Convolutional Dynamic Alignment Networks for Interpretable Classifications

Moritz Böhle
MPI for Informatics
Saarland Informatics Campus

Mario Fritz
CISPA Helmholtz Center
for Information Security

Bernt Schiele
MPI for Informatics
Saarland Informatics Campus

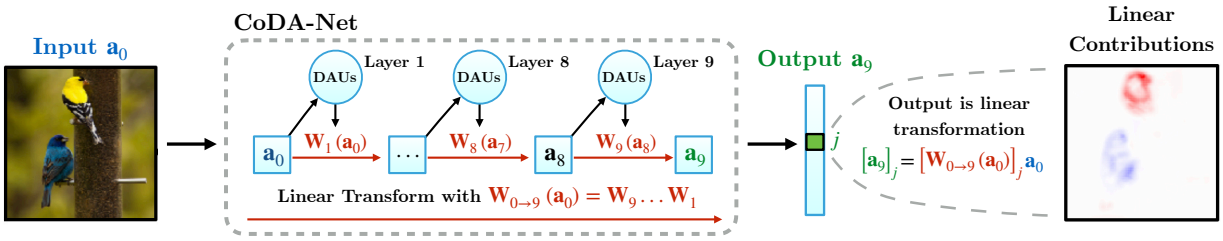

Figure 1: Sketch of a 9-layer CoDA-Net, which computes its **output** $\mathbf{a}_9$ for an **input** $\mathbf{a}_0$ as a linear transform via a matrix $\mathbf{W}_{0\to9}(\mathbf{a}_0)$, such that the output can be linearly decomposed into input contributions (see right). $\mathbf{W}_{0\to9}$ is computed successively via multiple layers of Dynamic Alignment Units (DAUs), which produce matrices $\mathbf{W}_l$ that align with their respective inputs $\mathbf{a}_{l-1}$. As a result, the combined matrix $\mathbf{W}_{0\to9}$ aligns well with taks-relevant patterns. Positive (negative) contributions for the class 'goldfinch' are shown in red (blue).

Abstract

We introduce a new family of neural network models called Convolutional Dynamic Alignment Networks[1] *(CoDA-Nets), which are performant classifiers with a high degree of inherent interpretability. Their core building blocks are Dynamic Alignment Units (DAUs), which linearly transform their input with weight vectors that dynamically align with task-relevant patterns. As a result, CoDA-Nets model the classification prediction through a series of input-dependent linear transformations, allowing for linear decomposition of the output into individual input contributions. Given the alignment of the DAUs, the resulting contribution maps align with discriminative input patterns. These model-inherent decompositions are of high visual quality and outperform existing attribution methods under quantitative metrics. Further, CoDA-Nets constitute performant classifiers, achieving on par results to ResNet and VGG models on e.g. CIFAR-10 and TinyImagenet.*

1. Introduction

Neural networks are powerful models that excel at a wide range of tasks. However, they are notoriously difficult to interpret and extracting explanations for their predictions is an open research problem. Linear models, in contrast, are generally considered interpretable, because the *contribution* ('the weighted input') of every dimension to the output is explicitly given. Interestingly, many modern neural networks implicitly model the output as a linear transformation of the input; a ReLU-based [21] neural network, e.g., is piece-wise linear and the output thus a linear transformation of the input, cf. [20]. However, due to the highly non-linear manner in which these linear transformations are 'chosen', the corresponding contributions per input dimension do not seem to represent the learnt model parameters well, cf. [1], and a lot of research is being conducted to find better explanations for the decisions of such neural networks, cf. [28, 30, 40, 25, 27, 33, 31, 4].

In this work, we introduce a novel network architecture, the **Convolutional Dynamic Alignment Networks (CoDA-Nets)**, for which the model-inherent contribution maps are faithful projections of the internal computations and thus good 'explanations' of the model prediction. There are two main components to the interpretability of the CoDA-Nets. First, the CoDA-Nets are **dynamic linear**, i.e., they compute their outputs through a series of input-dependent linear transforms, which are based on our novel **Dynamic Alignment Units (DAUs)**. As in linear models, the output can thus be decomposed into individual input contributions, see Fig. 1. Second, the DAUs are structurally biased to compute weight vectors that **align with relevant patterns** in their inputs. In combination, the CoDA-Nets thus inherently produce contribution maps that are 'optimised for interpretability': since each linear transformation matrix and thus their combination is optimised to align with discriminative features, the contribution maps reflect the most discriminative features *as used by the model*.

With this work, we present a new direction for building inherently more interpretable neural network architectures with high modelling capacity. In detail, we would like to highlight the following contributions:

[1] Code will be available at github.com/moboehle/CoDA-Nets.

(1) We introduce the Dynamic Alignment Units (DAUs), which improve the interpretability of neural networks and have two key properties: they are *dynamic linear* and align their weights with discriminative input patterns.
(2) Further, we show that networks of DAUs *inherit* these two properties. In particular, we introduce Convolutional Dynamic Alignment Networks (CoDA-Nets), which are built out of multiple layers of DAUs. As a result, the *model-inherent contribution maps* of CoDA-Nets highlight discriminative patterns in the input.
(3) We further show that the alignment of the DAUs can be promoted by applying a 'temperature scaling' to the final output of the CoDA-Nets.
(4) We show that the resulting contribution maps perform well under commonly employed *quantitative* criteria for attribution methods. Moreover, under *qualitative* inspection, we note that they exhibit a high degree of detail.
(5) Beyond interpretability, CoDA-Nets are performant classifiers and yield competitive classification accuracies on the CIFAR-10 and TinyImagenet datasets.

2. Related work

Interpretability. In order to make machine learning models more interpretable, a variety of techniques has been developed. On the one hand, research has been undertaken to develop model-agnostic explanation methods for which the model behaviour under different inputs is analysed; this includes among others [19, 22, 23]. While their generality and the applicability to any model are advantageous, these methods typically require evaluating the respective model several times and are therefore costly approximations of model behaviour. On the other hand, many techniques that explicitly take advantage of the internal computations have been proposed for explaining the model predictions, including, for example, [28, 30, 40, 25, 27, 33, 31, 4].
In contrast to techniques that aim to explain models *post-hoc*, some recent work has focused on designing new types of network architectures, which are *inherently* more interpretable. Examples of this are the prototype-based neural networks [7], the BagNet [6] and the self-explaining neural networks (SENNs) [3]. Similarly to our proposed architectures, the SENNs and the BagNets derive their explanations from a linear decomposition of the output into contributions from the input (features). This *dynamic linearity*, i.e., the property that the output is computed via some form of an input-dependent linear mapping, is additionally shared by the entire model family of piece-wise linear networks (e.g., ReLU-based networks). In fact, the contribution maps of the CoDA-Nets are conceptually similar to evaluating the 'Input$\times$Gradient' (IxG), cf. [1], on piece-wise linear models, which also yields a linear decomposition in form of a contribution map. However, in contrast to the piece-wise linear functions, we combine this *dynamic linearity* with a structural bias towards an alignment between the contribution maps and the discriminative patterns in the input. This results in explanations of much higher quality, whereas IxG on piece-wise linear models has been found to yield unsatisfactory explanations of model behaviour [1].
Architectural similarities. In our CoDA-Nets, the convolutional kernels are dependent on the specific patch that they are applied on; i.e., a CoDA-Layer might apply different filters at every position in the input. As such, these layers can be regarded as an instance of dynamic local filtering layers as introduced in [15]. Further, our dynamic alignment units (DAUs) share some high-level similarities to attention networks, cf. [35], in the sense that each DAU has a limited budget to distribute over its dynamic weight vectors (bounded norm), which is then used to compute a weighted sum. However, whereas in attention networks the weighted sum is typically computed over vectors, which might even differ from the input to the attention module, a DAU outputs a *scalar* which is a weighted sum of all scalar entries in the input. Moreover, we note that at their optimum (maximal average output over a set of inputs), the DAUs solve a constrained low-rank matrix approximation problem [9]. While low-rank approximations have been used for increasing parameter efficiency in neural networks, cf. [36], this concept has to the best of our knowledge not been used in order to endow neural networks with a structural bias towards finding low-rank approximations of the input for increased interpretability in classification tasks. Lastly, the CoDA-Nets are related to capsule networks. However, whereas in classical capsule networks the activation vectors of the capsules directly serve as input to the next layer, in CoDA-Nets the corresponding vectors are used as convolutional filters. We include a detailed comparison in the supplement.

3. Dynamic Alignment Networks

In this section, we present our novel type of network architecture: the Convolutional Dynamic Alignment Networks (CoDA-Nets). For this, we first introduce Dynamic Alignment Units (DAUs) as the basic building blocks of CoDA-Nets and discuss two of their key properties in sec. 3.1. Concretely, we show that these units linearly transform their inputs with dynamic (input-dependent) weight vectors and, additionally, that they are biased to align these weights with the input during optimisation. We then discuss how DAUs can be used for classification (sec. 3.2) and how we build performant networks out of multiple layers of convolutional DAUs (sec. 3.3). Importantly, the resulting *linear decompositions* of the network outputs are optimised to align with discriminative patterns in the input, making them highly suitable for interpreting the network predictions.

In particular, we structure this section around the following **three important properties** (**P1-P3**) of the DAUs:

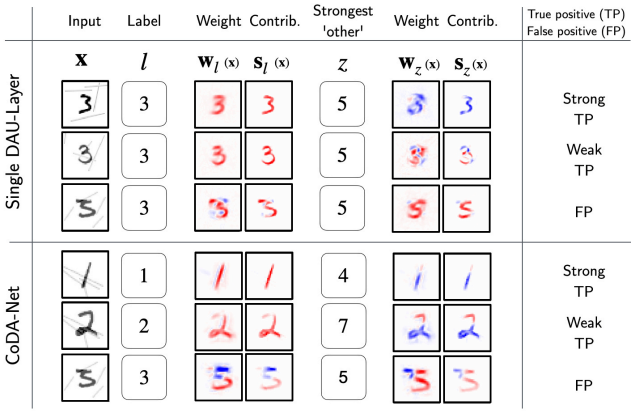

Figure 2: For different inputs $\mathbf{x}$, we visualise the linear weights and contributions (for the single layer, see eq. (4), for the CoDA-Net eq. (8)) for the ground truth label l and the strongest non-label output z. As can be seen, the weights align well with the input images. The first three rows are based on a single DAU layer, the last three on a 5 layer CoDA-Net. The first two samples (rows) per model are correctly classified and the last one is misclassified.

P1: Dynamic linearity. The DAU output o is computed as a dynamic (input-dependent) linear transformation of the input $\mathbf{x}$, such that $o = \mathbf{w}(\mathbf{x})^T\mathbf{x} = \sum_j w_j(\mathbf{x})x_j$. Hence, o can be decomposed into contributions from individual input dimensions, which are given by $w_j(\mathbf{x})x_j$ for dimension j.

P2: Alignment maximisation. Maximising the average output of a single DAU over a set of inputs $\mathbf{x}_i$ maximises the alignment between inputs $\mathbf{x}_i$ and the weight vectors $\mathbf{w}(\mathbf{x}_i)$. As the modelling capacity of $\mathbf{w}(\mathbf{x})$ is restricted, $\mathbf{w}(\mathbf{x})$ will encode the most frequent patterns in the set of inputs $\mathbf{x}_i$.

P3: Inheritance. When combining multiple DAU layers to form a Dynamic Alignment Network (DA-Net), the properties P1 and P2 are *inherited*. In particular, DA-Nets are dynamic linear (P1) and maximising the last layer's output induces an output maximisation in the constituent DAUs (P2).

These properties increase the interpretability of a DA-Net, such as a CoDA-Net (sec. 3.3) for the following reasons. First, the output of a DA-Net can be decomposed into contributions from the individual input dimensions, similar to linear models (cf. Fig. 1, P1 and P3). Second, we note that optimising a neural network for classification applies a maximisation to the outputs of the last layer for every sample. This maximisation aligns the dynamic weight vectors $\mathbf{w}(\mathbf{x})$ of the constituent DAUs of the DA-Net with their respective inputs (cf. Fig. 2, P2 and P3).

Importantly, the weight vectors will align with the *discriminative* patterns in their inputs when optimised for classification as we show in sec. 3.2. As a result, the model-inherent contribution maps of CoDA-Nets are optimised to align well with *discriminative input patterns* in the input image and the interpretability of our models thus forms part of the global optimisation procedure.

3.1. Dynamic Alignment Units

We define the Dynamic Alignment Units (DAUs) by

$$\text{DAU}(\mathbf{x}) = g(\mathbf{ABx} + \mathbf{b})^T\mathbf{x} = \mathbf{w}(\mathbf{x})^T\mathbf{x} \quad . \tag{1}$$

Here, $\mathbf{x} \in \mathbb{R}^d$ is an input vector, $\mathbf{A} \in \mathbb{R}^{d\times r}$ and $\mathbf{B} \in \mathbb{R}^{r\times d}$ are trainable transformation matrices, $\mathbf{b} \in \mathbb{R}^d$ a trainable bias vector, and $g(\mathbf{u}) = \alpha(||\mathbf{u}||)\mathbf{u}$ is a non-linear function that scales the norm of its input. In contrast to using a single matrix $\mathbf{M} \in \mathbb{R}^{d\times d}$, using $\mathbf{AB}$ allows us to control the maximum rank r of the transformation and to reduce the number of parameters; we will hence refer to r as the rank of a DAU. As can be seen by the right-hand side of eq. (1), the DAU linearly transforms the input $\mathbf{x}$ (P1). At the same time, given the quadratic form ($\mathbf{x}^T\mathbf{B}^T\mathbf{A}^T\mathbf{x}$) and the rescaling function $\alpha(||\mathbf{u}||)$, the output of the DAU is a non-linear function of its input. In this work, we focus our analysis on two choices for $g(\mathbf{u})$ in particular[2], namely rescaling to unit norm (L2) and the squashing function (SQ, see [24]):

$$\text{L2}(\mathbf{u}) = \frac{\mathbf{u}}{||\mathbf{u}||_2} \quad \text{and} \quad \text{SQ}(\mathbf{u}) = \text{L2}(\mathbf{u}) \times \frac{||\mathbf{u}||_2^2}{1 + ||\mathbf{u}||_2^2} \tag{2}$$

Under these rescaling functions, the norm of the weight vector is upper-bounded: $||\mathbf{w}(\mathbf{x})|| \leq 1$. Therefore, the output of the DAUs is upper-bounded by the norm of the input:

$$\text{DAU}(\mathbf{x}) = ||\mathbf{w}(\mathbf{x})||\,||\mathbf{x}||\cos(\angle(\mathbf{x}, \mathbf{w}(\mathbf{x}))) \leq ||\mathbf{x}|| \tag{3}$$

As a corollary, for a given input $\mathbf{x}_i$, the DAUs can only achieve this upper bound if $\mathbf{x}_i$ is an eigenvector (EV) of the linear transform $\mathbf{ABx} + \mathbf{b}$. Otherwise, the cosine in eq. (3) will not be maximal[3]. As can be seen in eq. (3), maximising the average output of a DAU over a set of inputs $\{\mathbf{x}_i | \, i = 1, ..., n\}$ maximises the alignment between $\mathbf{w}(\mathbf{x})$ and $\mathbf{x}$ (P2). In particular, it optimises the parameters of the DAU such that the *most frequent input patterns* are encoded as EVs in the linear transform $\mathbf{ABx} + \mathbf{b}$, similar to an r-dimensional PCA decomposition (r the rank of $\mathbf{AB}$). In fact, as discussed in the supplement, the optimum of the DAU maximisation solves a low-rank matrix approximation [9] problem similar to singular value decomposition. As an illustration of this property, in Fig. 3 we show the 3 EVs[4] of matrix $\mathbf{AB}$ (with rank $r = 3$, bias $\mathbf{b} = \mathbf{0}$) after optimising a DAU over a set of n noisy samples of 3 specific MNIST [18] images; for this, we used $n = 3072$ and zero-mean Gaussian noise. As expected, the EVs of $\mathbf{AB}$ encode the original, noise-free images, since this on average maximises the alignment (eq. (3)) between the weight vectors $\mathbf{w}(\mathbf{x}_i)$ and the input samples $\mathbf{x}_i$ over the dataset.

[2] In preliminary experiments we observed comparable behaviour over a range of different normalisation functions such as, e.g., L1 normalisation.

[3] Note that $\mathbf{w}(\mathbf{x})$ is proportional to $\mathbf{ABx} + \mathbf{b}$. The cosine in eq. (3), in turn, is maximal if and only if $\mathbf{w}(\mathbf{x}_i)$ is proportional to $\mathbf{x}_i$ and thus, by transitivity, if $\mathbf{x}_i$ is proportional to $\mathbf{ABx}_i + \mathbf{b}$. This means that $\mathbf{x}_i$ has to be an EV of $\mathbf{ABx} + \mathbf{b}$ to achieve maximal output.

[4] Given $r = 3$, the EVs maximally span a 3-dimensional subspace.

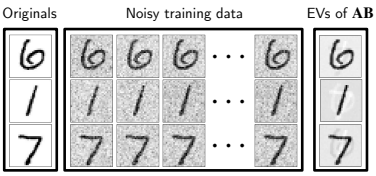

Figure 3: Eigenvectors (EVs) of $\mathbf{AB}$ after maximising the output of a rank-3 DAU over a set of noisy samples of 3 MNIST digits. Effectively, the DAUs encode the most frequent components in their EVs, similar to a principal component analysis (PCA).

3.2. DAUs for classification

DAUs can be used directly for classification by applying k DAUs in parallel to obtain an output $\hat{\mathbf{y}}(\mathbf{x}) = [\text{DAU}_1(\mathbf{x}), ..., \text{DAU}_k(\mathbf{x})]$. Note that this is a linear transformation $\hat{\mathbf{y}}(\mathbf{x}){=}\mathbf{W}(\mathbf{x})\mathbf{x}$, with each row in $\mathbf{W}{\in}\mathbb{R}^{k\times d}$ corresponding to the weight vector $\mathbf{w}_j^T$ of a specific DAU j. In particular, consider a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) | \mathbf{x}_i \in \mathbb{R}^d, \mathbf{y}_i \in \mathbb{R}^k\}$ of k classes with 'one-hot' encoded labels $\mathbf{y}_i$ for the inputs $\mathbf{x}_i$. To optimise the DAUs as classifiers on $\mathcal{D}$, we can apply a sigmoid non-linearity to each DAU output and optimise the loss function $\mathcal{L} = \sum_i \text{BCE}(\sigma(\hat{\mathbf{y}}_i), \mathbf{y}_i)$, where BCE denotes the binary cross-entropy and σ applies the sigmoid function to each entry in $\hat{\mathbf{y}}_i$. Note that for a given sample, BCE either maximises (DAU for correct class) or minimises (DAU for incorrect classes) the output of each DAU. Hence, this classification loss will still maximise the (signed) cosine between the weight vectors $\mathbf{w}(\mathbf{x}_i)$ and $\mathbf{x}_i$.

To illustrate this property, in Fig. 2 (top) we show the weights $\mathbf{w}(\mathbf{x}_i)$ for several samples of the digit '3' after optimising the DAUs for classification on a noisy MNIST dataset; the first two are correctly classified, the last one is misclassified as a '5'. As can be seen, the weights align with the respective input (the weights for different samples are different). However, different parts of the input are either positively or negatively correlated with a class, which is reflected in the weights: for example, the extended stroke on top of the '3' in the misclassified sample is assigned *negative weight* and, since the background noise is *uncorrelated* with the class labels, it is not represented in the weights.

In a classification setting, the DAUs thus encode *the most frequent discriminative patterns* in the linear transform $\mathbf{ABx} + \mathbf{b}$ such that the dynamic weights $\mathbf{w}(\mathbf{x})$ align well with these patterns. Additionally, since the output for class j is a linear transformation of the input (P1), we can compute the contribution vector $\mathbf{s}_j$ containing the per-pixel contributions to this output by the element-wise product ($\odot$)

$$\mathbf{s}_j(\mathbf{x}_i) = \mathbf{w}_j(\mathbf{x}_i) \odot \mathbf{x}_i \quad , \tag{4}$$

see Figs. 1 and 2. Such linear decompositions constitute the model-inherent 'explanations' which we evaluate in sec. 4.

3.3. Convolutional Dynamic Alignment Networks

The modelling capacity of a single layer of DAUs is limited, similar to a single linear classifier. However, DAUs can be used as the basic building block for deep convolutional neural networks, which yields powerful classifiers. Importantly, in this section we show that such a Convolutional Dynamic Alignment Network (CoDA-Net) inherits the properties (P3) of the DAUs by maintaining both the dynamic linearity (P1) as well as the alignment maximisation (P2). For a convolutional dynamic alignment layer, each filter is modelled by a DAU, similar to dynamic local filtering layers [15]. Note that the output of such a layer is also a dynamic linear transformation of the input to that layer, since a convolution is equivalent to a linear layer with certain constraints on the weights, cf. [26]. We include the implementation details in the supplement. Finally, at the end of this section, we highlight an important difference between output maximisation and optimising for classification with the BCE loss. In this context we discuss the effect of *temperature scaling* and present the loss function we optimise in our experiments.

Dynamic linearity (P1). In order to see that the linearity is maintained, we note that the successive application of multiple layers of DAUs also results in a dynamic linear mapping. Let $\mathbf{W}_l$ denote the linear transformation matrix produced by a layer of DAUs and let $\mathbf{a}_{l-1}$ be the input vector to that layer; as mentioned before, each row in the matrix $\mathbf{W}_l$ corresponds to the weight vector of a single DAU[5]. As such, the output of this layer is given by

$$\mathbf{a}_l = \mathbf{W}_l(\mathbf{a}_{l-1})\mathbf{a}_{l-1} \quad . \tag{5}$$

In a network of DAUs, the successive linear transformations can thus be collapsed. In particular, *for any pair of activation vectors* $\mathbf{a}_{l_1}$ and $\mathbf{a}_{l_2}$ with $l_1 < l_2$, the vector $\mathbf{a}_{l_2}$ can be expressed as a linear transformation of $\mathbf{a}_{l_1}$:

$$\mathbf{a}_{l_2} = \mathbf{W}_{l_1 \to l_2}(\mathbf{a}_{l_1})\, \mathbf{a}_{l_1} \tag{6}$$

$$with \quad \mathbf{W}_{l_1 \to l_2}(\mathbf{a}_{l_1}) = \textstyle\prod_{k=l_1+1}^{l_2} \mathbf{W}_k(\mathbf{a}_{k-1}) \quad . \tag{7}$$

For example, the matrix $\mathbf{W}_{0\to L}(\mathbf{a}_0 = \mathbf{x}) = \mathbf{W}(\mathbf{x})$ models the linear transformation from the input to the output space, see Fig. 1. Since this linearity holds between any two layers, the j-th entry of any activation vector $\mathbf{a}_l$ in the network can be decomposed into input contributions via:

$$\mathbf{s}_j^l(\mathbf{x}_i) = [\mathbf{W}_{0\to l}(\mathbf{x}_i)]_j^T \odot \mathbf{x}_i \quad , \tag{8}$$

with $[\mathbf{W}]_j$ the j-th row in the matrix.

[5]Note that this also holds for convolutional DAU layers. Specifically, each row in the matrix $\mathbf{W}_l$ corresponds to a single DAU applied to exactly one spatial location in the input and the input with spatial dimensions is vectorised to yield $\mathbf{a}_{l-1}$. For further details, we kindly refer the reader to [26] and the implementation details in the supplement of this work.

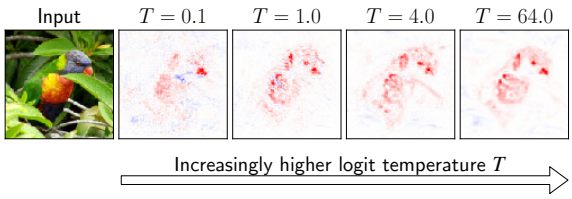

Figure 4: By lowering the upper bound (cf. eq. (3)), the correlation maximisation in the DAUs can be emphasised. We show contribution maps for a model trained with different temperatures.

Alignment maximisation (P2). Note that the output of a CoDA-Net is bounded independent of the network parameters: since each DAU operation can—independent of its parameters—at most reproduce the norm of its input (eq. (3)), the linear concatenation of these operations necessarily also has an upper bound which does not depend on the parameters. Therefore, in order to achieve maximal outputs on average (e.g., the class logit over the subset of images of that class), all DAUs in the network need to produce weights $\mathbf{w}(\mathbf{a}_l)$ that align well with the class features. In other words, the weights will align with discriminative patterns in the input. For example, in Fig. 2 (bottom), we visualise the 'global matrices' $\mathbf{W}_{0\to L}$ and the corresponding contributions (eq. (8)) for a $L = 5$ layer CoDA-Net. As before, the weights align with discriminative patterns in the input and do not encode the uninformative noise.

Temperature scaling and loss function. So far we have assumed that minimising the BCE loss for a given sample is equivalent to applying a maximisation or minimisation loss to the individual outputs of a CoDA-Net. While this is in principle correct, BCE introduces an additional, non-negligible effect: *saturation*. Specifically, it is possible for a CoDA-Net to achieve a low BCE loss without the need to produce well-aligned weight vectors. As soon as the classification accuracy is high and the outputs of the networks are large, the gradient—and therefore the *alignment pressure*—will vanish. This effect can, however, easily be mitigated: as discussed in the previous paragraph, the output of a CoDA-Net is upper-bounded *independent of the network parameters*, since each individual DAU in the network is upper-bounded. By scaling the network output with a temperature parameter T such that $\hat{\mathbf{y}}(\mathbf{x}) = T^{-1}\mathbf{W}_{0\to L}(\mathbf{x})\,\mathbf{x}$, we can explicitly decrease this upper bound and thereby increase the *alignment pressure* in the DAUs by avoiding the early saturation due to BCE. In particular, the lower the upper bound is, the stronger the induced DAU output maximisation should be, since the network needs to accumulate more signal to obtain large class logits (and thus a negligible gradient). This is indeed what we observe both qualitatively, cf. Fig. 4, and quantitatively, cf. Fig. 6 (right column). Alternatively, the representation of the network's computation as a linear mapping allows to directly regularise what properties these linear mappings should fulfill. For example, we show in the supplement that by regularising the absolute values of the matrix $\mathbf{W}_{0\to L}$, we can induce sparsity in the signal alignments, which can lead to sharper heatmaps. The overall loss for an input $\mathbf{x}_i$ and the target vector $\mathbf{y}_i$ is thus computed as

$$\mathcal{L}(\mathbf{x}_i, \mathbf{y}_i) = \mathrm{BCE}(\sigma(T^{-1}\mathbf{W}_{0\to L}(\mathbf{x}_i)\,\mathbf{x}_i + \mathbf{b}_0)\,,\,\mathbf{y}_i) \quad (9)$$

$$+ \lambda\langle|\mathbf{W}_{0\to L}(\mathbf{x}_i)|\rangle \quad . \quad (10)$$

Here, λ is the strength of the regularisation, σ applies the sigmoid activation to each vector entry, $\mathbf{b}_0$ is a fixed bias term, and $\langle|\mathbf{W}_{0\to L}(\mathbf{x}_i)|\rangle$ refers to the mean over the absolute values of all entries in the matrix $\mathbf{W}_{0\to L}(\mathbf{x}_i)$.

3.4. Implementation details

Shared matrix B. In our experiments, we opted to share the matrix $\mathbf{B} \in \mathbb{R}^{r\times d}$ between all DAUs in a given layer. This increases parameter efficiency by having the DAUs share a common r-dimensional subspace and still fixes the maximal rank of each DAU to the chosen value of r.

Input encoding. In sec. 3.1, we showed that the norm-weighted cosine similarity between the dynamic weights and the layer inputs is optimised and the output of a DAU is at most the norm of its input. This favours pixels with large RGB values, since these have a larger norm and can thus produce larger outputs in the maximisation task. To mitigate this bias, we add the negative image as three additional color channels and thus encode each pixel in the input as $[r, g, b, 1-r, 1-g, 1-b]$, with $r, g, b \in [0, 1]$.

4. Results

In sec. 4.1, we describe the experimental setup, assess the classification performance of the CoDA-Nets and discuss their efficiency. Further, in sec. 4.2 we evaluate the model-inherent contribution maps derived from $\mathbf{W}_{0\to L}$ (cf. eq. (8)) and compare them both *qualitatively* (Fig. 5) as well as *quantitatively* (Fig. 6) to other attribution methods.

4.1. Setup and model performance

Datasets. We evaluate and compare the accuracies of the CoDA-Nets to other work on the CIFAR-10 [16] and the TinyImagenet [10] datasets. We use the same datasets for the quantitative evaluations of the model-inherent contribution maps. Additionally, we qualitatively show high-resolution examples from a CoDA-Net trained on the first 100 classes of the Imagenet dataset.

Models. We evaluate models of four different sizes denoted by (S/M/L/XL)-CoDA on CIFAR-10 (S and M), Imagenet-100 (L), and TinyImagenet (XL); these models have 8M (S), 28M (M), 48M (L), and 62M (XL) parameters respectively; see the supplement for an evaluation of the impact of model size on accuracy. All models feature 9 convolutional DAU layers and a final sum-pooling layer, and mainly vary in the

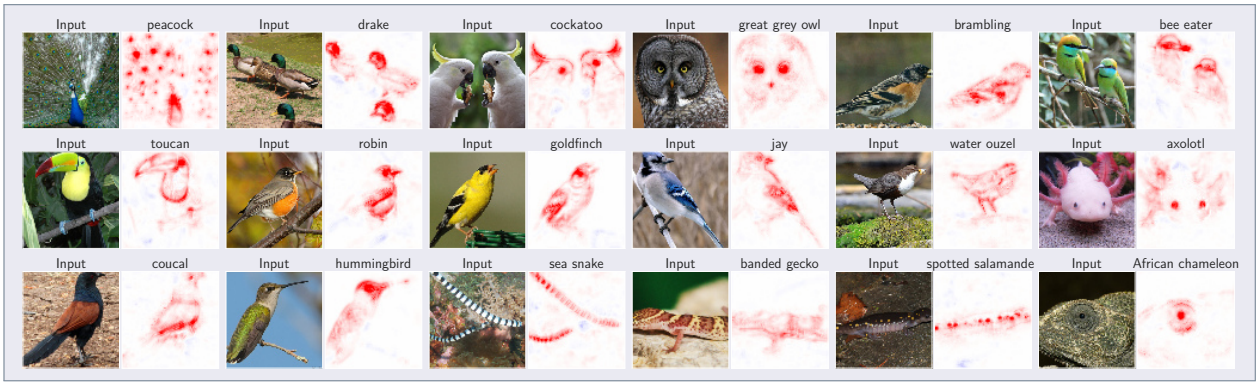

Figure 5: Model-inherent contribution maps for the most confident predictions for 18 different classes, sorted by confidence (high to low). We show positive (negative) contributions (eq. (8)) per spatial location for the ground truth class logit in red (blue).

Model	C10	Model	T-IM
SENNs [3]	78.5%	ResNet-34 [32]	52.0%
VGG-19 [11]	91.5%	VGG 16 [17]	52.2%
DE-CapsNet [14]	93.0%	VGG 16 + aug [17]	56.4%
ResNet-56 [12]	93.6%	IRRCNN [2]	52.2%
WRN-28-2 [12]	94.9%	ResNet-110 [34]	56.6%
WRN-28-2 + aug [8]	95.8%	WRN-40-20 [13]	63.8%
S-CoDA-SQ (λ)	93.8%	XL-CoDA-SQ (T)	54.4%
S-CoDA-L2 (λ)	92.6%	XL-CoDA-SQ + aug (T)	58.4%
S-CoDA-SQ (T)	93.2%		
S-CoDA-L2 (T)	93.0%		
M-CoDA-SQ + aug (λ)	96.5%		

Table 1: CIFAR-10 (**C10**) and TinyImagenet (**T-IM**) classification accuracies. Results taken from specified references. The prefix of the CoDAs indicates model size, the suffix the non-linearity used (eq. (2)). With (λ) and (T) we denote if models were trained with regularisation or increased temperature T, see eq. (9).

number of features, the rank r of the DAUs, and the convolutional strides for reducing the spatial dimensions. No additional methods such as residual connections, dropout, or batch normalisation are used. This 9-layer architecture was initially optimised for the CIFAR-10 dataset and subsequently adapted to the TinyImagenet and Imagenet-100 datasets. Further, we investigate the effect that the temperature T, the regularisation λ, and the non-linearities (L2, SQ, see eq. (2)) have on the CoDA-Nets. Given the computational cost of the regularisation (two additional passes to extract and regularise $\mathbf{W}_{0\to L}$), we evaluate the regularisation on models trained on CIFAR-10. Lastly, models marked with T (λ) in Table 1 were trained with λ=0 (T=64, equiv. to 'average pooling'). Details on architectures and training procedure are included in the supplement.

Classification performance. In Table 1 we compare the performances of our CoDA-Nets to several other published results. Note that the referenced numbers are meant to be used as a gauge for assessing the CoDA-Net performance and do not exhaustively represent the state of the art. In particular, we would like to highlight that the CoDA-Net performance is on par to models of the VGG [29] and ResNet [12] model families on both datasets. Moreover, under the same data augmentation (RandAugment [8]), it achieves similar results as the WideResNet-28-2 [37] on CIFAR-10. Additionally, we list the reported results of the SENNs [3] and the DE-CapsNet [14] architectures for CIFAR-10. Similar to our CoDA-Nets, the SENNs were designed to improve network interpretability and are also based on the idea of explicitly modelling the output as a dynamic linear transformation of the input. On the other hand, the CoDA-Nets share similarities to capsule networks, which we discuss in the supplement; to the best of our knowledge, the DE-CapsNet currently achieves the state of the art in the field of capsule networks on CIFAR-10. Overall, we observed that the CoDA-Nets deliver competitive performances that are fairly robust to the non-linearity (L2, SQ), the temperature (T), and the regularisation strength (λ). We note that on average SQ performed better than L2, which we ascribe to the fact that SQ avoids up-scaling vectors with low norm ($||\mathbf{v}|| < 1$, see eq. (2)).

Efficiency considerations. The CoDa-Nets achieve good accuracies on the presented datasets, exhibit training behaviour that is robust over a wide range of hyperparameters, and are as fast as a typical ResNet at inference time. However, under the current formulation and without highly optimised GPU implementations for the DAUs, training times are significantly longer for the CoDA-Nets. While we are currently working on an improved and optimised version of CoDA-Nets, we were not yet able to generate results for the full ImageNet dataset. On the 100 classes subset, however, the evaluated L-CoDA-SQ network achieved competitive performance (76.5% accuracy, for details see supplement) and offers highly detailed explanations for its predictions, as we show in Figs. 5 and 8.

4.2. Interpretability of CoDA-Nets

In the following, we evaluate the model-inherent contribution maps and compare them to other commonly used methods for importance attribution. The evaluations are based on the XL-CoDA-SQ (T=6400) for TinyImagenet and the S-CoDA-SQ (T=1000) for CIFAR-10, see Table 1 for the

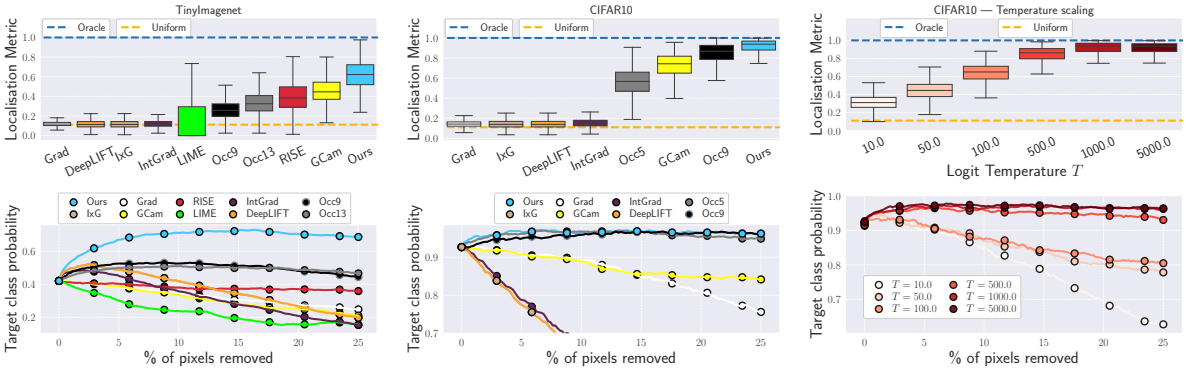

Figure 6: **Top row:** Results for the localisation metric, see eq. (11). **Bottom row:** Pixel removal metric. In particular, we plot the mean target class probability after removing the $x\%$ of the *least important* pixels. We show the results of a CoDA-Net trained on TinyImagenet **(left column)**, as well as on CIFAR-10 **(center column)**. Additionally, we show the effect of the temperature parameter on the interpretability of the CoDA-Nets **(right column)**: as expected, a higher temperature leads to higher interpretability (sec. 3.4).

respective accuracies. Further, we evaluate the effect of training the same CIFAR-10 architecture with different temperatures T; as discussed in sec. 3.3, we expect the interpretability to *increase* along with T, since for larger T a stronger alignment is required in order for the models to obtain large class logits. Evaluations of models trained with L1-regularisation of the matrices $\mathbf{M}_{0\to L}$ (eq. (9)) and of models with the L2 non-linearity (eq. (2)) are included in the supplement. The respective results are similar to those presented here. Before turning to the results, however, in the following we will first present the attribution methods used for comparison and discuss the evaluation metrics employed for quantifying their interpretability.

Attribution methods. We compare the model-inherent contribution maps (cf. eq. (8)) to other common approaches for importance attribution. In particular, we evaluate against several perturbation based methods such as RISE [22], LIME [23], and several occlusion attributions [38] (Occ-K, with K the size of the occlusion patch). Additionally, we evaluate against common gradient-based methods. These include the gradient of the class logits with respect to the input image [5] (Grad), 'Input×Gradient' (IxG, cf. [1]), GradCam [25] (GCam), Integrated Gradients [33] (IntG), and DeepLIFT [27]. As a baseline, we also evaluated these methods on a pre-trained ResNet-56 [12] on CIFAR-10, for which we show the results in the supplement.

Evaluation metrics. Our quantitative evaluation of the attribution maps is based on the following two methods: we **(1)** evaluate a localisation metric by adapting the pointing game [39] to the CIFAR-10 and TinyImagenet datasets, and **(2)** analyse the model behaviour under the pixel removal strategy employed in [31]. For **(1)**, we evaluate the attribution methods on a grid of $n \times n$ with $n = 3$ images sampled from the corresponding datasets; in every grid of images, each class may occur at most once. For a visualisation with $n = 2$, see Fig. 7. For each occurring class, we can measure how much positive importance an attribution method assigns to the respective class image. Let $\mathcal{I}_c$ be the image for class c, then the score s_c for this class is calculated as

$$s_c = \frac{1}{Z} \sum_{p_c \in \mathcal{I}_c} p_c \quad \text{with} \quad Z = \sum_k \sum_{p_c \in \mathcal{I}_k} p_c \quad , \quad (11)$$

with p_c the positive attribution for class c assigned to the spatial location p. This metric has the same clear oracle score $s_c = 1$ for all attribution methods (all positive attributions located in the correct grid image) and a clear score for completely random attributions $s_c = 1/n^2$ (the positive attributions are uniformly distributed over the different grid images). Since this metric depends on the classification ac-

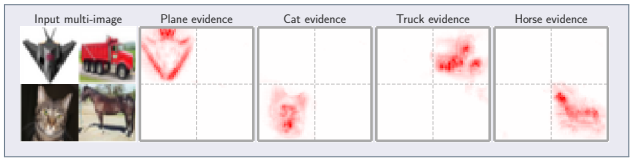

Figure 7: A multi-image on the CIFAR-10 dataset. The CoDA-Net contribution maps highlight the individual class-images well.

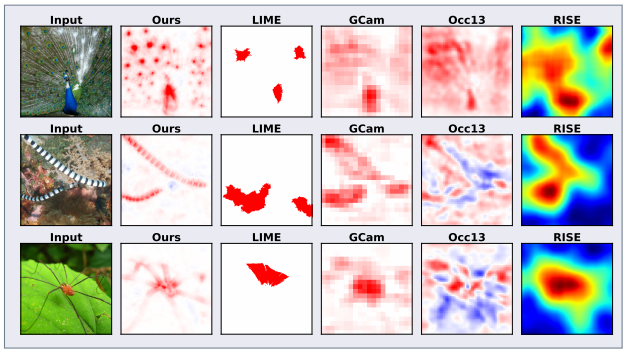

Figure 8: Comparison to the strongest post-hoc methods. While the regions of importance roughly coincide, the inherent contribution maps of the CoDA-Nets offer the most detail. Note that to improve the RISE visualisation, we chose its default colormap [22]; the most (least) important values are still shown in red (blue).

curacy of the models, we sample the first 500 (CIFAR-10) or 250 (TinyImagenet) images according to their class score for the ground-truth class[6]; note that since all attributions are evaluated for the same model on the same set of images, this does not favour any particular attribution method.
For (2), we show how the model's class score behaves under the removal of an increasing amount of *least important* pixels, where the importance is obtained via the respective attribution method. Since the first pixels to be removed are typically assigned negative or relatively little importance, we expect the model to initially increase its confidence (removing pixels with *negative* impact) or maintain a similar level of confidence (removing pixels with *low* impact) if the evaluated attribution method produces an accurate ranking of the pixel importance values. Conversely, if we were to remove the *most important* pixels first, we would expect the model confidence to quickly decrease. However, as noted by [31], removing the most important pixels first introduces artifacts in the most important regions of the image and is therefore potentially more unstable than removing the least important pixels first. Nevertheless, the model-inherent contribution maps perform well in this setting, too, as we show in the supplement. Lastly, in the supplement we qualitatively show that they pass the 'sanity check' of [1].

Quantitative results. In Fig. 6, we compare the contribution maps of the CoDA-Nets to other attributions under the evaluation metrics discussed above. It can be seen that the CoDA-Nets (1) perform well under the localisation metric given by eq. (11) and outperform all the other attribution methods evaluated on the same model, both for TinyImagenet (top row, left) and CIFAR-10 (top row, center); note that we excluded RISE and LIME on CIFAR-10, since the default parameters do not seem to transfer well to this low-resolution dataset. Moreover, (2) the CoDA-Nets perform well in the pixel-removal setting: the *least salient* locations according to the model-inherent contributions indeed seem to be among the least relevant for the given class score on both datasets, see Fig. 6 (bottom row, left and center). Further, in Fig. 6 (right column), we show the effect of temperature scaling on the interpretability of CoDA-Nets trained on CIFAR-10. The results indicate that the alignment maximisation is indeed crucial for interpretability and constitutes an important difference of the CoDA-Nets to other dynamic linear networks such as piece-wise linear networks (ReLU-based networks). In particular, by structurally requiring a strong alignment for confident classifications, the interpretability of the CoDA-Nets forms part of the optimisation objective. Increasing the temperature increases the alignment and thereby the interpretability of the CoDA-Nets. While we observe a downward trend in classification accuracy when increasing T, the best model at $T = 10$ only slightly improved the accuracy compared to $T = 1000$ ($93.2\% \rightarrow 93.6\%$); for more details, see supplement.

In summary, the results show that by combining dynamic linearity with a structural bias towards an alignment with discriminative patterns, we obtain models which inherently provide an interpretable linear decomposition of their predictions. Further, given that we better understand the relationship between the intermediate computations and the optimisation of the final output in the CoDA-Nets, we can emphasise model interpretability in a principled way by increasing the 'alignment pressure' via *temperature scaling*.

Qualitative results. In Fig. 5, we visualise spatial contribution maps of the L-CoDA-SQ model (trained on Imagenet-100) for some of its most confident predictions. Note that these contribution maps are linear decompositions of the output and the sum over these maps yields the respective class logit. In Fig. 8, we additionally present a visual comparison to the best-performing post-hoc attribution methods; note that RISE cannot be displayed well under the same color coding and we thus use its default visualisation. We observe that the different methods are not inconsistent with each other and roughly highlight similar regions. However, the inherent contribution maps are of much higher detail and compared to the perturbation-based methods do not require multiple model evaluations. Much more importantly, however, all the other methods are attempts at approximating the model behaviour *post-hoc*, while the CoDA-Net contribution maps in Fig. 5 are derived from the model-inherent linear mapping that is used to compute the model output.

5. Discussion and conclusion

In this work, we presented a new family of neural networks, the CoDA-Nets, and show that they are performant classifiers with a high degree of interpretability. For this, we first introduced the Dynamic Alignment Units, which model their output as a dynamic linear transformation of their input and have a structural bias towards alignment maximisation. Using the DAUs to model filters in a convolutional network, we obtain the Convolutional Dynamic Alignment Networks (CoDA-Nets). The successive linear mappings by means of the DAUs within the network make it possible to linearly decompose the output into contributions from individual input dimensions. In order to assess the quality of these contribution maps, see eq. (8), we compare against other attribution methods. We find that the CoDA-Net contribution maps consistently perform well under commonly used quantitative metrics. Beyond their *interpretability*, the CoDA-Nets constitute performant classifiers: their accuracy on CIFAR-10 and the TinyImagenet dataset are on par to the commonly employed VGG and ResNet models.

[6] We can only expect an attribution to specifically highlight a class image if this image can be correctly classified on its own. If all grid images have similarly low attributions, the localisation score will be random.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian J. Goodfellow, Moritz Hardt, and Been Kim. Sanity Checks for Saliency Maps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[2] Md Zahangir Alom, Mahmudul Hasan, Chris Yakopcic, Tarek M. Taha, and Vijayan K. Asari. Improved inception-residual convolutional neural network for object recognition. *Neural Computing and Applications*, 2020.

[3] David Alvarez-Melis and Tommi S. Jaakkola. Towards Robust Interpretability with Self-Explaining Neural Networks. In *Advances in Neural Information Processing (NeurIPS)*, 2018.

[4] 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*, 2015.

[5] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 2010.

[6] Wieland Brendel and Matthias Bethge. Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet. In *International Conference on Learning Representations (ICLR)*. OpenReview.net, 2019.

[7] 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 (NeurIPS)*, 2019.

[8] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. RandAugment: Practical data augmentation with no separate search. *CoRR*, abs/1909.13719, 2019.

[9] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1936.

[10] Johnson et al. Tiny ImageNet Visual Recognition Challenge. https://tiny-imagenet.herokuapp.com/. Accessed: 2020-11-10.

[11] Henry Gouk, Bernhard Pfahringer, Eibe Frank, and Michael J. Cree. MaxGain: Regularisation of Neural Networks by Constraining Activation Magnitudes. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018*, 2018.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using Pre-Training Can Improve Model Robustness and Uncertainty. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of Machine Learning Research (PMLR)*, 2019.

[14] Bohan Jia and Qiyu Huang. DE-CapsNet: A Diverse Enhanced Capsule Network with Disperse Dynamic Routing. *Applied Sciences*, 2020.

[15] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[16] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[17] learningai.io. VGGNet and Tiny ImageNet. https://learningai.io/projects/2017/06/29/tiny-imagenet.html. Accessed: 2020-11-08.

[18] Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[19] Scott M. Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[20] Guido F. Montúfar, Razvan Pascanu, KyungHyun Cho, and Yoshua Bengio. On the Number of Linear Regions of Deep Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[21] Vinod Nair and Geoffrey E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *International Conference on Machine Learning (ICML)*, 2010.

[22] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: Randomized Input Sampling for Explanation of Black-box Models. In *British Machine Vision Conference (BMVC)*, 2018.

[23] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the predictions of any classifier. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2016.

[24] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic Routing Between Capsules. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[25] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *International Conference on Computer Vision (ICCV)*, 2017.

[26] Irhum Shafkat. Intuitively Understanding Convolutions for Deep Learning. https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1#ad33, 2018.

[27] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning Important Features Through Propagating Activation Differences. In *International Conference on Machine Learning (ICML)*, 2017.

[28] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *International Conference on Learning Representations (ICLR), Workshop*, 2014.

[29] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Yoshua Bengio and Yann LeCun, editors, *International Conference on Learning Representations (ICLR)*, 2015.

[30] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for Simplicity:

The All Convolutional Net. In *International Conference on Learning Representations (ICLR), Workshop*, 2015.

[31] Suraj Srinivas and François Fleuret. Full-Gradient Representation for Neural Network Visualization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[32] Lei Sun. ResNet on Tiny ImageNet. http://cs231n.stanford.edu/reports/2017/pdfs/12.pdf, 2016. Accessed: 2020-11-16.

[33] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In Doina Precup and Yee Whye Teh, editors, *International Conference on Machine Learning (ICML)*, 2017.

[34] Chakkrit Termritthikun, Yeshi Jamtsho, and Paisarn Muneesawang. An improved residual network model for image recognition using a combination of snapshot ensembles and the cutout technique. *Multimedia Tools and Applications*, 2020.

[35] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning (ICML)*, 2015.

[36] Xiyu Yu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. On compressing deep models by low rank and sparse decomposition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[37] Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *British Machine Vision Conference (BMVC)*, 2016.

[38] Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *European Conference on Computer Vision (ECCV)*, 2014.

[39] 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.*, 2018.

[40] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.