CVPR JUNE 3-7, 2026 DENVER COLORADO

Hidden Monotonicity

Explaining Deep Neural Networks via their DC Decomposition

Jakob Paul Zimmermann[1,2] Georg Loho[2]

TECHNISCHE UNIVERSITÄT BERLIN

[1]TU Berlin

[2]Discrete Geometry Group

Freie Universität Berlin

Why Look for Hidden Monotonicity?

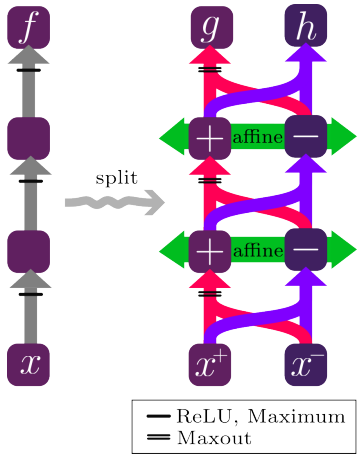
- Attribution asks: **which evidence drove this prediction?**
- Gradients, CAM, LRP, perturbation methods: useful but noisy, and the signs are often considered meaningless.
- Monotone networks are interpretable because evidence only accumulates.
- But monotonicity by design costs expressivity and is hard to apply to pretrained CNNs.

Question

Can we use monotonicity **after training**, without changing the model predictions?

The Split

$$f(x) = g(x) - h(x)$$



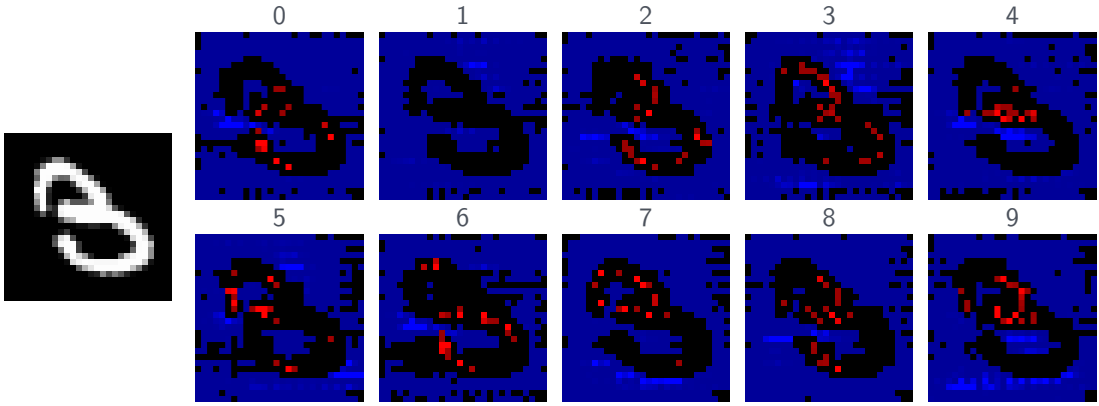
- Constructive layer-by-layer split: g, h are nonneg-weight Maxout networks.
- Works on MLPs, CNNs, ResNets.
- No retraining; no extra weight memory.

The key identity

$$\mathrm{ReLU}(a - b) = \max\{a, b\} - b$$

How do the two streams interact?

To study this we train the difference $g - h$ of two convex networks g and h. Let us consider the overlay of their gradients per class logit.



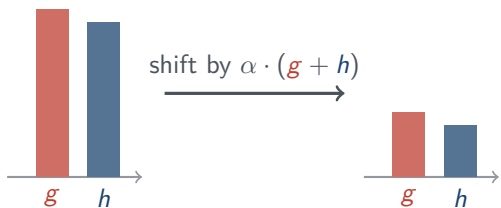
Mitigating the Numerical Obstacle

Problem

Nonnegative split-streams lose the cancellations of the original signed network, so activations and gradients can become very large.

Solution

- Shift both streams simultaneously in forward and backward pass.
- Differences still reconstruct the original network.



Three novel attribution methods

- SplitGrad: the accumulated plain stabilized sensitivities of the split-streams.
- SplitCAM: apply the standard LayerCAM formula with the split-stream activations and gradients.
- SplitLRP: run epsilon-LRP concurrently through the positive and negative split-streams.

How stabilization improves explanations

The stabilization parameter α shifts all stream gradients simultaneously in each layer during the backward pass.

VGG16 on ImageNet-S. SplitGrad measured at layer 26.

	original gradient	*sweet spot*		*raw split-stream gradient*
Method / metric	$\alpha = 0.50$	$\alpha = 0.40$	$\alpha = 0.35$	$\alpha = 0.00$
SplitGrad Point. ↑	**0.898**	0.947	**0.956**	0.401
SplitGrad Attr. Loc. ↑	0.543	0.605	**0.617**	0.449
SplitGrad Max Sens. ↓	0.616	0.440	0.238	**0.028**
SplitGrad Pixel Fl. @20 ↑	1.527	**1.581**	1.448	1.247

Results on ImageNet-S

VGG16 / ImageNet-S. **Bold** = best in column.

Method	Layer	Faithfulness	Localization	Robustness
		Select.↓	Attr. Loc.↑	Max Sens.↓
SplitCAM ($\alpha = 0.4$)	14	**2.727**	0.563	1.012
SplitCAM ($\alpha = 0.4$)	12	4.720	**0.651**	0.389
SplitCAM ($\alpha = 0.4, \mathrm{wta}$)	26	4.711	0.635	0.456
SplitLRP	26	5.168	0.588	**0.282**
Guided Backprop	5	3.448	0.614	0.375
LayerCAM	14	2.792	0.576	2.656
LRP $\gamma = 0.25$	0	3.189	0.615	0.519

Takeaway

- ReLU networks **hide monotone structure**.
- We **expose it without retraining**.
- **SplitCAM, SplitLRP, SplitGrad** turn that structure into better explanations.

Open Questions

- **Optimal DC decompositions:** minimize stream gradients or Newton-polytope complexity.
- **Transformers:** extend SplitCAM and SplitLRP to attention, softmax, and GeLU.

Interested in explainability, geometry, or monotone architectures?

Feel free to contact and exchange!

Jakob Paul Zimmermann
jakob-paul-zimmermann.com

Georg Loho
lohomath.github.io