



B-cos Networks: Alignment Is All We Need for Interpretability

Moritz Böhle¹, Mario Fritz², and Bernt Schiele¹

¹Max Planck Institute for Informatics, ²CISPA Helmholtz Center for Information Security



mboehle@mpi-inf.mpg.de
github.com/moboehle/B-cos

Summary: B-cos Networks to go

- DNNs typically explained by post-hoc methods
- Instead, we make DNNs **inherently interpretable**
- For this, we introduce $\mathbf{B}\text{-cos}(\mathbf{x}) = \mathbf{w}^T(\mathbf{x}) \mathbf{x}$
⇒ B-cos networks are dynamic linear $y(\mathbf{x}) = \mathbf{W}_{1 \rightarrow L}(\mathbf{x}) \mathbf{x}$
- $\mathbf{W}_{1 \rightarrow L}(\mathbf{x})$ optimised to align with \mathbf{x} ⇒ **easy to interpret**
- Compatible with standard CNN architectures

Method: The B-cos Transformation

B-cos Transformation

$$\mathbf{B}\text{-cos}(\mathbf{x}) = \underbrace{\|\widehat{\mathbf{w}}\|}_{=1} \|\mathbf{x}\| |\cos(\mathbf{x}, \mathbf{w})|^B \times \text{sgn}(\cos(\mathbf{x}, \mathbf{w}))$$

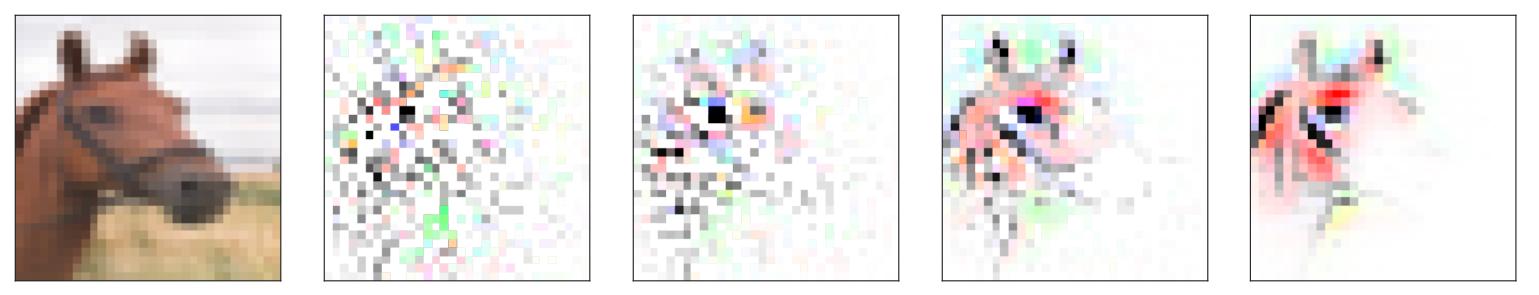
B-cos Properties

Dynamic linear
 $\mathbf{B}\text{-cos}(\mathbf{x}) = \mathbf{w}^T(\mathbf{x}) \mathbf{x}$

Bounded
 $\mathbf{B}\text{-cos}(\mathbf{x}) \leq \|\mathbf{x}\|$

Maximal if aligned
 $\mathbf{B}\text{-cos}(\mathbf{x}) = \|\mathbf{x}\| \Leftrightarrow \mathbf{x} \parallel \mathbf{w}$

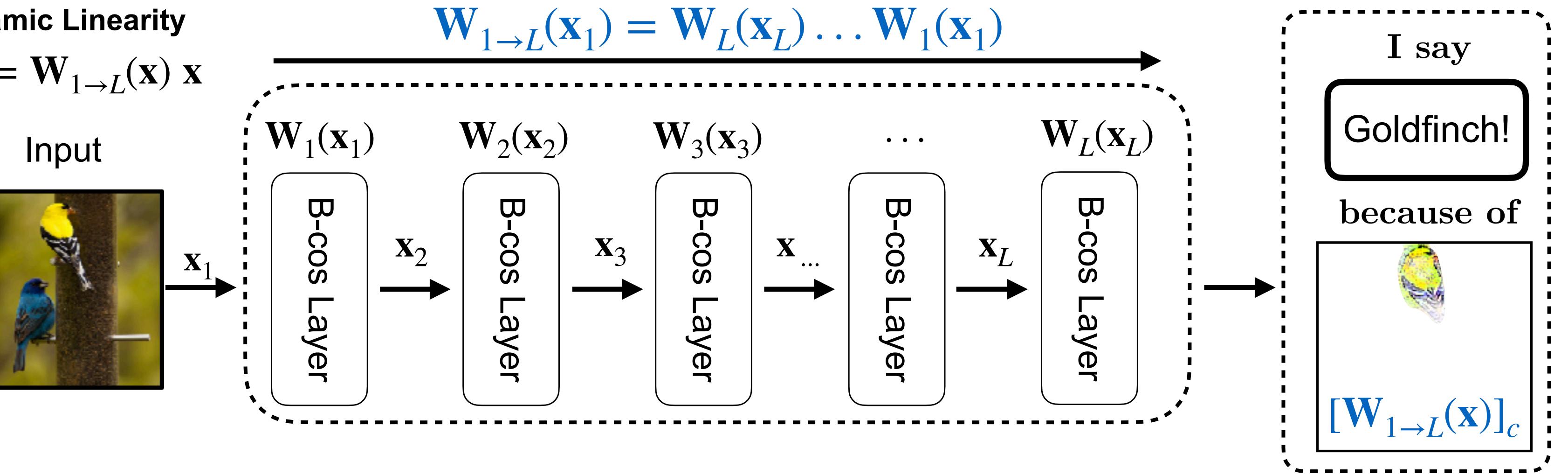
- $\mathbf{B}\text{-cos}(\mathbf{x})$ is a **modified** linear transformation
that **suppresses outputs for badly aligned weights**
⇒ can **replace linear transformations** in CNNs!
- During optimisation, B-cos induces **weight alignment**



Higher B → higher weight alignment

High-level Overview: Dynamic Linearity + Input Alignment

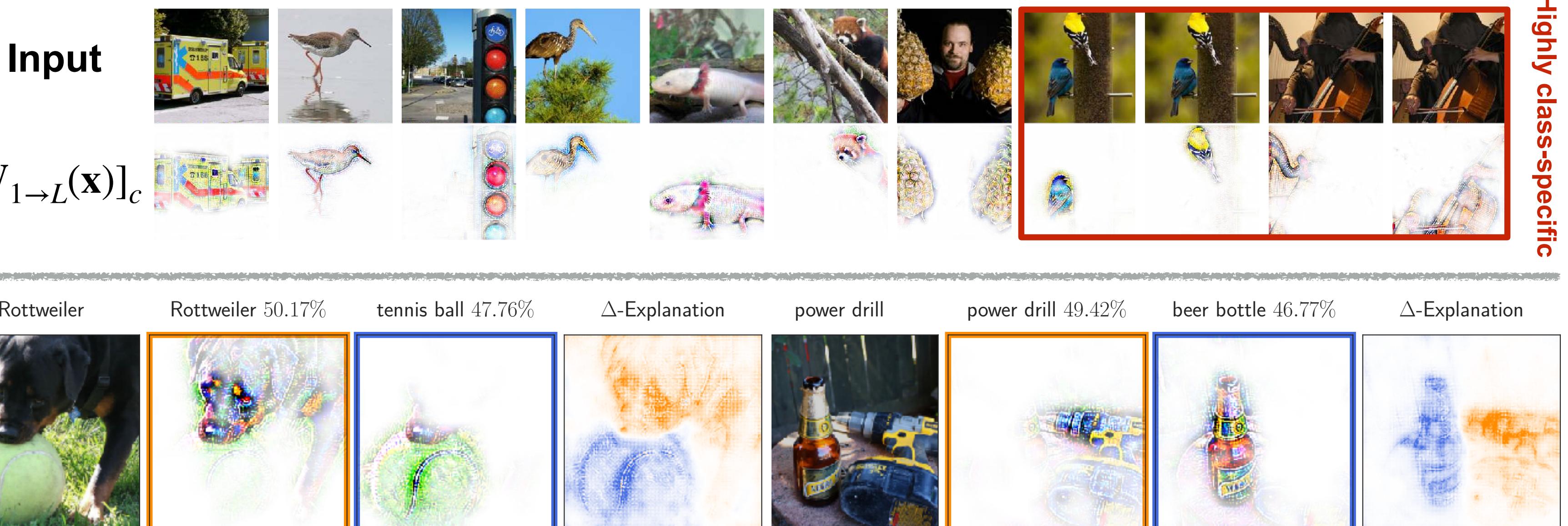
Dynamic Linearity
 $y(\mathbf{x}) = \mathbf{W}_{1 \rightarrow L}(\mathbf{x}) \mathbf{x}$



Given the **dynamic linearity**, $\mathbf{W}_{1 \rightarrow L}(\mathbf{x})$ **faithfully summarises the model**

By promoting **weight-input alignment**, we make $\mathbf{W}_{1 \rightarrow L}(\mathbf{x})$ **interpretable** (see B-cos, left)

Model-inherent explanations $\mathbf{W}_{1 \rightarrow L}(\mathbf{x})$ can directly be visualised in colour (no overlay)!

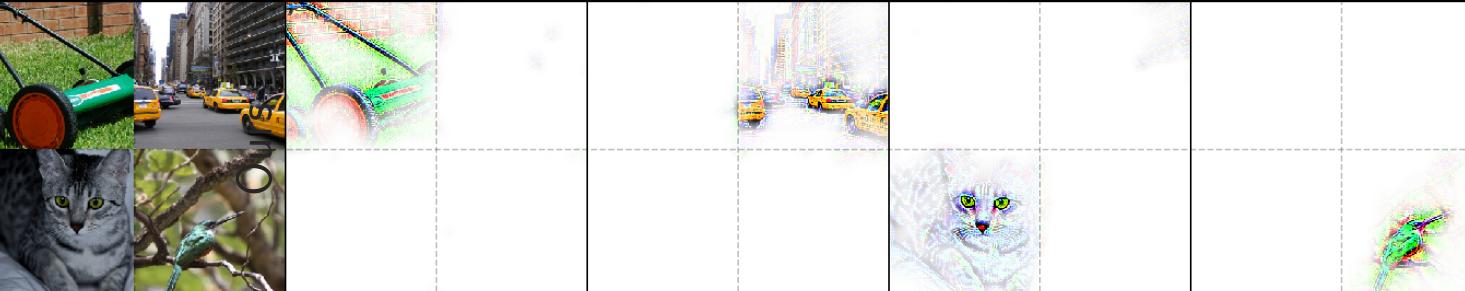
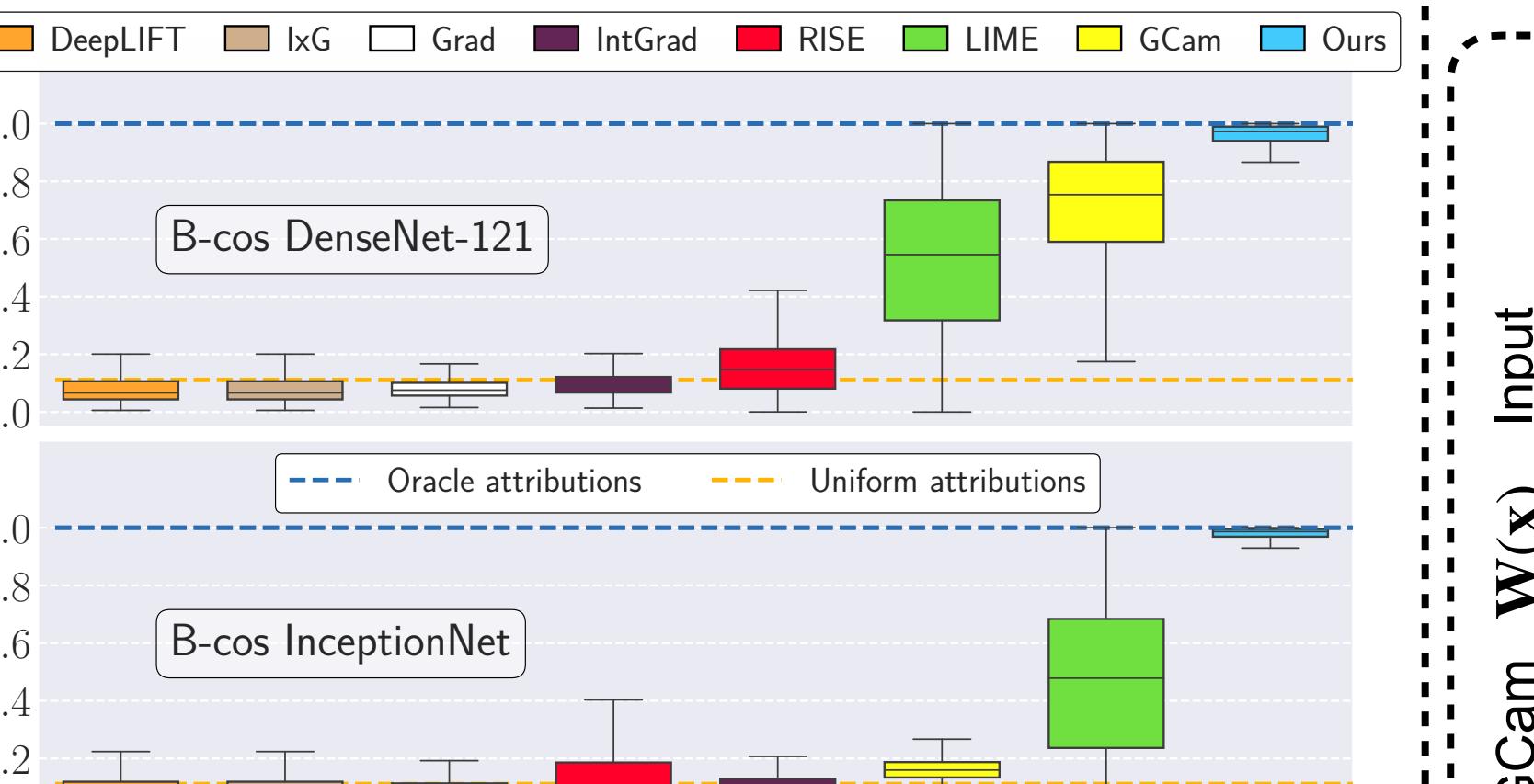


We can also compute linear contributions $\mathbf{s}_c = [\mathbf{W}_{1 \rightarrow L}]_c^T \odot \mathbf{x}$. E.g., here, we show $\Delta = \mathbf{s}_{c_1} - \mathbf{s}_{c_2}$.

References: DeepLIFT (Shrikumar, '17), IntGrad (Sundararajan, '17), RISE (Petsiuk, '18), LIME (Ribeiro, '16), GCam (Selvaraju, '17), ResNet (He, '16), DenseNet (Huang, '17), Inception (Szegedy, '16), VGG (Simonyan, '15), ImageNet (Deng, '09)

Results: interpretable and accurate

Quantitative comparison: Localisation



In contrast to post-hoc methods **inherent explanations** are
(1) model-faithful (2) highly detailed (3) parameter-free

We can explain any neuron in layer l in the same way via $\mathbf{W}_{1 \rightarrow l}(\mathbf{x})$:

"Wheel Neuron"

Explanations for highest actvts.
of neuron 739
in layer 87



B-cos CNNs achieve **comparable accuracy** on ImageNet as baselines

	VGG-11		ResNet-34		DenseNet-121		InceptionNet	
	pre	B-cos	pre	B-cos	pre	B-cos	pre	B-cos
69.0	69.6	73.3	71.7	74.4	73.3	77.3	75.4	
$\Delta = +0.6$		$\Delta = -1.6$		$\Delta = -1.1$		$\Delta = -1.9$		
B-cos DenseNet-121 training⁺							74.4 ($\Delta=0.0$)	