What Helps—and What Hurts: Bidirectional Explanations for Vision Transformers

Qin Su, Thomas Tie Luo* (presenter)
University of Kentucky

PAKDD, June 9-12, 2026

Central Question

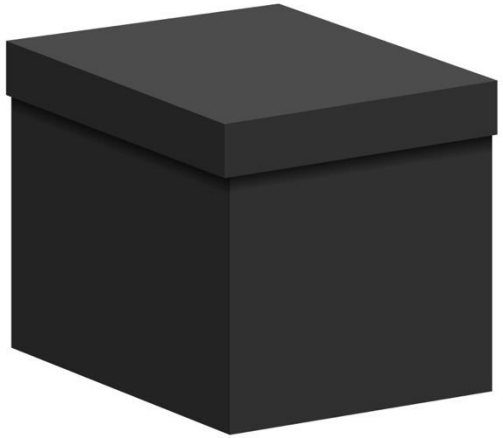
Existing CAM methods answer: Why grasshopper?
BiCAM also answers: Why not acorn/others?

Existing CAM methods answer: Why elephant?
BiCAM also answers: Why not zebra?

and vice versa.

Motivation

- Vision Transformers (ViTs) dominate modern vision systems
  - But largely remain as a “black box”
- Existing XAI mostly identify **supporting** evidence
  - “This image is classified as ____ because …”
- We ask: Do models also **reject competing classes?**
  - *Understanding rejection is equally important*



Related XAI work (ViT focus)

1. **Attention-based**
   - Raw attention only weakly relates to feature importance
   - Attention Rollout / Attention Flow [ACL'20]: aggregate attention over multi-layers but often over-smooth token differences
2. **Gradient-based**
   - Aggregate attention + gradients
   - Beyond Attention [CVPR'21] (based on LRP), Beyond Intuition [TMLR'22]
   - CAM-style: AG-CAM [AAAI'24], TS-CAM [ICCV'21], CDAM [TMLR'24]
3. **Shapley-based**
   - ViT-Shapley [ICLR'23]: estimate patch-level Shapley values by training an explainer network
   - Estimation only speeds up inference; training is time-consuming (~19 hrs over 10 classes only) and dataset-specific

- **Common gap:** missing negative evidence

Method - Bidirectional Class Activation Mapping (BiCAM)

- Design principle 1: ***What* explanation signals** should be collected?

Attention ($A^{(l)}$)	information routing
Value projections ($V^{(l)}$)	patch-level features
Gradients ($\partial y_c / \partial o_{cls}^{(l)}$)	class sensitivity

- Design principle 2: ***Where*** to collect these signals?
  - Existing methods: recursively aggregate all layers
    - Attribution maps are noisy, over-smoothed, and less class-discriminative
  - Observation: Class-discriminative info concentrates in **deeper** layers [Raghu et al.'21][Li et al.'22]
  - We use the last ℓ layers only

Empirically choose
$\ell = 2/3\ L$

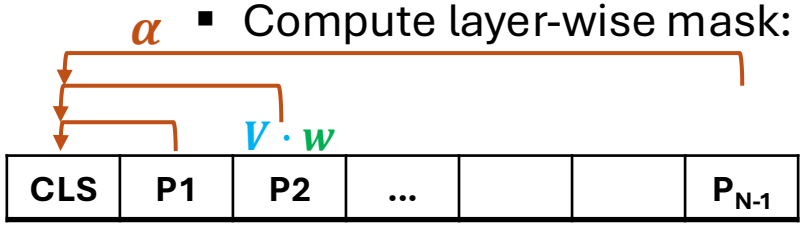
ViT-B: 16, ViT-L: 24, ViT-H: 32

Ablation on layer window ℓ and temperature T (ViT-B/16, VOC 2012).

	Layer window ℓ (T=2)				*Temperature T (ℓ=8)*		
	ℓ=12	ℓ=8	ℓ=4	ℓ=1	T=1	**T=2**	T=3
Pix.Acc	0.8505	**0.8559**	0.8498	0.8413	0.8402	**0.8559**	0.8492
IoU	0.2840	**0.3700**	0.2093	0.1491	0.1051	**0.3700**	0.2060
F1	0.4043	**0.5104**	0.3173	0.2277	0.1711	**0.5104**	0.3140
Prec.	0.5980	**0.6095**	0.6765	0.5343	0.6401	0.6095	**0.6789**
Rec.	0.3949	**0.5863**	0.2649	0.1824	0.1144	**0.5863**	0.2589
Faith.	**0.4746**	0.4626	0.4570	0.3499	0.4472	0.4626	**0.4900**

How BiCAM works

1. **Forward pass: extract attention + values**
   From each of the last ℓ layers and each head h :
   - Extract attention $A^{(l)}_{h,0,:}$: CLS-to-patch attention *before softmax*, $l \in [L-\ell+1, L]$
   - Apply softmax w/ temperature T to soften attention: $\alpha^l_h = \text{softmax}(A^{(l)}_{h,0,:}/T) \in R^N$
   - Extract value projection $V^{(l)}_h$: contains all the token embeddings ($R^{N\times d}$)
2. **Backward pass: compute class-specific gradients**
   - Backprop class score y_c to obtain $w^{(l)}_c = \frac{\partial y_c}{\partial o^{(l)}_{cls}}$
     - $o^{(l)}_{cls}$ is the [CLS] token embedding ($o^{(l)}_{cls} \in R^d$)
3. **Construct attribution maps: (value × gradient) × attention**
   - Compute layer-wise mask:



$$\text{mask}^{(l)} = \sum_{h=1}^{H} \left((V^{(l)}_h \cdot w^{(l)}_c) \odot \alpha^{(l)}_h \right)$$

1st: dot product
2nd: element-wise

Weight each token by class sensitivity (R^N)

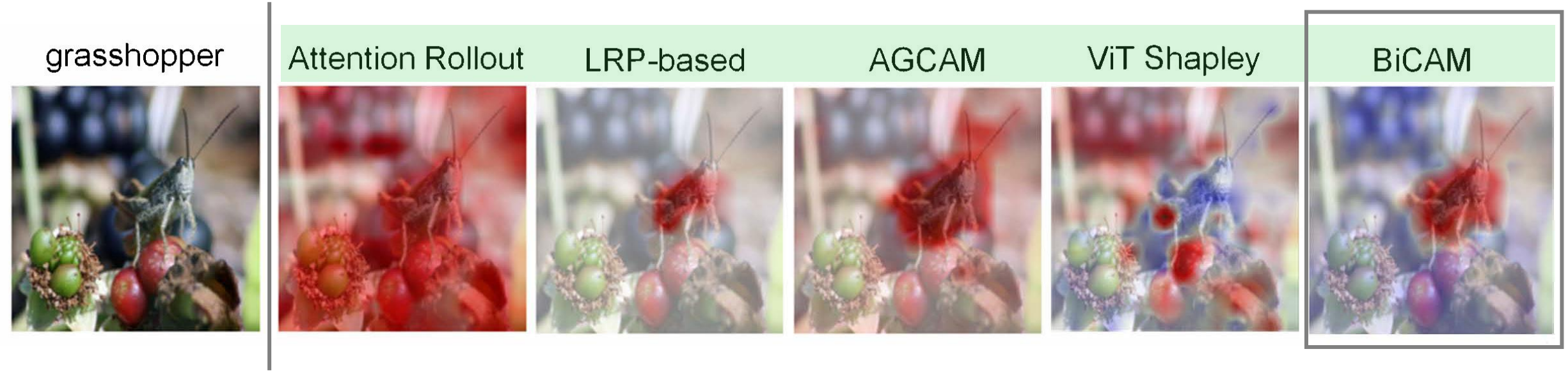
attention weight from CLS to each token (R^N)

No ReLU clipping throughout the workflow

4. **Final heatmap:**
   - Aggregate masks across ℓ layers: $mask = \sum_{l\in[L-\ell+1,L]} mask^{(l)} \in R^N$
   - Remove [CLS] token (R^{N-1}) → Reshape to 2-D ($\sqrt{N-1}, \sqrt{N-1}$) → Upsample to [H, W]

Qualitative Result: Single-Object scenario

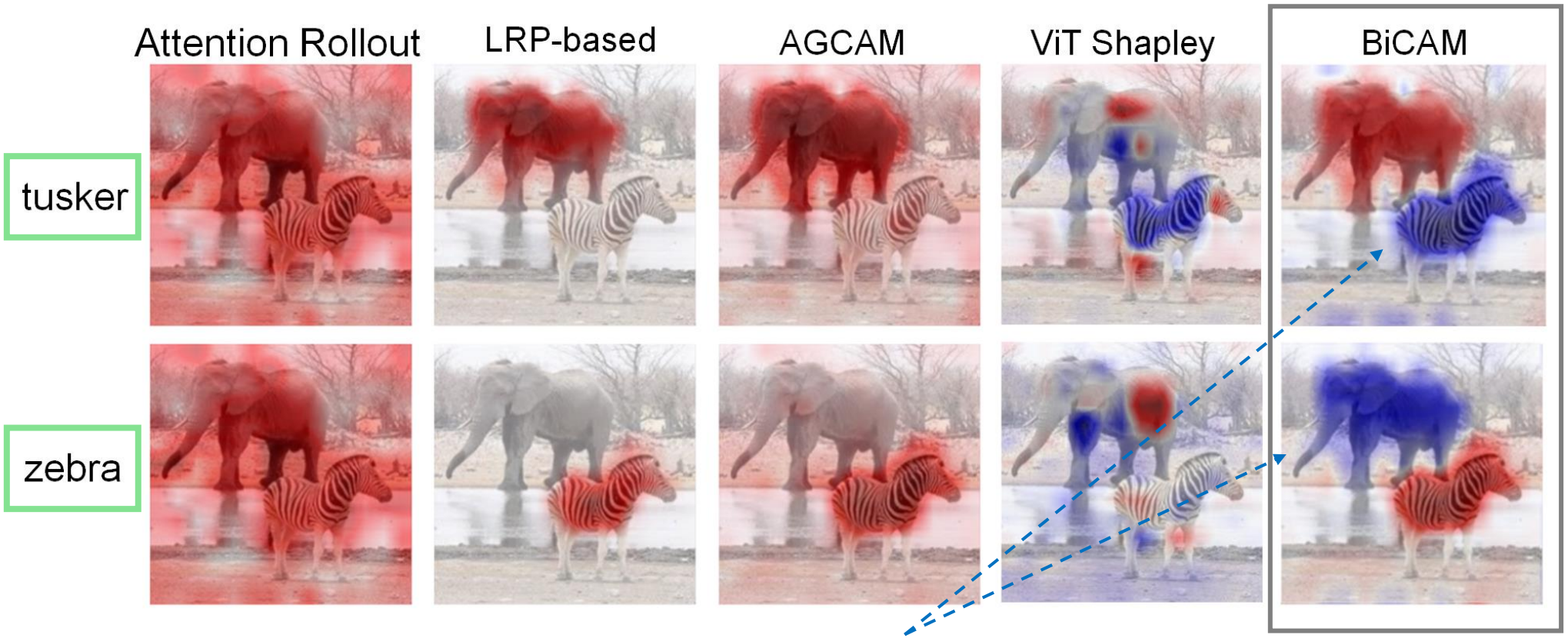
ViT-B/16 on ImageNet



- **Red: supportive** evidence → grasshopper
- **Blue: suppressive** evidence → background and distracting objects

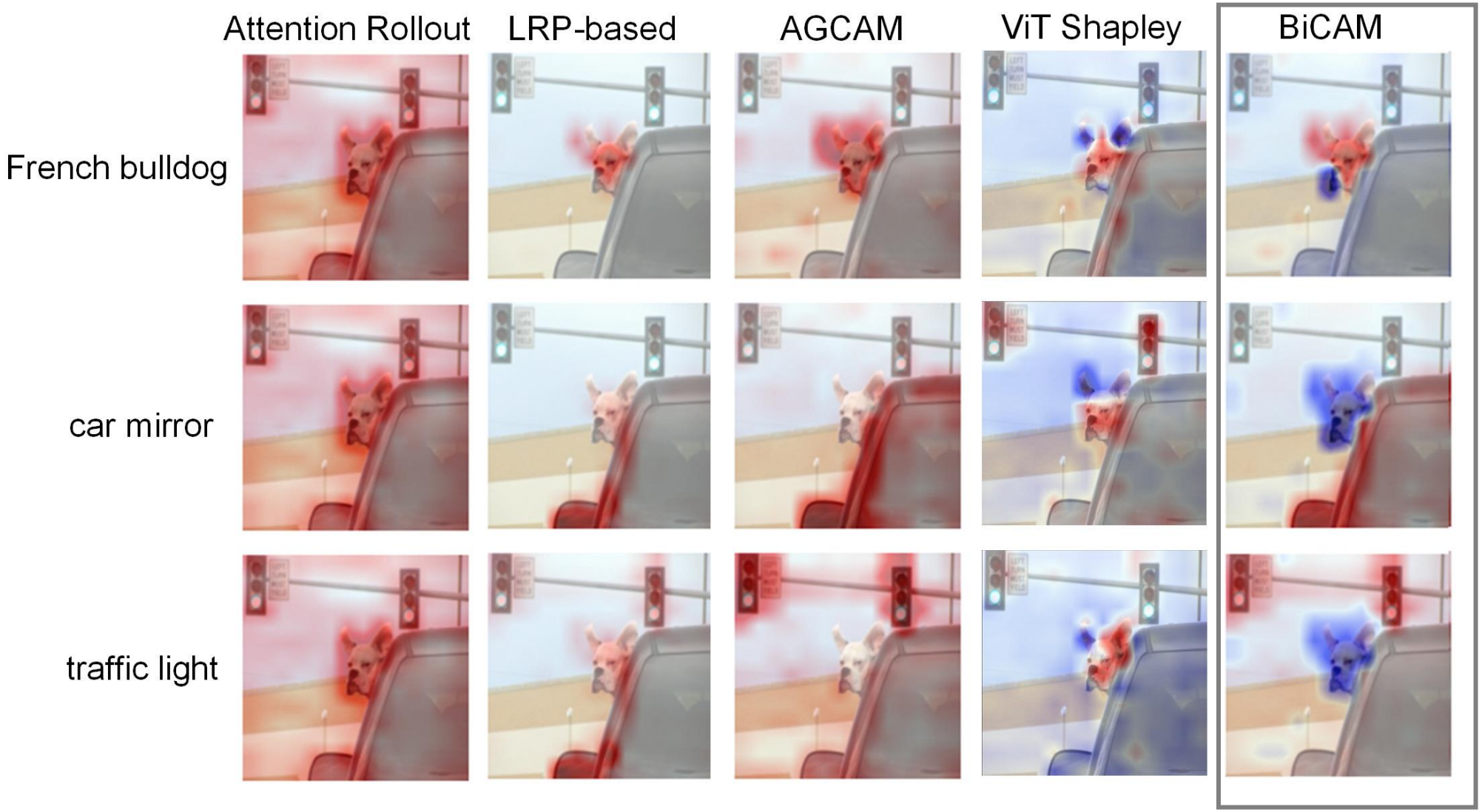
Qualitative Results: Multi-Object scenario

ViT-B/16 on COCO



- Negative attribution identifies **competing objects**
- Baselines: diffuse, non-specific maps
- But: *Do **two-object** scenarios particularly favor such **binary** methods?*

More than Two objects

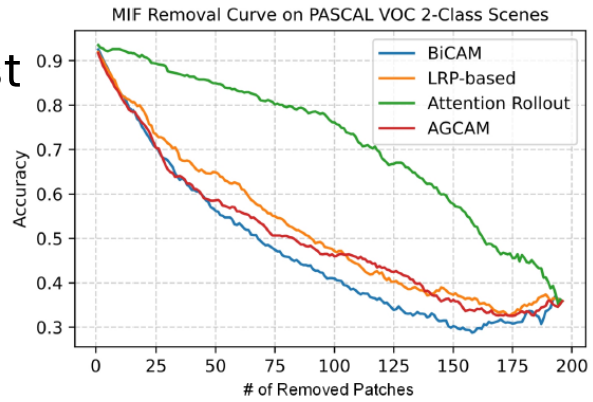


Quantitative Result: Faithfulness

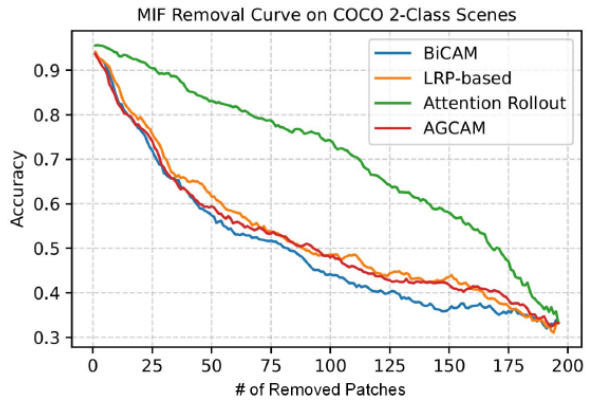
- *Feature perturbation*
  - **MIF** (Most Important Feature): remove features from the most important to the least
    - Good explanation: steep drop → plateau
  - **LIF** (Least Important Feature): remove from the least important to the most
    - Good explanation: slow drop → plunge
- *Faithfulness score* = $\mathbf{AUC_{LIF} - AUC_{MIF}}$

	Attn Rollout	LRP	AGCAM	BiCAM
ImageNet				
LIF↑	0.4739	0.5140	0.5298	**0.5478**
MIF↓	0.2053	0.1736	**0.1607**	0.1654
Faith↑	0.2685	0.3404	0.3691	**0.3824**
VOC 2-class				
LIF↑	0.8623	0.9231	0.9280	**0.9313**
MIF↓	0.7142	0.5201	0.5010	**0.4686**
Faith↑	0.1481	0.4030	0.4270	**0.4626**
COCO 2-class				
LIF↑	0.8856	0.9307	0.9376	**0.9407**
MIF↓	0.7052	0.5327	0.5235	**0.4997**
Faith↑	0.1805	0.3980	0.4141	**0.4410**

MIF: Lower is better

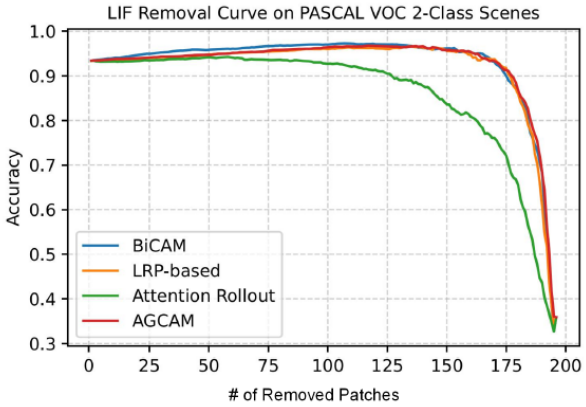


(a) VOC - MIF removal

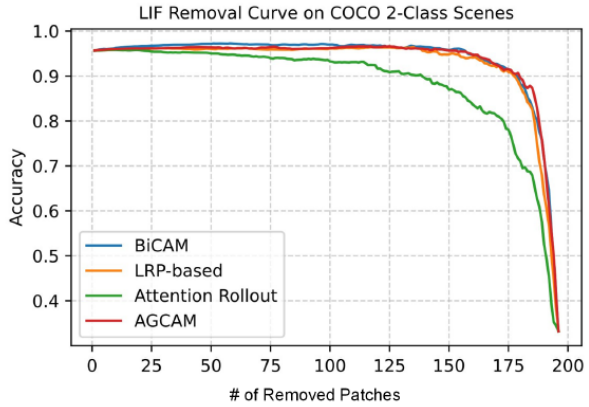


(b) COCO - MIF removal

LIF: Higher is better



(c) VOC - LIF removal



(d) COCO - LIF removal

Classification Accuracy

Quantitative Result: Localization

- ImageNet (LOC), PASCAL VOC, COCO
- Pixel accuracy, IoU, F1, precision, recall

- BiCAM performs best overall
- AGCAM performs second best

- BiCAM (neg.) has **no direct counterpart** (baselines produce +ve only); yet this table places it at a **disadvantageous** position because **class labels** push models to focus on learning features for **target** objects, not background or non-target.
- Despite this, the numbers show that the negative map captures semantically meaningful competing regions *rather than random noise*.

- ViT-Shapley is excluded because its amortized explainer could not scale up to 1,000-class evaluation.

Method	Pix. Acc.	IoU	F1	Prec.	Rec.
ImageNet-1k					
Attn Rollout	0.6209	0.3597	0.4893	0.7326	0.4657
LRP-based	0.5863	0.2029	0.3055	**0.9110**	0.2176
AGCAM	**0.7341**	0.5212	0.6515	0.8299	0.6276
BiCAM	0.6253	**0.5419**	**0.6624**	0.5901	**0.9288**
VOC 2012					
Attn Rollout	0.8105	0.0686	0.1153	0.2604	0.1191
LRP-based	0.8420	0.1677	0.2464	0.4918	0.2021
AGCAM	**0.8561**	0.3561	0.4926	0.5932	0.5502
BiCAM (Pos.)	0.8559	**0.3700**	**0.5104**	**0.6095**	**0.5863**
BiCAM (Neg.)	0.7705	0.2588	0.3786	0.3642	0.5779
COCO 2017					
Attn Rollout	0.8314	0.1332	0.2023	0.3276	0.2526
LRP-based	0.8523	0.1162	0.1871	0.4274	0.1835
AGCAM	**0.8740**	0.2807	0.3996	0.4210	**0.6279**
BiCAM (Pos.)	0.8707	**0.2971**	**0.4191**	**0.5535**	0.5154
BiCAM (Neg.)	0.8487	0.1141	0.1724	0.2711	0.2074

Computational Efficiency

Inference speed / memory

Table 4: Runtime comparison (ViT-B/16, RTX 4090, ImageNet-1k).

Method	ms/img	MB/img	Train time
Attn Rollout	23.3	0.15	None
LRP	134.6	97.2	None
ViT-Shapley	10.1*	–	19 hrs*
AGCAM	20.0	0.22	None
BiCAM	**16.0**	0.24	None

*ImageNette (10 classes), measured on RTX 2080 Ti [7].

- 8.4x faster (60 img/s) than LRP-based methods
- GPU memory-efficient
- No training of any network (unlike ViT-Shapley, which needs to train a surrogate and an explainer)

- ***What design contributed to efficiency?***
  - Single forward-backward pass
  - Selective layer aggregation strategy

Application: Adversarial Detection

- **Hypothesis: Adversarial attacks disturb the balance between +ve and -ve evidence.**
- We define *Positive-to-Negative Ratio* (PNR) :

$$PNR = \frac{\sum_i ReLU(M_i)}{\sum_i ReLU(-M_i) + \epsilon}$$

M_i : attribution value of patch i

- Define PNR perturbation: $\Delta PNR = PNR_{adv} - PNR_{clean}$
- If ΔPNR exhibits a pattern under different attacks, it can serve as an indicator to detect attacks

Results on COCO:

Detection performance using *ΔPNR*

Attack	ΔPNR	std	AUROC	AUPR	Threshold
Clean	0.00	N/A	-	-	-
PGD	+0.57	0.31	0.781	0.749	0.248
C&W	+0.79	0.37	0.842	0.808	0.332
MI-FGSM	+0.46	0.27	0.764	0.731	0.219
Avg.	+0.61	0.32	0.796	0.763	0.266

Benchmark attacks

Derived by maximizing *Youden's J statistic*

Simple and lightweight:

- no dedicated detection algorithm
- no supervised training (as opposed to *adversarial training*)

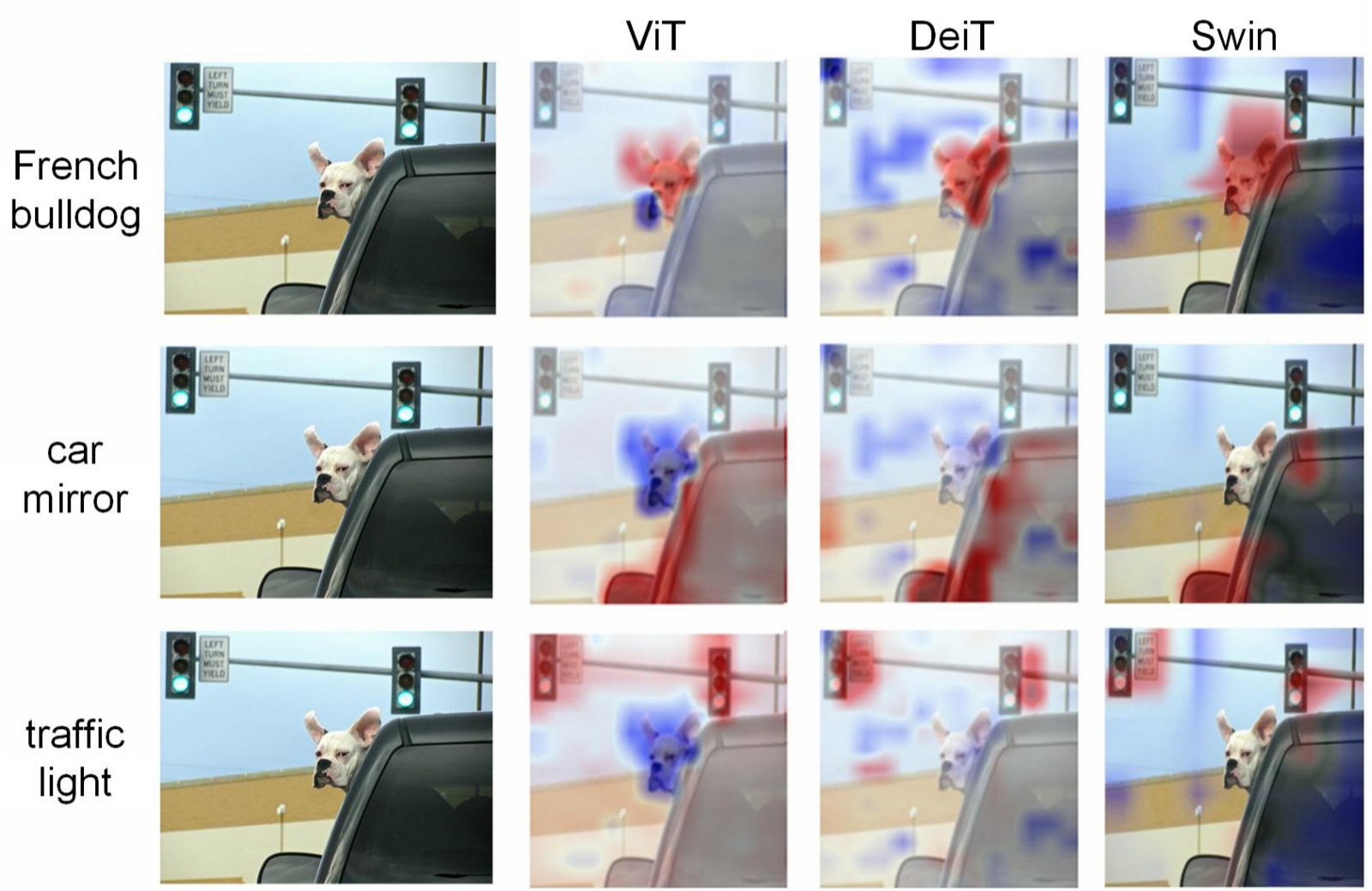
Generalizes to other vision models

BiCAM requires only:

- Attention
- Value projections
- Gradients

Demonstration under the most challenging scenario: Multi-object ➔

- DeiT: similar
- Swin: dissimilar



Conclusion

- A dedicated bidirectional CAM for ViT.
- Reveals supportive and suppressive evidence.
- Strong localization, faithfulness, and efficiency.
- Enables new downstream apps (e.g., adversarial detection) and generalizes to other ViTs.

- Paper available at: https://arxiv.org/abs/2603.01605

Why is bidirectional explanation useful?

- **Contrastive, augmented explanation**
  - Answers both: “Why this class?” + “Why not others?”
  - Closer to human reasoning; more convincing / boosts confidence
  - Suggests that models ***actively reject alternatives***