanations in $\mathcal{Q}$). The numerical results of the following experiments are listed in Table 1 and presented graphically in Figure 3. Examples of explanations for this experiment are shown in Figure 2 and further examples from experiments in this section are found in the supplemental material (Figure S7-S18).

L2X-F. First, we try adding L2X-F modules to obtain an interpretable baseline as discussed in Section 3.3. Accuracy and loss for all three datasets are considerably worse than the corresponding measurements for the baseline, i.e. adding interpretability through L2X-F modules comes at a considerable cost in model performance.

BLA. Next, we investigate our BLA modules, with a temperature $\theta = 0.1$ and without thresholding. The accuracies improve greatly compared to the L2X-F interpretable baseline, while not quite reaching the uninterpretable baseline. The losses also improve compared to the L2X-F baseline, but do not come quite as close to the baseline losses.

BLA-T. Next, we investigate the BLA modules, again with temperature $\theta = 0.1$ using thresholding with $\gamma = 0.98/49 = 0.02$. Compared to BLA (the same method without thresholding), the accuracies improved slightly. For dataset CvsD the improvement is not statistically significant, for StanDogs the improvement is significant with the accuracy still significantly worse than the uninterpretable

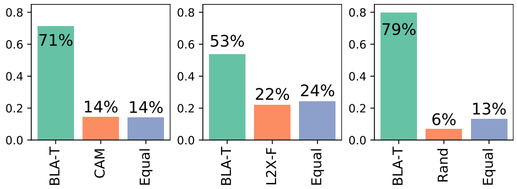


Figure 4: User preference for explanations, from left to right: BLA-T vs. (Grad-)CAM; BLA-T vs. L2X-F; BLA-T vs. a random explanation from the same distribution.

Dataset	LIME/CAM	LIME/BLA-T
CvsD	0.47	0.37
StanDogs	0.33	0.47
CUB-200	0.08	0.37

Table 2: Mean Spearman rank correlation between LIME and CAM, respectively LIME and BLA-T explanations.

baseline, and for CUB-200 the accuracy is no longer significantly different to uninterpretable baseline. Curiously, the losses increased a fair bit compared to BLA, but remained lower than the L2X-F losses.

BLA-PH. Finally, we try the same setup as BLA-T, but with frozen heads $\mathcal{L}$, i.e. only the explanation module $\mathcal{Q}$ being updated during training. Except for the loss for dataset A, all of the measurements are not statistically significantly different from the corresponding measurements for BLA-T. This means models with $\mathcal{L}$ frozen perform on average as well as those where $\mathcal{L}$ is trained together with $\mathcal{Q}$. However, the variances of these measurements increased drastically.

We ran our experiments on NVIDIA GeForce GTX 1080 GPUs. While each of the models for any of the three datasets can be trained in a matter of minutes, due the large number of runs the experiments described above take up about 11 hours of GPU time. Our implementation is available at: `https://github.com/fhstp/bla`

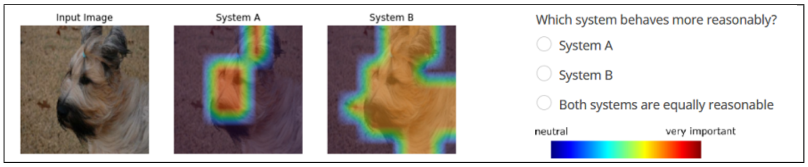
ImageNet. We demonstrate that our method scales to the ImageNet 2010 dataset by Russakovsky et al. (2015), consisting of medium resolution color images of 1000 categories. We train on 1,261,406 images and validate on 50,000, resizing to 224×224. In the fashion described above, we train one baseline model, and models augmented with BLA and BLA-T modules, for 2 epochs each. We obtain an uninterpretable baseline accuracy of 0.469. For BLA we obtain 0.458 and for BLA-T 0.525, improving upon the baseline accuracy. An additional 8 hours of GPU time were used.

4.3 On faithfulness

While learned explanation mechanisms are faithful to the model by construction, getting back to points raised in Section 3.2 we may wonder: a) How faithful are BLA explanations to the uninterpretable baseline model, i.e. is it appropriate to use BLA in a post-hoc setup and b) is our choice of visualization faithful? Inspired by an occlusion experiment in Selvaraju et al. (2017), we use LIME Ribeiro et al. (2016), which was designed for local faithfulness, as baseline. For the uninterpretable baseline model, we compute LIME scores for patches of size 32×32 of the 224×224 input images from the validation set, resulting in explanations of size 7×7, by randomly sampling 10^3 occlusion patterns and fitting a least squares model. We compute the Spearman rank correlation between LIME explanations and the 7×7 soft explanations produced by $\mathcal{Q}$ (using BLA-T). Additionally we compute rank correlation between LIME and (Grad-)CAM (Table 2). We observe that both BLA-T and (Grad-)CAM consistently achieve moderate rank correlation, except on dataset CUB-200 where the rank correlation for (Grad-)CAM is low. We conclude that our method's faithfulness is comparable to that of (Grad-)CAM on datasets CvsD, StanDogs and considerable more faithful on dataset CUB-200.

4.4 User study

In a user study we investigate the following questions: **1)** How do users rate BLA-T explanations compared to explanations generated by the popular (Grad-)CAM method (Zhou et al., 2016; Selvaraju et al., 2017) which also produces explanations at the level of the convolutional feature map and **2)** compared to L2X-F fixed size explanations? **3)** Can users tell the difference between BLA-T and random explanations from the same distribution (a BLA-T explanation from another, random image in the dataset imposed on the original image)? The latter question is intended as a sanity check to verify if our method actually produces meaningful output, as it is easy to fool oneself by wishful thinking when relying on one's own visual assessment (Adebayo et al., 2018).



Which system behaves more reasonably?

○ System A

○ System B

○ Both systems are equally reasonable

Figure 5: A question to users, here L2X-F ("System A", not to be confused with dataset A) vs. BLA-T ("System B"). Instructions were: "Two systems (A and B) were trained to differentiate dog breeds. Both systems decide for the class *briard* in this example. Below, the two different systems show you on which parts of the image they base their decision. Which system behaves more reasonably?"

The participants in this study were 62 unpaid volunteers from diverse educational backgrounds (for further details refer to Figures S3-S6 in the supplemental material). In each question, participants were presented with a random image from one of the three datasets and two explanations, corresponding to one of the three questions above. The position of the explanations was randomized. Participants were asked which of the two explanations, if any, they consider more reasonable. For each question type and dataset, participants answered four questions (Figure 5).

We obtain the following results (Figure 4): **1)** When compared to (Grad-)CAM, BLA-T was chosen 71% of times and (Grad-)CAM 14% of times. We consider such clear user preference for our method over the well established (Grad-)CAM method to be strong evidence that BLA will be able to serve as a useful new tool in the field of explainable artificial intelligence. **2)** When compared to L2X-F, BLA-T was chosen 53% of times and L2X-F 22% of times. The difference between these methods is less decisive than for the previous question, as evidenced by the 24% "equally reasonable" responses, however overall there is still a clear preference for BLA-T. Thus, our method is an improvement over "learning to explain" Chen et al. (2018) on feature level not only conceptually by being able to produce explanations of variable size, but also according to user preference. **3)** Our sanity check is passed easily, with 79% choosing the true BLA-T explanations and only 6% the random explanation. A more detailed evaluation is found in the supplemental material (Figure S2).

5 Conclusion

We presented BLA as a method to learn explanations in deep image classification networks, selecting "important" elements of the convolutional feature map for each instance. The key idea of using the activation function $\beta(x) = \min(x, 0)$ to bound the logits of the attention map allowed us to canonically produce explanations of variable size. Beyond explainability, BLA can also serve a general purpose method to produce differentiable approximations for the selection of subsets.

The main concern with trainable, built-in explainability methods is that they affect model performance. In Section 4.1 we showed that unlike L2X method of Chen et al. (2018) our method does not affect accuracy on a subsample of the MNIST dataset. Our experiment attempting to scale L2X to the cats vs. dogs dataset failed, demonstrating the need for a more scalable approach such as the one presented in this work. In Section 4.2 we showed that L2X-F, i.e. using the L2X method on feature level, also comes at a considerable cost of decreased accuracy. However, using BLA modules the performance came close to that of the uninterpretable baseline models. The accuracies slightly improved further when using thresholding in the BLA module, which for Caltech birds datatset even resulted in accuracy statistically indistinguishable from the baseline accuracy. However, thresholding came at the price of increased loss. A pure transfer learning approach, only training $\mathcal{Q}$ while holding the head $\mathcal{L}$ fixed, resulted in no change in performance on average, but increased variance.

Our quantitative experimental results are thus summarized as follows: built-in explainability for an image classification model using the BLA attention module comes at a very moderate cost of accuracy, while (unlike post-hoc methods) providing by construction faithful insight into the workings of the classifier. Finally, conducting a user study, we found strong evidence that participants prefer our method over (Grad)-CAM and L2X on feature level.

Acknowledgements: This work was funded by the Austrian Research Promotion Agency (FFG, project no. 866880 and 880546) and the Vienna Science and Technology Fund (WWTF, project no. ICT20-016).

References

J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31:9505–9515, 2018.

A. Charnes, B. Golany, M. Keane, and J. Rousseau. Extremal principle solutions of games in characteristic function form: core, chebychev and shapley value generalizations. In *Econometrics of planning and efficiency*, pages 123–133. Springer, 1988.

C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su. This looks like that: deep learning for interpretable image recognition. In *Advances in neural information processing systems*, pages 8930–8941, 2019.

J. Chen, L. Song, M. Wainwright, and M. Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pages 883–892. PMLR, 2018.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

J. Elson, J. J. Douceur, J. Howell, and J. Saul. Asirra: A captcha that exploits interest-aligned manual image categorization. In *Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, Inc., October 2007.

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

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

P. Hase and M. Bansal. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5540–5552, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.491.

P. Hase, C. Chen, O. Li, and C. Rudin. Interpretable image recognition with hierarchical prototypes. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 7, pages 32–40, 2019.

S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim. A benchmark for interpretability methods in deep neural networks. *Advances in neural information processing systems*, 32, 2019.

E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

N. Jethani, M. Sudarshan, Y. Aphinyanaphongs, and R. Ranganath. Have we learned to explain?: How interpretability methods can learn to encode predictions in their interpretations. In *International Conference on Artificial Intelligence and Statistics*, pages 1459–1467. PMLR, 2021.

S. Jetley, N. A. Lord, N. Lee, and P. H. Torr. Learn to pay attention. *arXiv preprint arXiv:1804.02391*, 2018.

J. V. Jeyakumar, J. Noor, Y.-H. Cheng, L. Garcia, and M. Srivastava. How can i explain this to you? an empirical study of deep neural network explanation methods. *Advances in Neural Information Processing Systems*, 33, 2020.

A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

P. W. Koh and P. Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Z. C. Lipton. The mythos of model interpretability. *Queue*, 16(3):31–57, 2018.

S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon. Bam: Bottleneck attention module. *arXiv preprint arXiv:1807.06514*, 2018.

Z. Qi, S. Khorram, and F. Li. Visualizing deep networks by optimizing with integrated gradients. In *CVPR Workshops*, volume 2, 2019.

M. T. Ribeiro, S. Singh, and C. 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*, pages 1135–1144, 2016.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

L. S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958, 2014.

M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

S. Woo, J. Park, J.-Y. Lee, and I. So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.