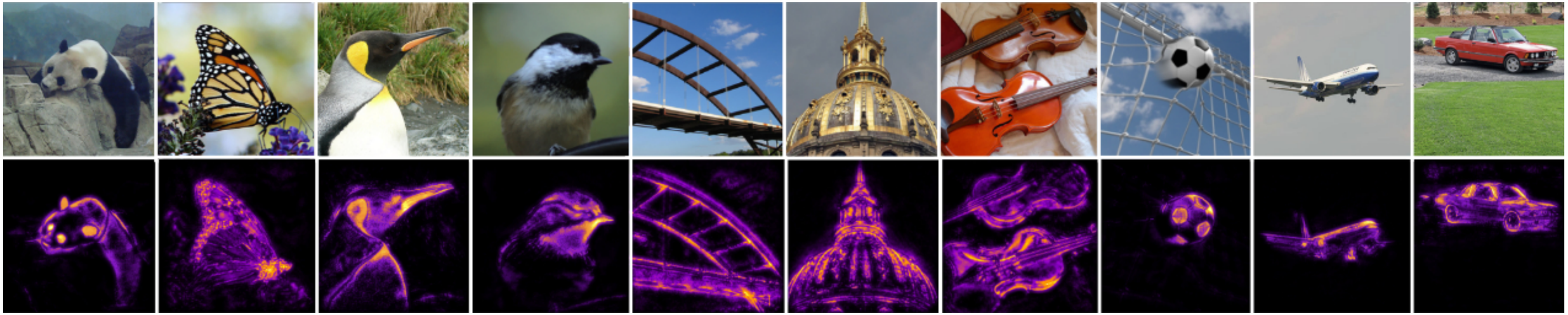
# DAVE : Distribution-aware Attribution via ViT Gradient Decomposition

*arxiv: https://arxiv.org/abs/2602.06613*

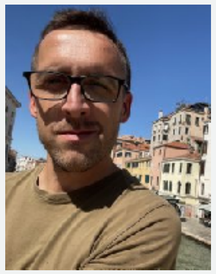



**Adam Wróbel**
GMUM, Jagiellonian University

**Siddhartha Gairola**
MPI for Informatics

**Jacek Tabor**
GMUM, Jagiellonian University

**Bernt Schiele**
MPI for Informatics



**Bartosz Zieliński**
GMUM, Jagiellonian University

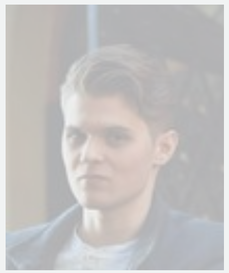
**Dawid Rymarczyk**
GMUM, Jagiellonian University

MAX PLANCK INSTITUTE FOR INFORMATICS

SIC Saarland Informatics Campus

# AVE: Distribution-aware Attribution via ViT Gradient Decomposition

*arxiv: https://arxiv.org/abs/2602.06613*

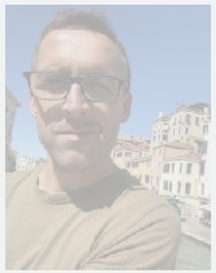



Adam Wróbel
GMUM, Jagiellonian University

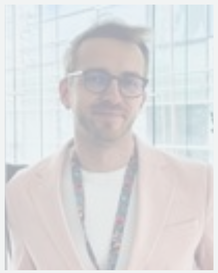
**Siddhartha Gairola**
MPI for Informatics

Jacek Tabor
GMUM, Jagiellonian University

Bernt Schiele
MPI for Informatics



Bartosz Zieliński
GMUM, Jagiellonian University



Dawid Rymarczyk
GMUM, Jagiellonian University

# Issues with existing Attribution Methods for Vision Transformers

Vision Transformers [1] or ViTs are highly prevalent in computer vision.

However getting good explanations still remains a challenge:

✦ gradient- and attention-based explanations exhibit structured, architecture-induced artifacts

# Issues with existing Attribution Methods for Vision Transformers

Vision Transformers [1] or ViTs are highly prevalent in computer vision.

However getting good explanations still remains a challenge:

- ✦ gradient- and attention-based explanations exhibit structured, architecture-induced artifacts
- ✦ leading either to unstable pixel-level attributions or to coarse patch-level explanations
- ✦ thus lack fine-grained visual evidence

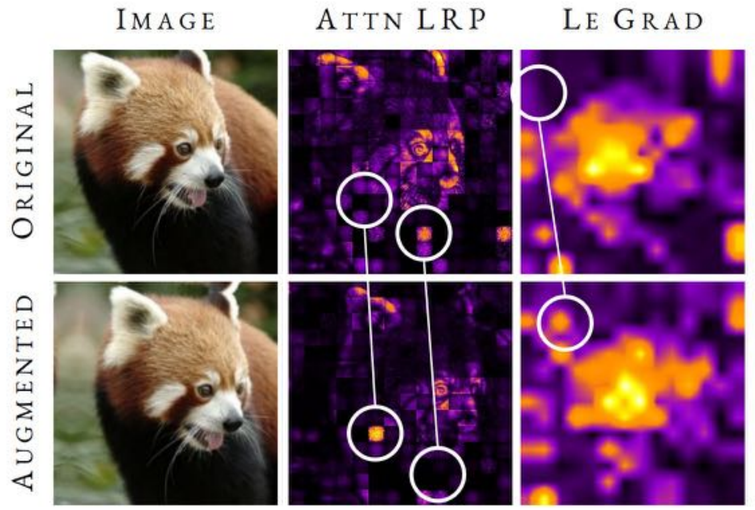
# Issues with existing Attribution Methods for Vision Transformers

Vision Transformers [1] or ViTs are highly prevalent in computer vision.

However getting good explanations still remains a challenge:

- ✦ gradient- and attention-based explanations exhibit structured, architecture-induced artifacts
- ✦ leading either to unstable pixel-level attributions or to coarse patch-level explanations
- ✦ thus lack fine-grained visual evidence

Under small augmentations
(5° rotation, 20px horizontal and 8px vertical shift)



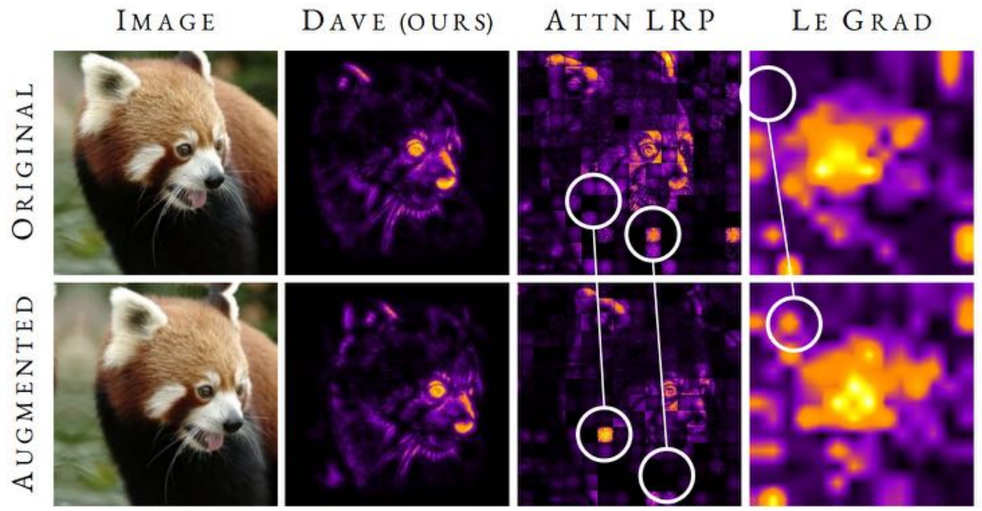
# Issues with existing Attribution Methods for Vision Transformers

Vision Transformers [1] or ViTs are highly prevalent in computer vision.

However getting good explanations still remains a challenge:

- ✦ gradient- and attention-based explanations exhibit structured, architecture-induced artifacts
- ✦ leading either to unstable pixel-level attributions or to coarse patch-level explanations
- ✦ thus lack fine-grained visual evidence

Under small augmentations
(5° rotation, 20px horizontal and 8px vertical shift)



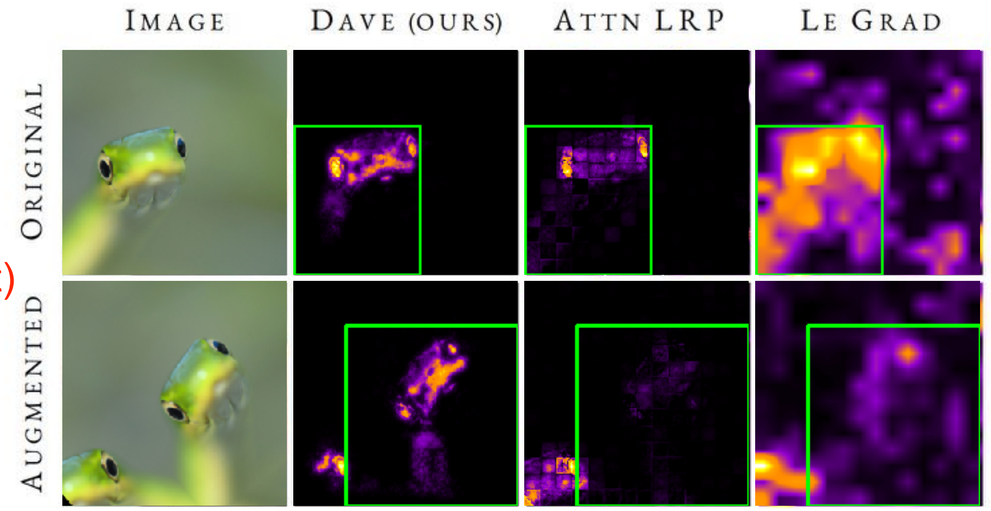
# Issues with existing Attribution Methods for Vision Transformers

Vision Transformers [1] or ViTs are highly prevalent in computer vision.

However getting good explanations still remains a challenge:

- ✦ gradient- and attention-based explanations exhibit structured, architecture-induced artifacts
- ✦ leading either to unstable pixel-level attributions or to coarse patch-level explanations
- ✦ thus lack fine-grained visual evidence

Under stronger augmentations
(-40° rotation, 50px horizontal and -30px vertical shift)



# What does DAVE solve?

We fix a persistent issue in ViT explainability: **unstable, artifact-heavy pixel attributions**.

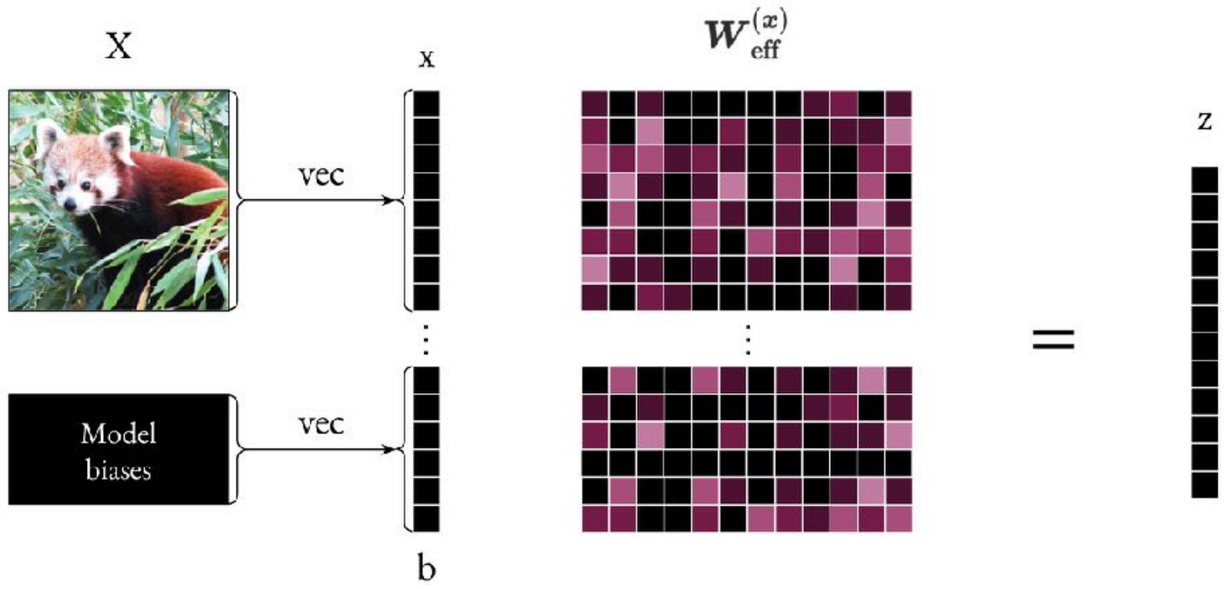
DAVE yields:

- ✓**fine-grained pixel-level maps**
- ✓**without patch-grid saliency.**

# DAVE - The Method

DAVE interprets an explanation (or attribution) as the **stable** and **locally equivariant** effective transformation that a ViT applies to its input.

*ViT is a Dynamic Linear Matrix* $\boldsymbol{z} = \mathrm{ViT}(\boldsymbol{X}) = \boldsymbol{W}_{\mathrm{eff}}^{(\boldsymbol{x})}[\boldsymbol{x}|\boldsymbol{b}]$




- If every base layer is of a form:

$$\mathrm{vec}(\boldsymbol{Z}) = \boldsymbol{W}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{b}$$

- Then the whole ViT is dynamic linear:

$$\mathrm{ViT}(\boldsymbol{X}) = \boldsymbol{W}_{\mathrm{eff}}^{(\boldsymbol{x})}[\boldsymbol{x}|\boldsymbol{b}]$$

# DAVE - The Method

Model each ViT layer as an input-dependent linear operator *L*(X) applied to X.

*(self-attention, layer-norm, residual connections, mlp blocks, activation functions)*

Then the input-gradient decomposes into:

(1) effective transformation *L*(X)

(2) operator variation (how *L* changes w.r.t. X)

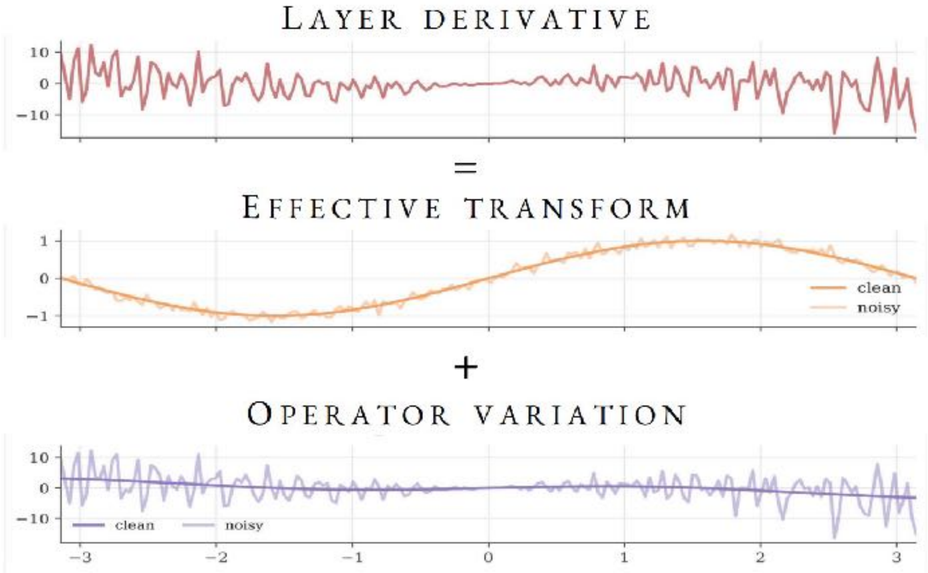
$$\underbrace{D_{\boldsymbol{X}} F}_{\substack{\text{layer} \\ \text{derivative}}} = \underbrace{L(\boldsymbol{X})}_{\substack{\text{effective} \\ \text{transformation}}} + \underbrace{\left(\left(D_{\boldsymbol{X}} L(\boldsymbol{X})(\cdot)\right) \boldsymbol{X}\right)}_{\substack{\text{operator} \\ \text{variation}}}$$

# DAVE - The Method

## ⚠ Why gradients break ?

Operator variation can amplify tiny perturbations

→ high-frequency junk in attributions.




$$\underbrace{D_{\boldsymbol{X}}F}_{\text{layer derivative}} = \underbrace{L(\boldsymbol{X})}_{\text{effective transformation}} + \underbrace{\left((D_{\boldsymbol{X}}L(\boldsymbol{X})(\cdot))\,\boldsymbol{X}\right)}_{\text{operator variation}}$$

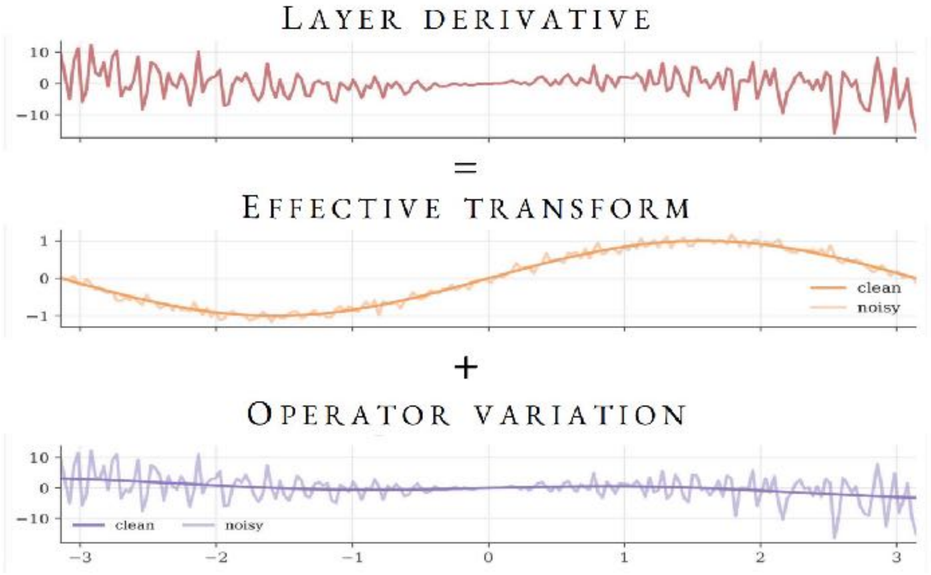
# DAVE - The Method

## ⚠ Why gradients break ?

Operator variation can amplify tiny perturbations

→ high-frequency junk in attributions.

DAVE drops this term and keeps the effective transformation as a cleaner attribution operator.




$$\underbrace{D_{\boldsymbol{X}}F}_{\text{layer derivative}} = \underbrace{L(\boldsymbol{X})}_{\text{effective transformation}} + \underbrace{\left((D_{\boldsymbol{X}}L(\boldsymbol{X})(\cdot))\,\boldsymbol{X}\right)}_{\text{operator variation}}$$

# DAVE - The Method

**Effective weights and how to find them efficiently**

- **Detach** all dynamic matrices on the forward pass (attention, GELU multiplier, etc.)
- Compute “detached” **backward pass**
- Like: CoDA, B-cos

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W}(\boldsymbol{x})\boldsymbol{x}$$

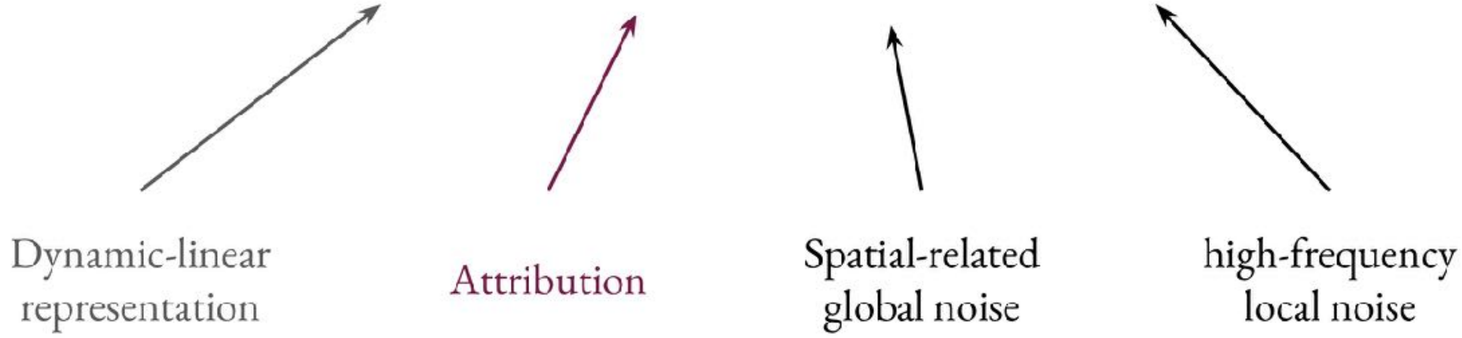
$$\boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x}) = \underbrace{\boldsymbol{W}(\boldsymbol{x})}_{\text{Our ViT matrix}} + \underbrace{\sum_{i=1}^{d} \boldsymbol{x}_i \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{x}_i}(\boldsymbol{x})}_{\text{JVP}}$$

$$\underbrace{D_{\boldsymbol{X}}F}_{\substack{\text{layer} \\ \text{derivative}}} = \underbrace{L(\boldsymbol{X})}_{\substack{\text{effective} \\ \text{transformation}}} + \underbrace{\left((D_{\boldsymbol{X}}L(\boldsymbol{X})(\cdot))\,\boldsymbol{X}\right)}_{\substack{\text{operator} \\ \text{variation}}}$$

# DAVE - The Method

We base our method on a **working hypothesis:** for a well-trained model

$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{C}^{\star}(\boldsymbol{x}) + \boldsymbol{n}_{\tau}(\boldsymbol{x}) + \boldsymbol{n}_{\epsilon}(\boldsymbol{x})$$




where:

$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{W}_{\text{eff}}^{(\boldsymbol{x})}\boldsymbol{x}$$

# DAVE - The Method

We base our method on a **working hypothesis:** for a well-trained model

$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{C}^{\star}(\boldsymbol{x}) + \boldsymbol{n}_{\tau}(\boldsymbol{x}) + \boldsymbol{n}_{\epsilon}(\boldsymbol{x})$$

Dynamic-linear representation

Attribution

Spatial-related global noise

high-frequency local noise

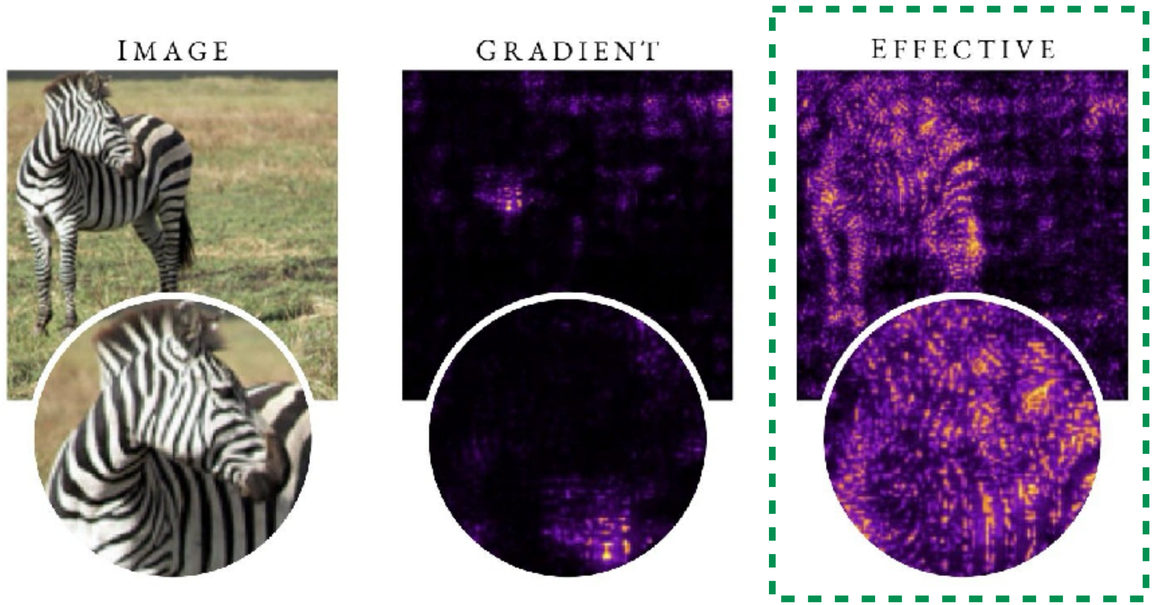
where:

$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{W}_{\text{eff}}^{(\boldsymbol{x})} \boldsymbol{x}$$

# DAVE - The Method

## Remove grid artifacts

Even the effective transformation can carry architecture-induced grid patterns.

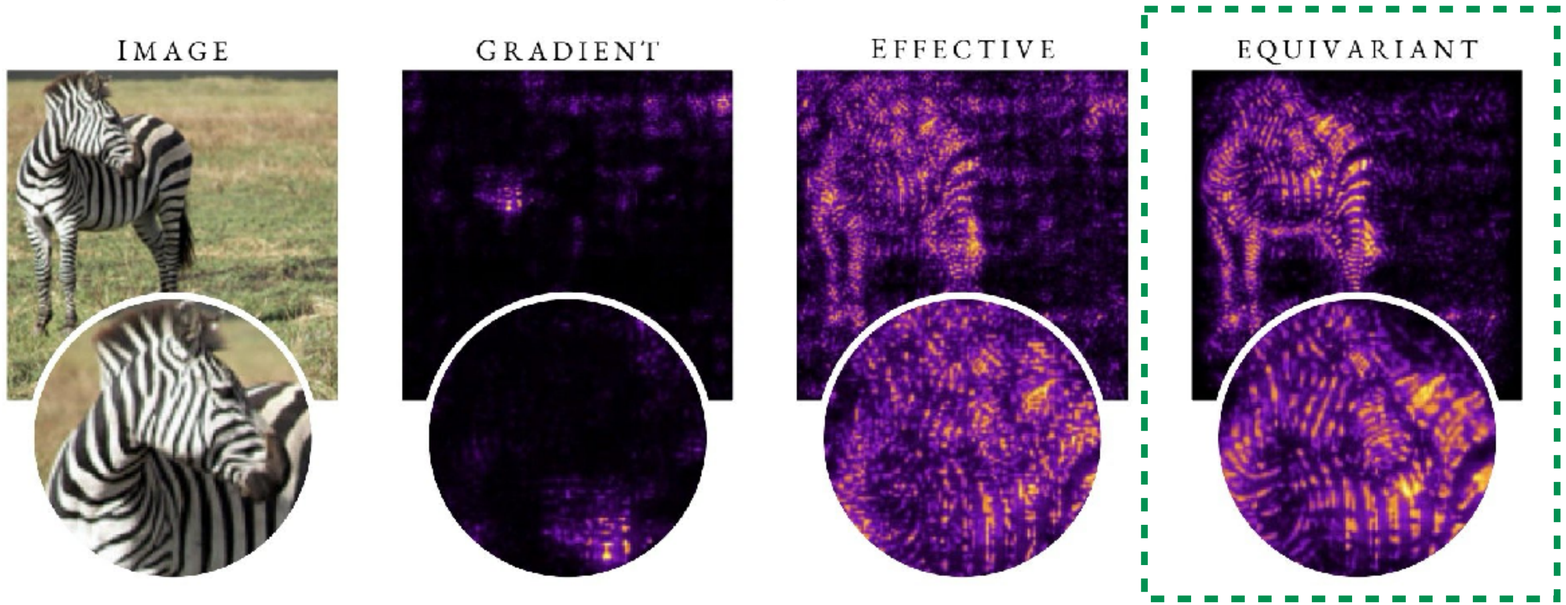


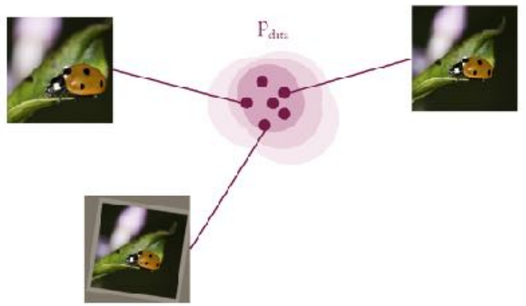
# DAVE - The Method

## Remove grid artifacts

Even the effective transformation can carry architecture-induced grid patterns.

DAVE filters them by enforcing **local equivariance: under small spatial transforms, the attribution must transform consistently.**



# DAVE - The Method

DAVE filters them by enforcing **local equivariance: under small spatial transforms, the attribution must transform consistently.**

- Global noise is spatially **"fixed"**:

$$\mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}(\boldsymbol{n}_{\tau}(\tau(\boldsymbol{x})))] = 0$$

$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{C}^{\star}(\boldsymbol{x}) + \boldsymbol{n}_{\tau}(\boldsymbol{x}) + \boldsymbol{n}_{\epsilon}(\boldsymbol{x})$$

Dynamic-linear representation

Attribution

Spatial-related global noise

high-frequency local noise

where:

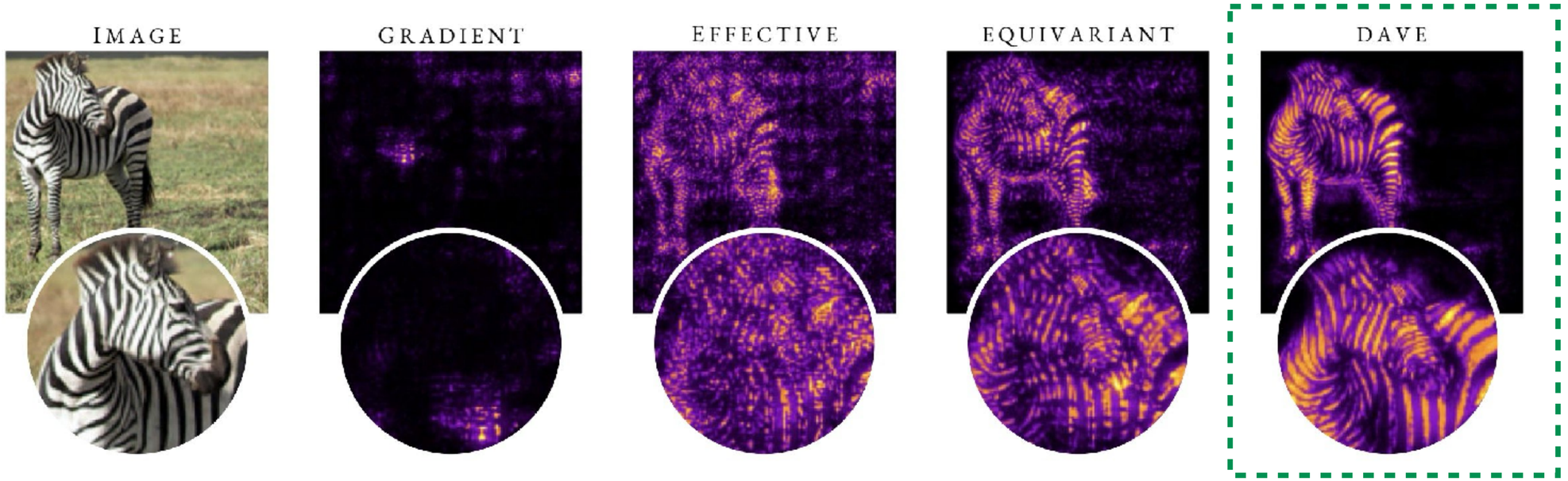
$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{W}_{\text{eff}}^{(\boldsymbol{x})}\boldsymbol{x}$$

# DAVE - The Method

## Stabilize (low-pass)

DAVE adds low-pass filtering by **averaging** the equivariant effective transformation under **small input perturbations** (Gaussian smoothing in expectation).

This removes components unstable to tiny input changes. (See below fig., last column.)



# DAVE - The Method

DAVE adds low-pass filtering by **averaging** the equivariant effective transformation under **small input perturbations** (Gaussian smoothing in expectation).

**removed by low-pass filtering**

$$C(x) = C^{\star}(x) + n_{\tau}(x) + n_{\epsilon}(x)$$

Dynamic-linear representation

Attribution

Spatial-related global noise

high-frequency local noise

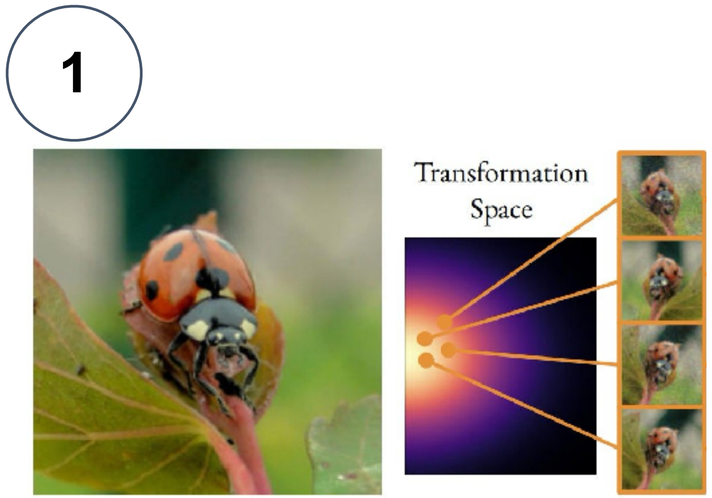
where:

$$C(x) = W_{\text{eff}}^{(x)} x$$

# DAVE - The Method

## 🔧 The pipeline

1. Sample small spatial transforms + noise,

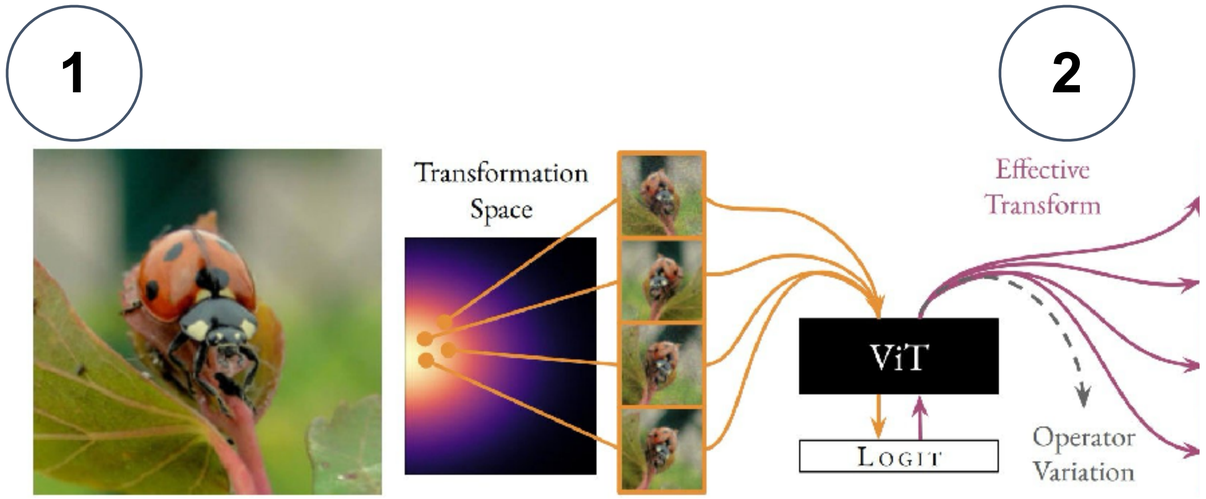


# DAVE - The Method

## 🔧 The pipeline

1. Sample small spatial transforms + noise,

2. compute effective transformation (conditioned forward blocks gradients through conditioning),



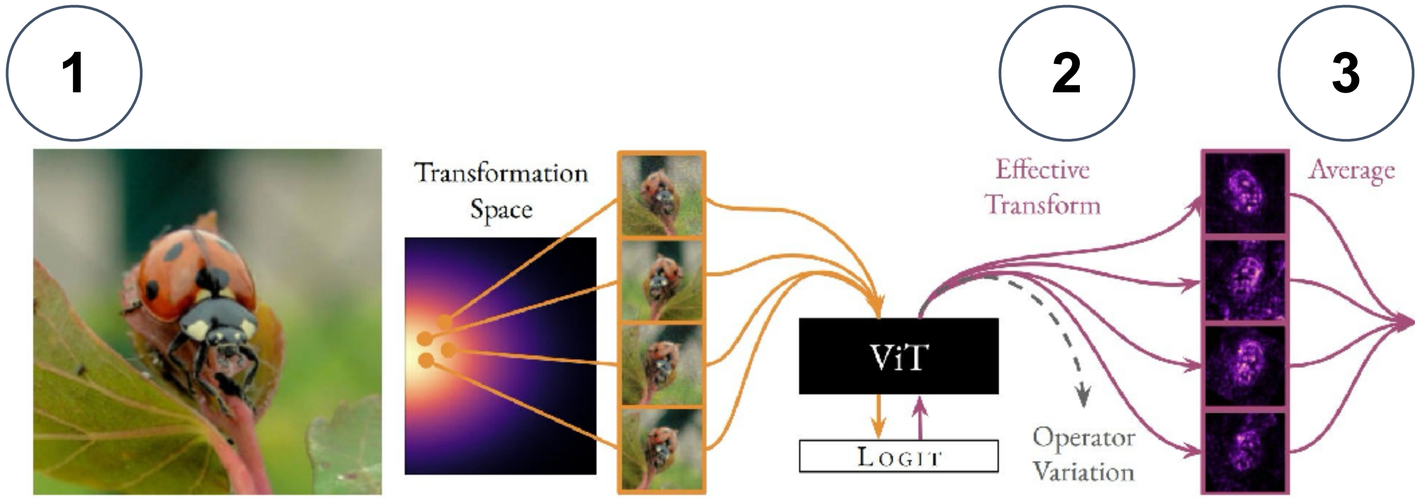
# DAVE - The Method

## 🔧 The pipeline

1. Sample small spatial transforms + noise,

2. compute effective transformation (conditioned forward blocks gradients through conditioning),

3. inverse-transform & average,



# DAVE - The Method

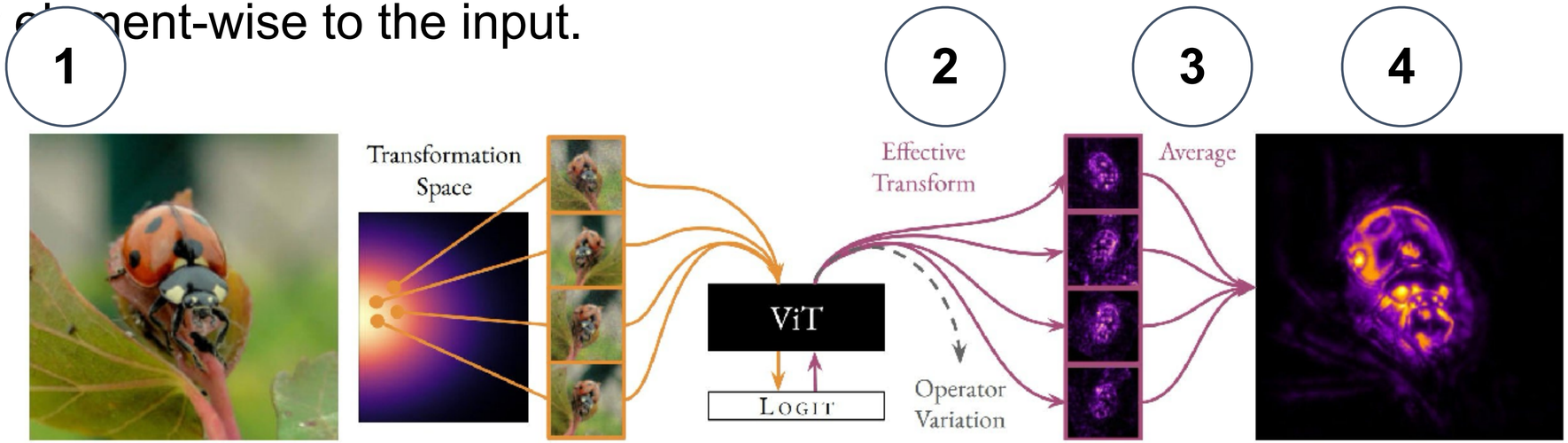
## 🔧 The pipeline

1. Sample small spatial transforms + noise,

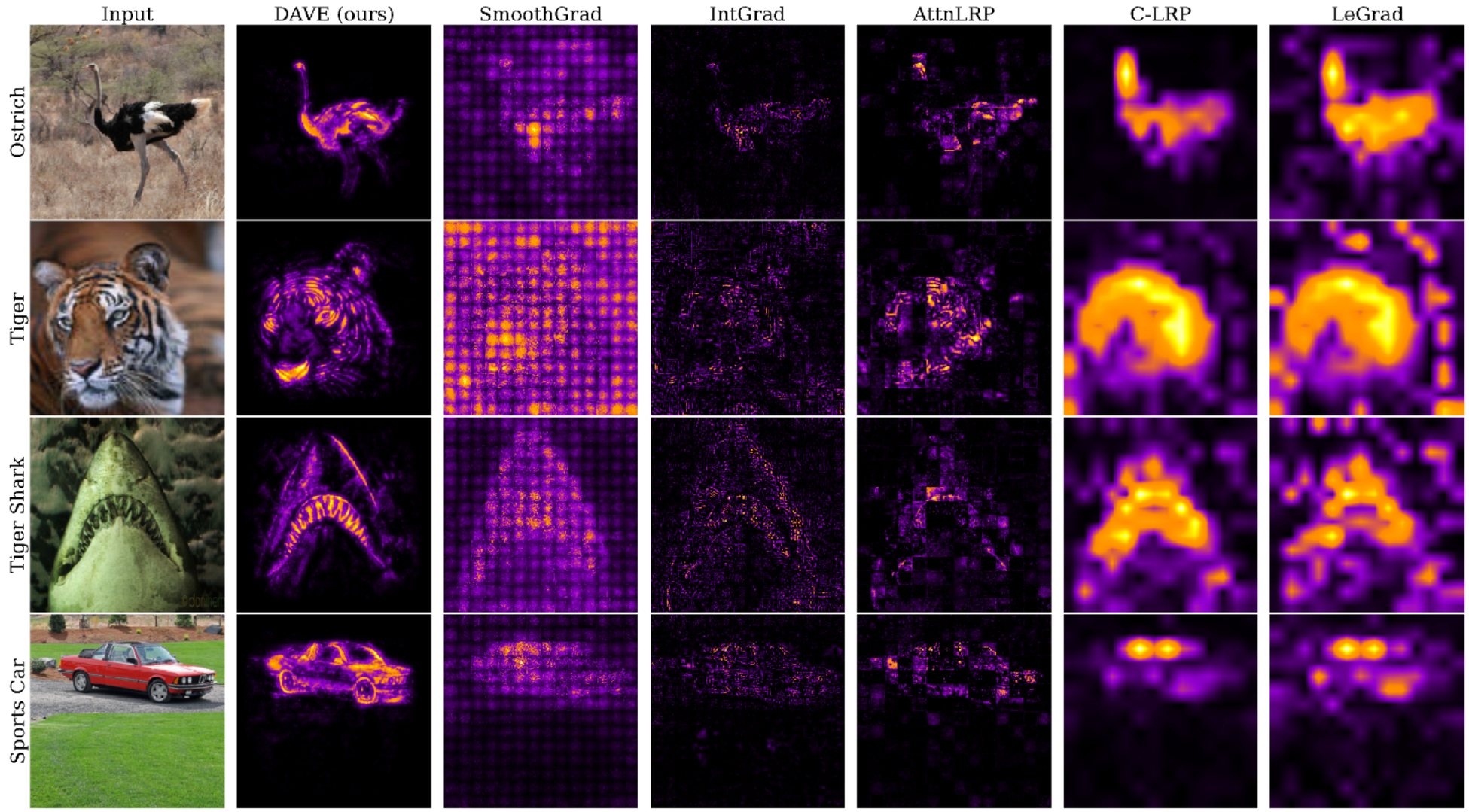
2. compute effective transformation (conditioned forward blocks gradients through conditioning),

3. inverse-transform & average,

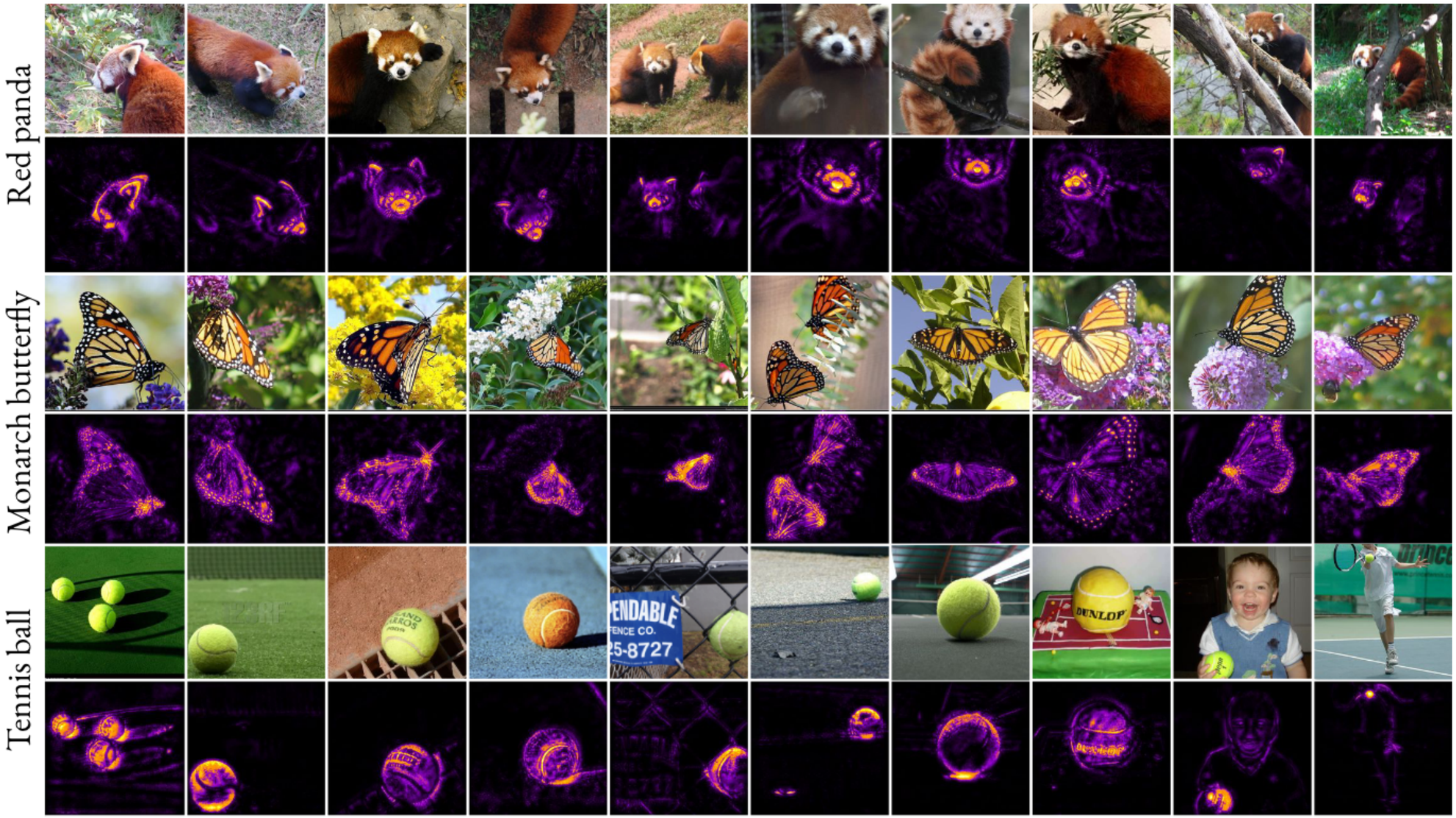
4. then apply element-wise to the input.



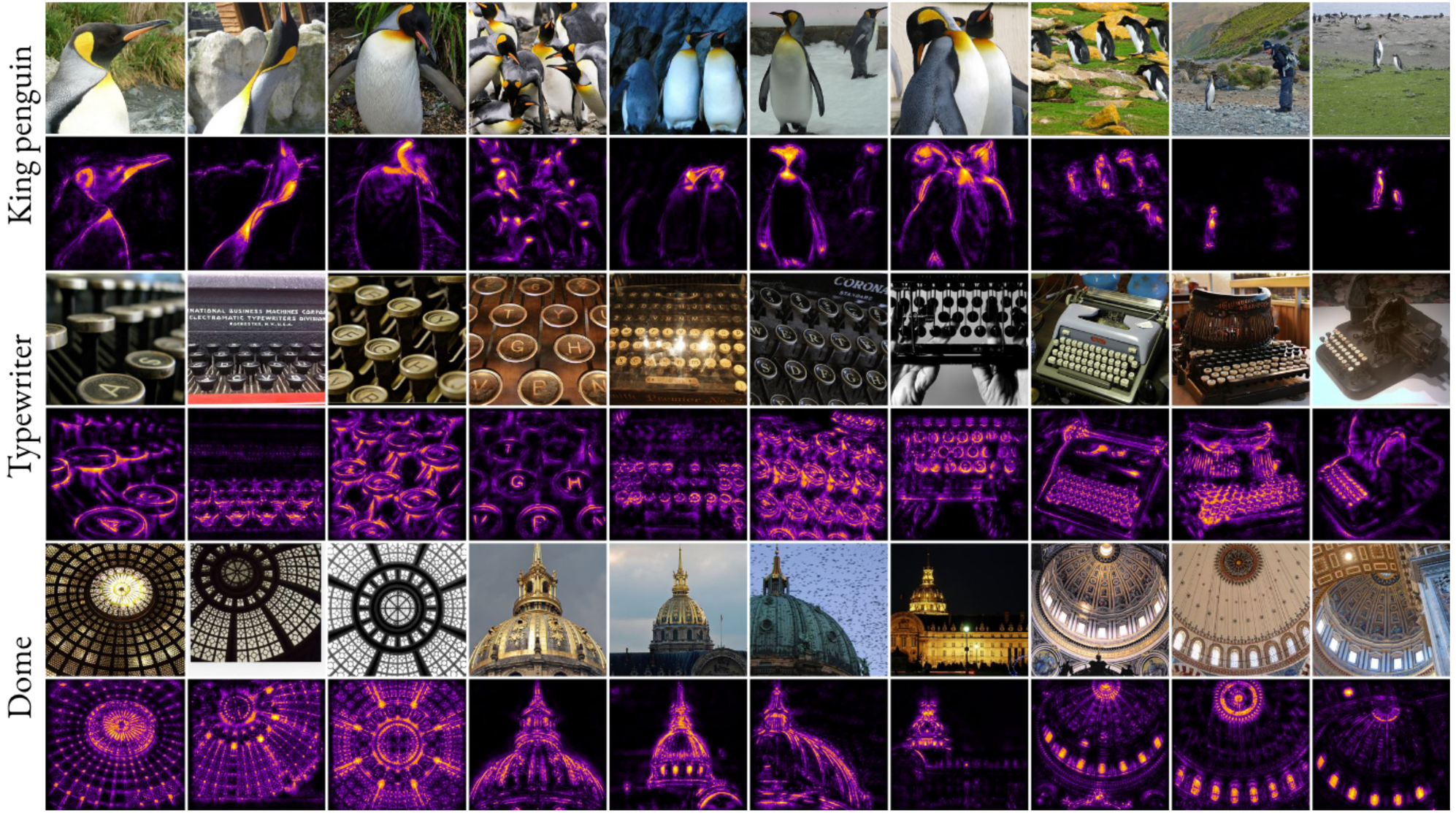
# Qualitative Results: DAVE vs existing popular methods



# Qualitative Results: class consistency of DAVE attributions



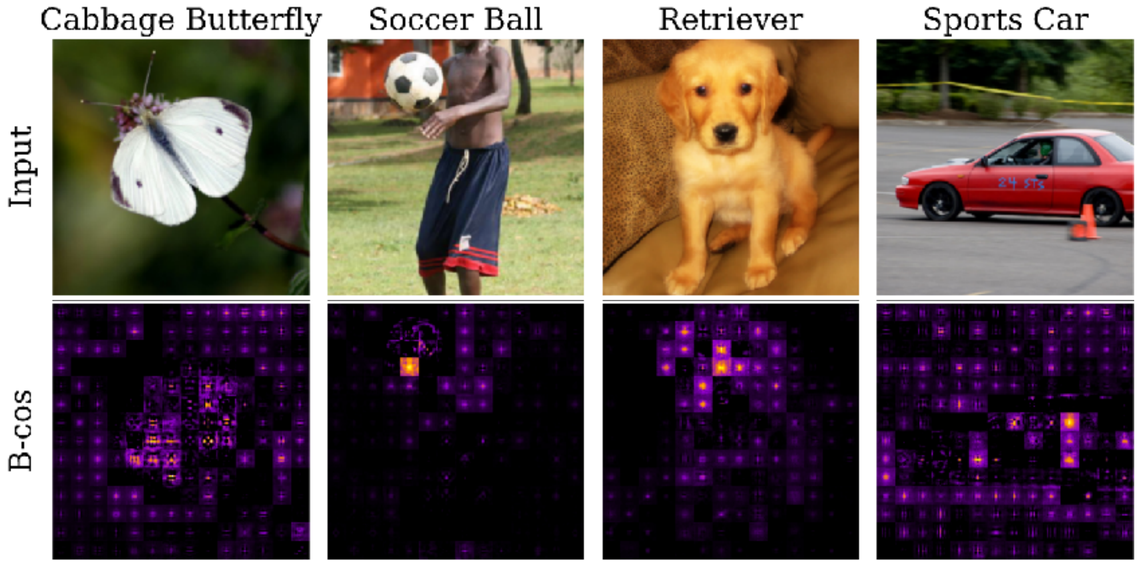
# Qualitative Results: class consistency of DAVE attributions



# Qualitative Results: DAVE improves inherently-interpretable B-cos Attributions

B-cos ViTs rely on a **conv-stem** for good explanations,

- it is not ideal since most modern ViT architectures do not rely on a conv-stem,
- it can also introduce unwanted inductive biases and move away from a purely transformer based architecture.

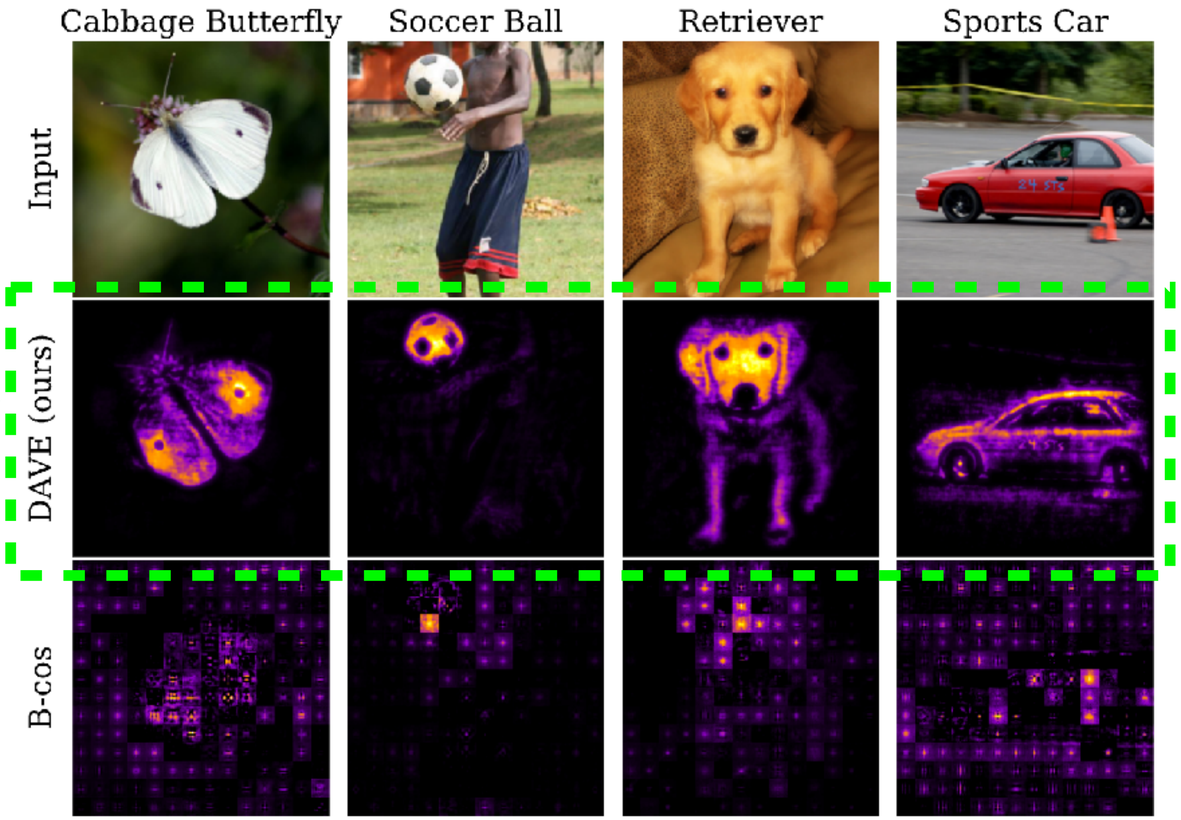


# Qualitative Results: DAVE improves inherently-interpretable B-cos Attributions

B-cos ViTs rely on a **conv-stem** for good explanations,

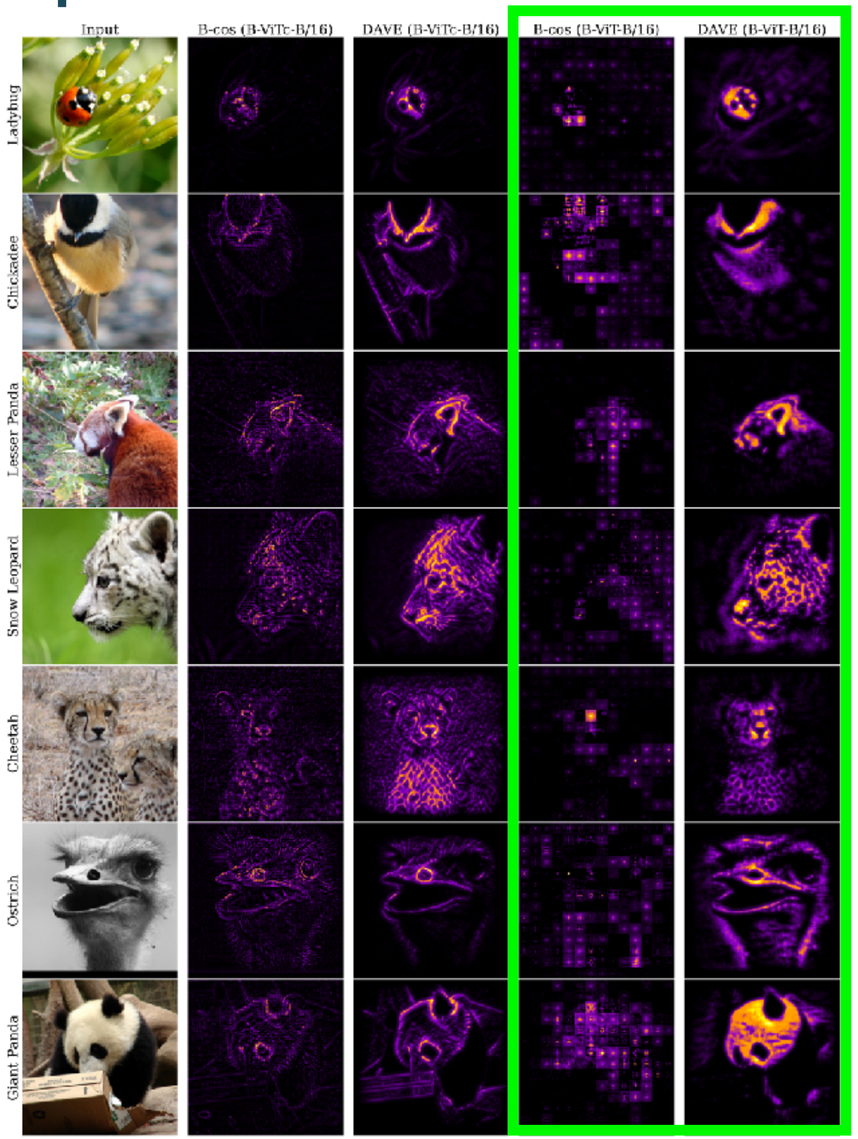
- it is not ideal since most modern ViT architectures do not rely on a conv-stem,
- it can also introduce unwanted inductive biases and move away from a purely transformer based architecture.

**✓DAVE, alleviates this reliance on a conv-stem.**

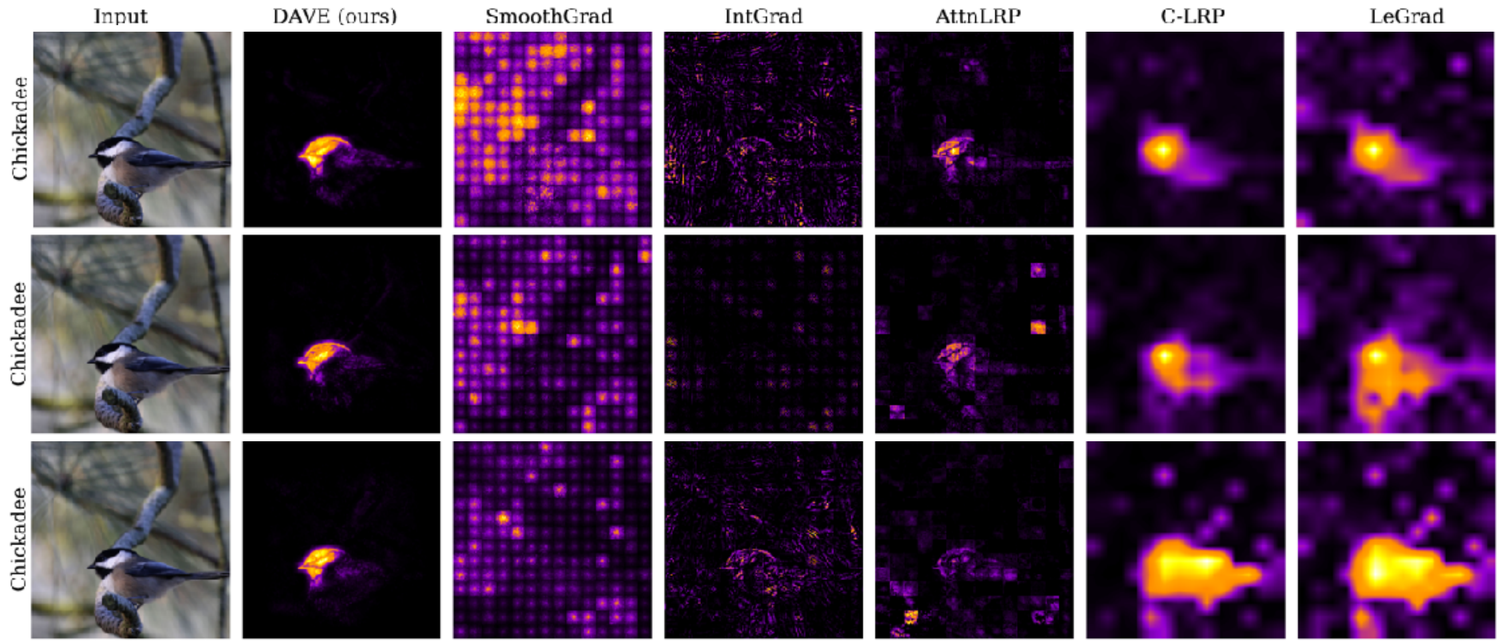


# Qualitative Results: DAVE improves inherently-interpretable B-cos Attributions

Can even be argued that the attributions on pure B-cos ViTs (w/o conv-stem) are nicer (at least qualitatively) ?



# Qualitative Results: consistency across diverse pre-training



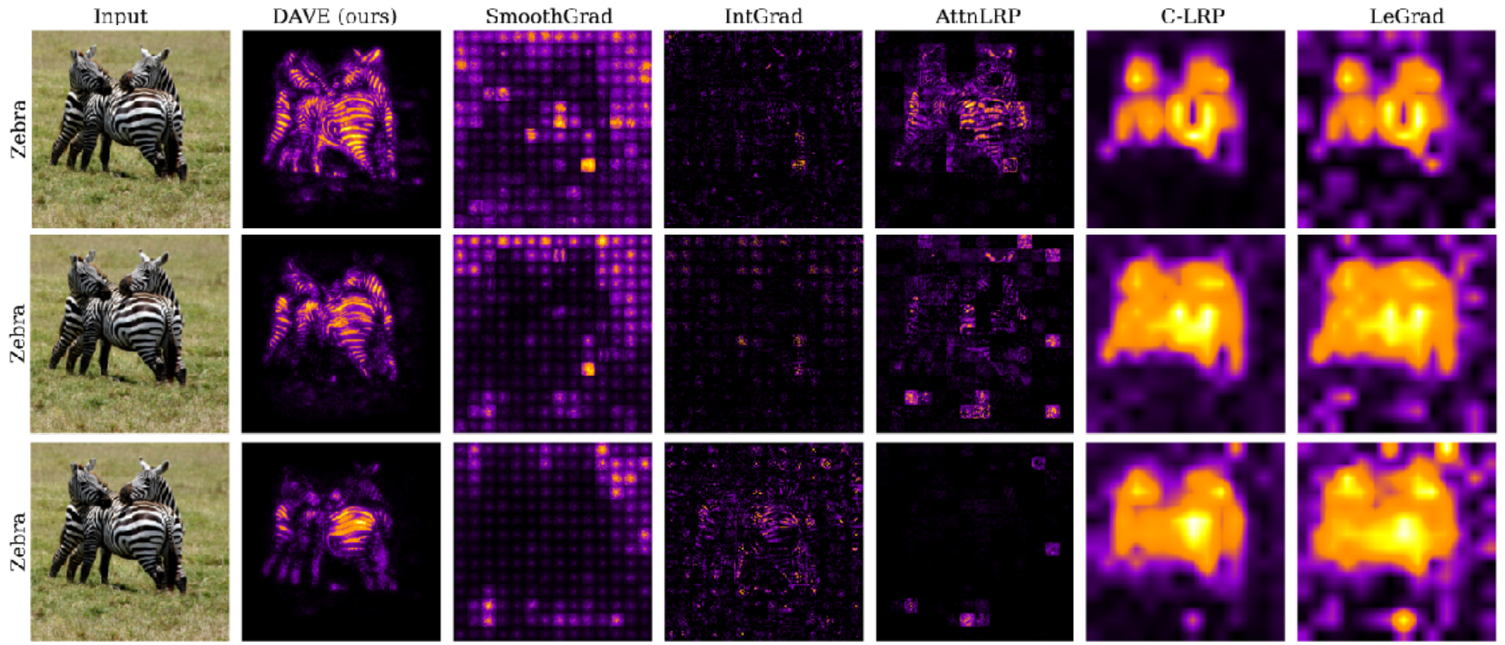

Deit-III-ViT-B/16

Deit-ViT-B/16

ViT-B/16

✓DAVE, yields consistent attributions across pre-trained backbones.

# Qualitative Results: consistency across diverse pre-training




Deit-III-ViT-B/16

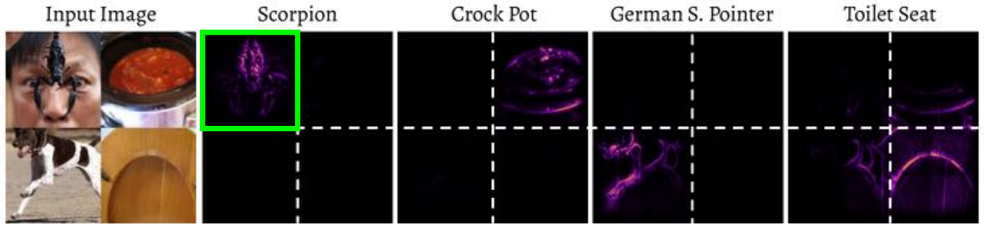
Deit-ViT-B/16

ViT-B/16

✓DAVE, yields consistent attributions across pre-trained backbones.

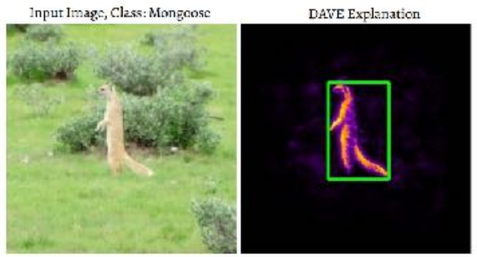
# Quantitative Results: localisation via Grid PG and Energy PG

| | GridPG (%) | | | |
|---|---|---|---|---|
| Method | ViT-B | DeiT-B | D-III-B | DINO-B |
| I×G | 32.67 | 30.25 | 30.01 | 33.28 |
| IntGrad | 39.86 | 36.11 | 31.68 | 36.98 |
| S-Grad | 34.27 | 30.18 | 31.48 | 33.13 |
| LeGrad | 47.71 | 42.58 | 34.62 | 28.96 |
| A-LRP | 58.40 | 54.63 | 53.84 | 37.49 |
| C-LRP | 54.98 | 55.47 | 52.27 | 49.99 |
| DAVE (ours) | **60.19** | **63.52** | **65.76** | **51.33** |
| Δ | **+1.79** | **+8.05** | **+11.92** | **+1.35** |

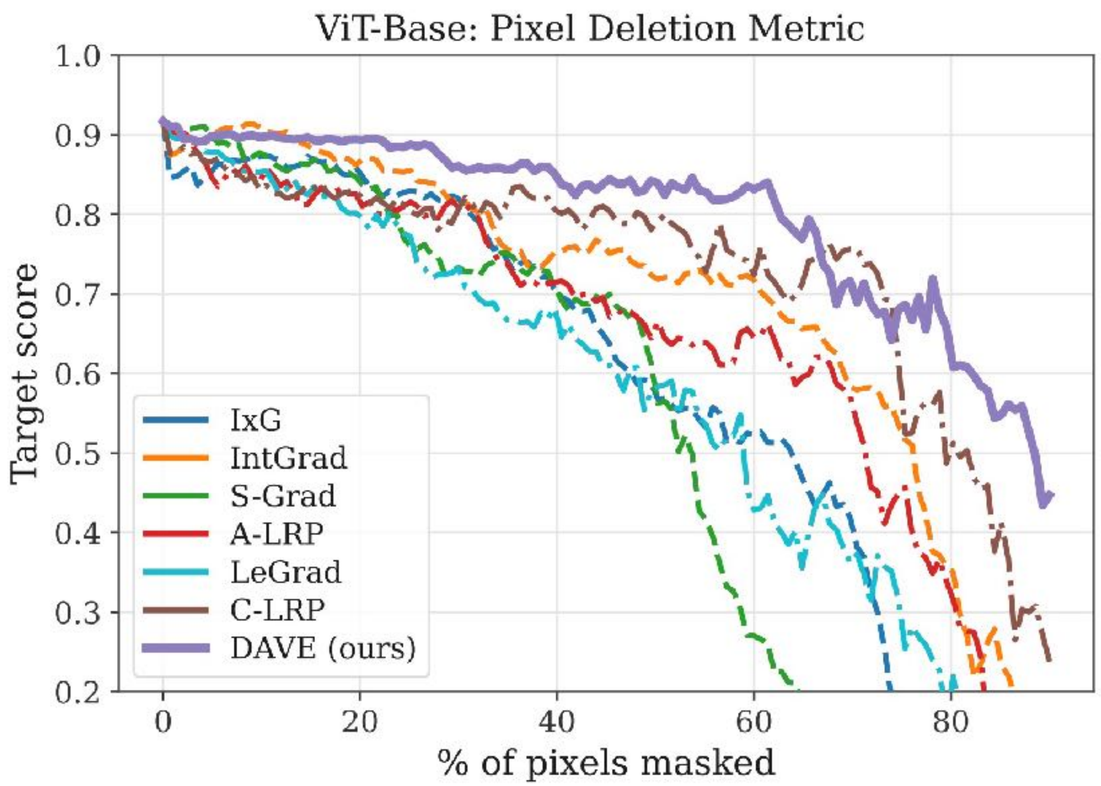


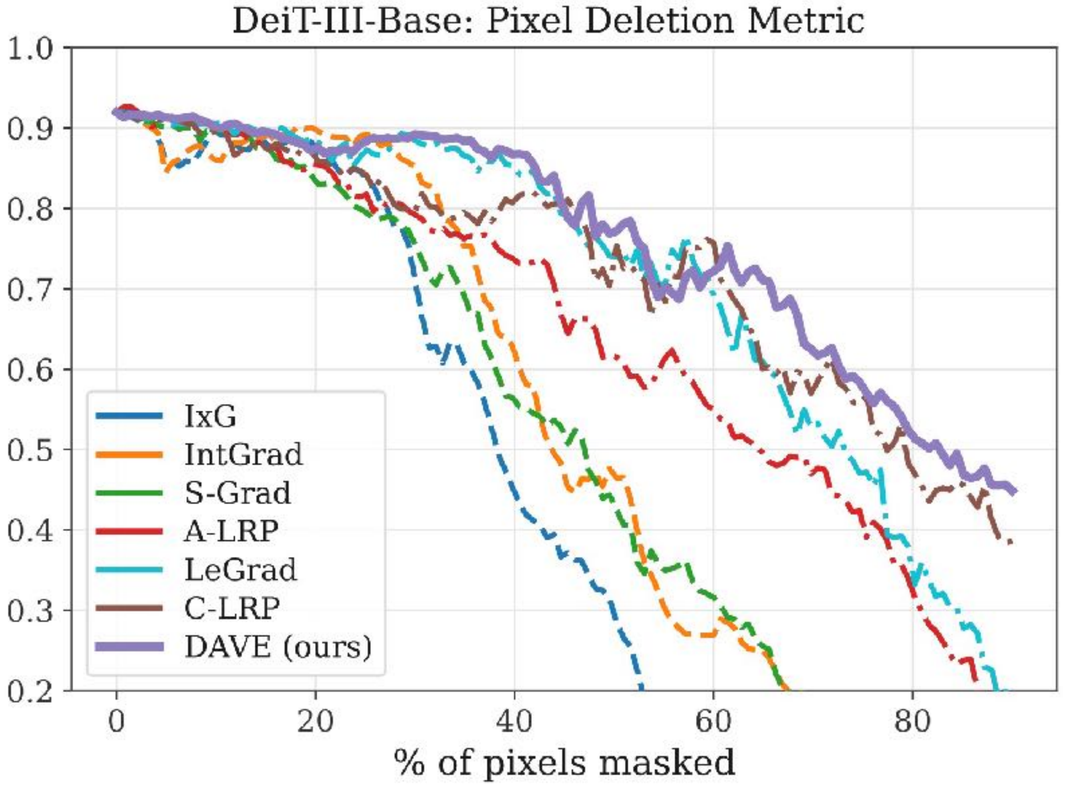
# Quantitative Results: localisation via Grid PG and Energy PG

| | GridPG (%) | | | | EnergyPG (%) | | | |
|---|---|---|---|---|---|---|---|---|
| Method | ViT-B | DeiT-B | D-III-B | DINO-B | ViT-B | DeiT-B | D-III-B | DINO-B |
| I×G | 32.67 | 30.25 | 30.01 | 33.28 | 55.33 | 62.72 | 67.32 | 69.98 |
| IntGrad | 39.86 | 36.11 | 31.68 | 36.98 | 58.51 | 64.22 | 68.12 | 72.15 |
| S-Grad | 34.27 | 30.18 | 31.48 | 33.13 | 56.02 | 60.78 | 68.10 | 70.93 |
| LeGrad | 47.71 | 42.58 | 34.62 | 28.96 | 80.06 | 77.83 | 77.54 | 82.26 |
| A-LRP | 58.40 | 54.63 | 53.84 | 37.49 | 60.75 | 68.16 | 77.65 | 75.98 |
| C-LRP | 54.98 | 55.47 | 52.27 | 49.99 | **80.82** | 79.62 | 81.94 | 81.56 |
| DAVE (ours) | **60.19** | **63.52** | **65.76** | **51.33** | 78.60 | **82.23** | **82.43** | **83.38** |
| Δ | +1.79 | +8.05 | +11.92 | +1.35 | -2.22 | +2.61 | +0.49 | +1.12 |



# Quantitative Results: pixel perturbation (pixel deletion)



# Next steps

Current results only for vision models (classification), also works on self-supervised linear probed models (DINO family):

- ✦ extension to vision language models (CLIP, SigLIP, Llava)
- ✦ extension to convolutional models
- ✦ human study - reviewers ask for that
- ✦ release nicely usable library
- ✦ bias contributions are still not accounted for, so still a problem.

# Fun-Fact about DAVE 

Adam coined the name DAVE - yes it’s actually an acronym for the method/title.

But it’s also a tribute to **Michelangelo’s David**, once hailed as the most precise human sculpture thus fitting, since our DAVE attributions are **highly precise and class-specific.** 🗿✨

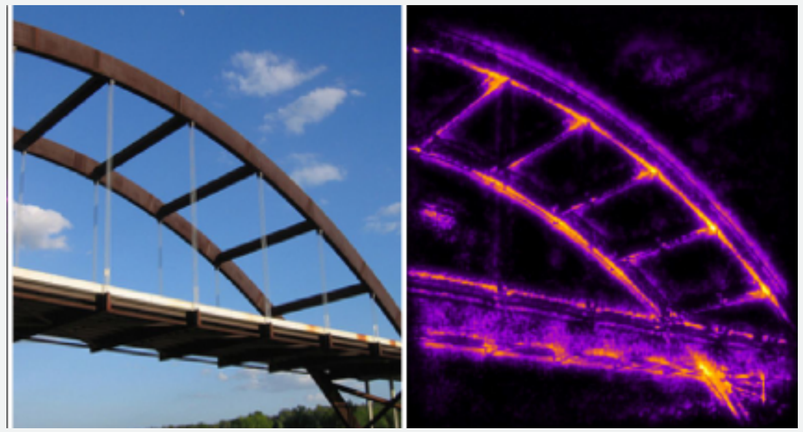
David by Michelangelo

*Galleria dell'Accademia, Florence, Italy*

David by Michelangelo

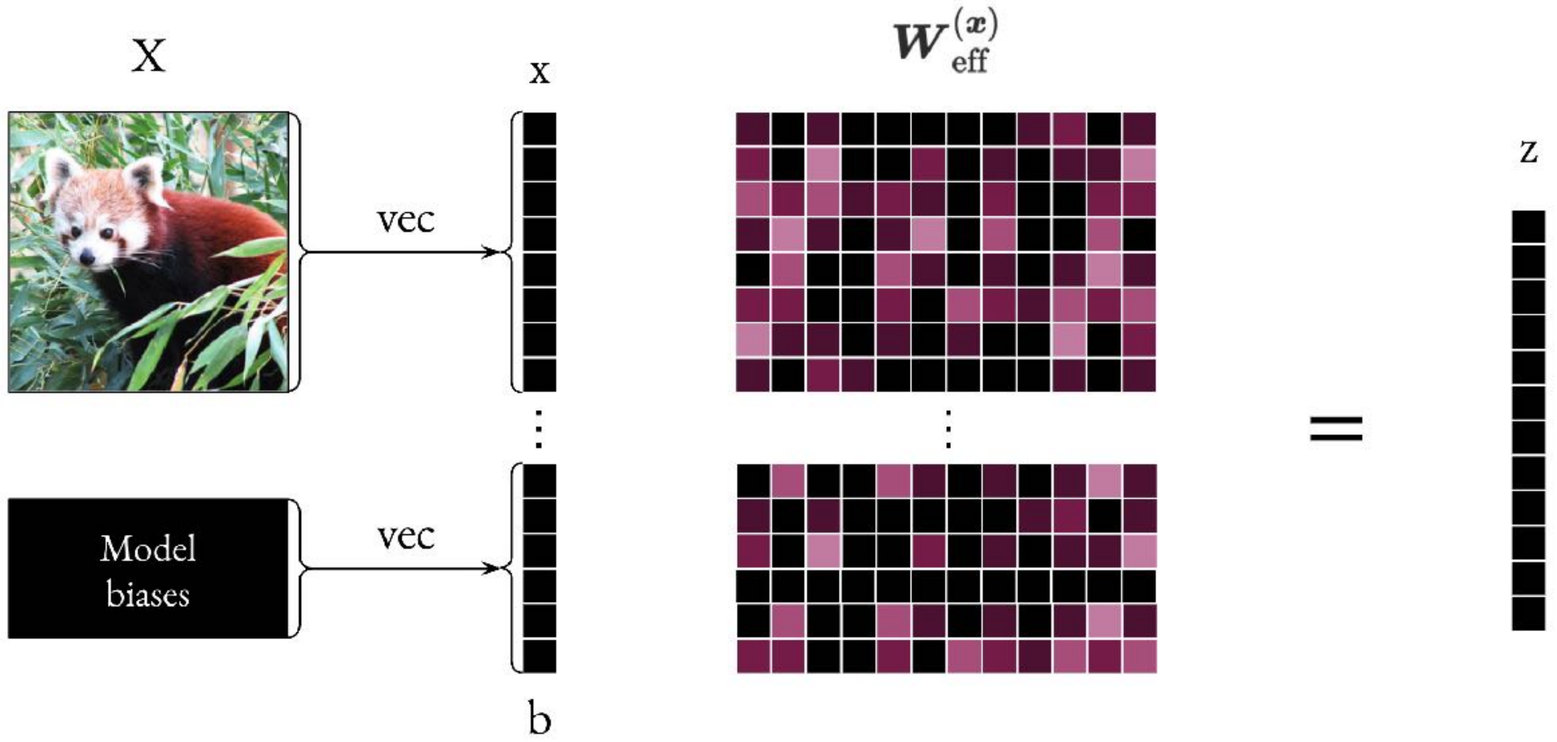
*https://github.com/a-vrobell/DAVE*

# Questions, comments, feedback!



# DAVE - Better Understanding

*ViT is a Dynamic Linear Matrix* $\boldsymbol{z} = \mathrm{ViT}(\boldsymbol{X}) = \boldsymbol{W}_{\mathrm{eff}}^{(\boldsymbol{x})}[\boldsymbol{x}|\boldsymbol{b}]$



# DAVE - Better Understanding

*Layer by layer: general assumption*

- If every base layer is of a form:

$$\mathrm{vec}(\boldsymbol{Z}) = \boldsymbol{W}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{b}$$

- Then the whole ViT:

$$\mathrm{ViT}(\boldsymbol{X}) = \boldsymbol{W}_{\mathrm{eff}}^{(\boldsymbol{x})}[\boldsymbol{x}|\boldsymbol{b}]$$

# DAVE - Better Understanding

## Layer Norm

$$\mathrm{LN}(\boldsymbol{x}) = \boldsymbol{\gamma} \odot \frac{\boldsymbol{x} - \mu(\boldsymbol{x})}{\sigma(\boldsymbol{x})} + \boldsymbol{\beta}$$

$$\mathrm{LN}(\boldsymbol{x}) = \underbrace{\left(\frac{1}{\sigma(\boldsymbol{x})}\,\Gamma\, H\right)}_{\boldsymbol{M}(\boldsymbol{x})}\boldsymbol{x} + \boldsymbol{\beta}$$

Where:

$$H = I - \frac{1}{d}\mathbf{1}\mathbf{1}^\top$$

$$\Gamma = \mathrm{diag}(\boldsymbol{\gamma})$$

# DAVE - Better Understanding

## Layer Norm

For a single token:

$$\mathrm{LN}(\boldsymbol{x}) = \underbrace{\left(\frac{1}{\sigma(\boldsymbol{x})}\,\Gamma\,H\right)}_{\boldsymbol{M}(\boldsymbol{x})}\boldsymbol{x} + \boldsymbol{\beta}$$

Where:

$$H = I - \frac{1}{d}\mathbf{1}\mathbf{1}^\top$$

$$\Gamma = \mathrm{diag}(\boldsymbol{\gamma})$$

# DAVE - Better Understanding

## Layer Norm

For a single token:

$$\mathrm{LN}(\boldsymbol{x}) = \underbrace{\left(\frac{1}{\sigma(\boldsymbol{x})}\,\Gamma\,H\right)}_{\boldsymbol{M}(\boldsymbol{x})}\boldsymbol{x} + \boldsymbol{\beta}$$

For vectorized tokens:

Blocked-diagonal matrix

Where:

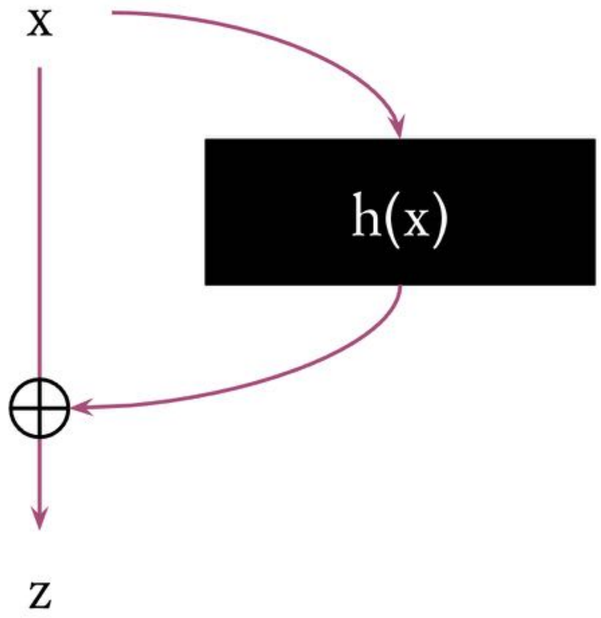
$$H = I - \frac{1}{d}\mathbf{1}\mathbf{1}^\top$$

$$\Gamma = \mathrm{diag}(\boldsymbol{\gamma})$$

# DAVE - Better Understanding

## Residual connections

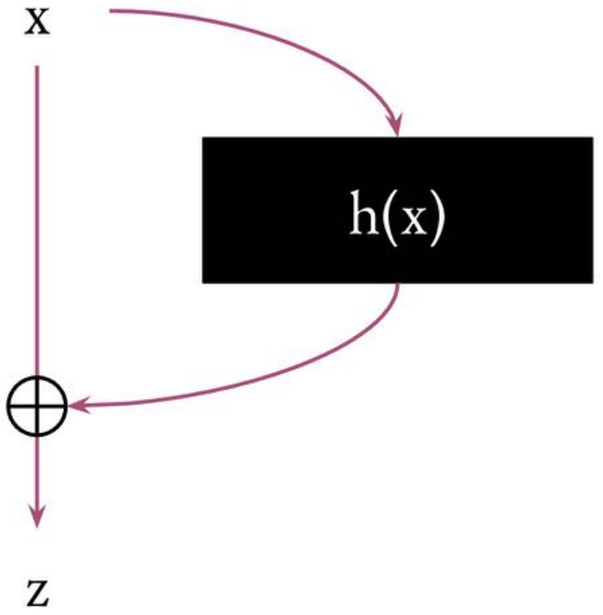
$$z = x + h(x)$$



# DAVE - Better Understanding

## Residual connections

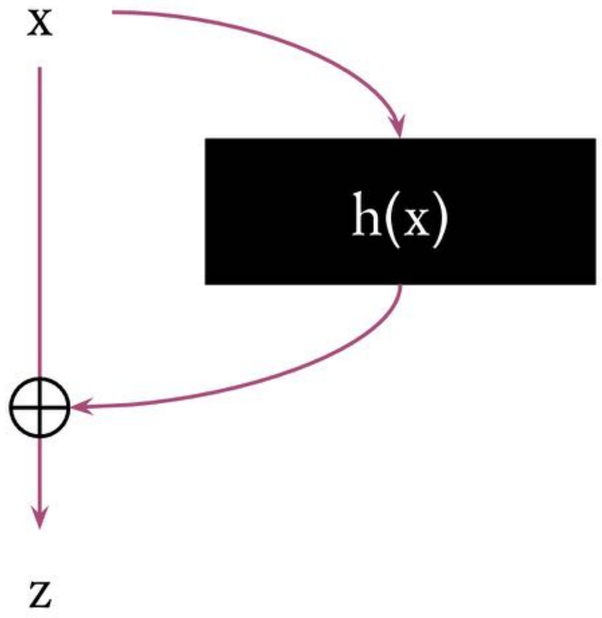
$$\begin{aligned} \boldsymbol{z} &= \boldsymbol{x} + h(\boldsymbol{x}) \\ &= \boldsymbol{x} + [\boldsymbol{W}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{b}] \end{aligned}$$



# DAVE - Better Understanding

## Residual connections

$$
\begin{aligned}
\boldsymbol{z} &= \boldsymbol{x} + h(\boldsymbol{x}) \\
&= \boldsymbol{x} + [\boldsymbol{W}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{b}] \\
&= [\boldsymbol{I} + \boldsymbol{W}(\boldsymbol{x})]\boldsymbol{x} + \boldsymbol{b}
\end{aligned}
$$



# DAVE - The Method

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W}(\boldsymbol{x})\boldsymbol{x}$$

$$\boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x}) = \underbrace{\boldsymbol{W}(\boldsymbol{x})}_{\text{Our ViT matrix}} + \underbrace{\sum_{i=1}^{d} x_i \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{x}_i}(\boldsymbol{x})}_{\text{JVP}}$$

# DAVE - The Method

**Effective weights and how to find them efficiently**

- **Detach** all dynamic matrices on the forward pass (attention, GELU multiplier, etc.)
- Compute “detached” **backward pass**
- Like: CoDA, B-cos

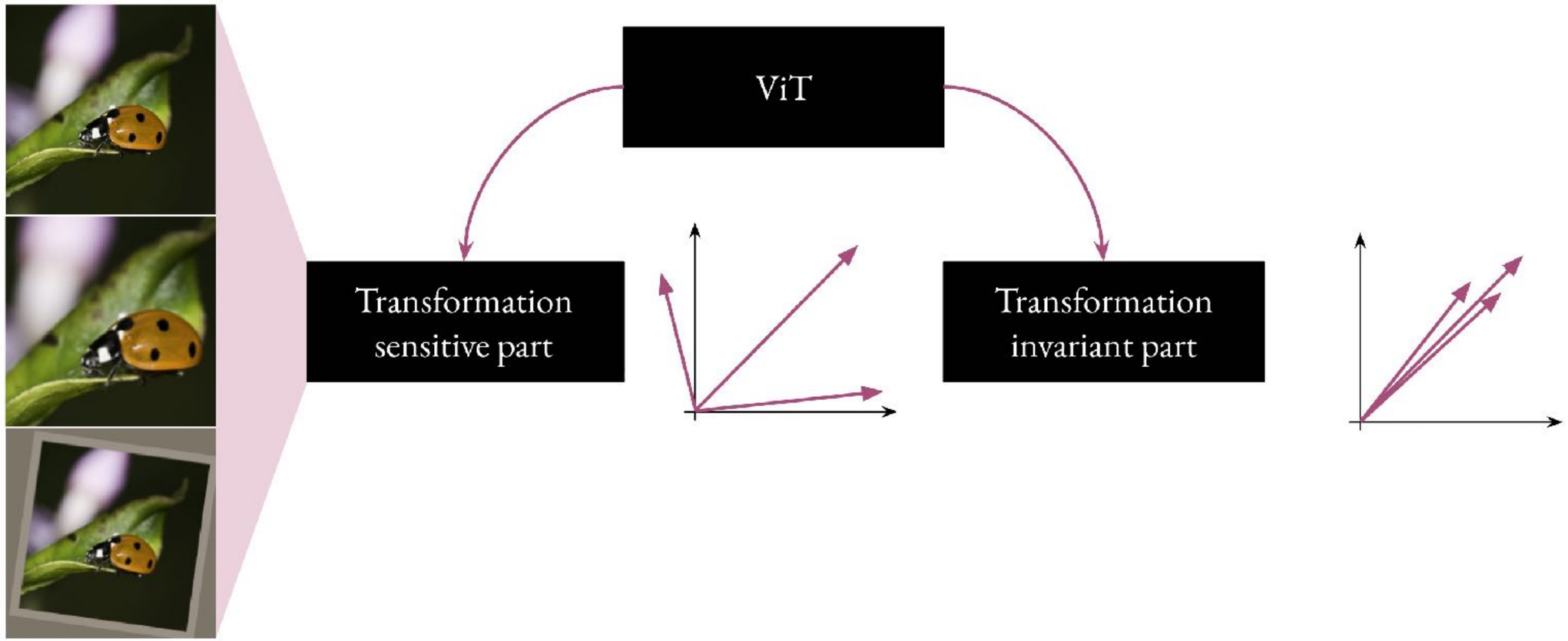
$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W}(\boldsymbol{x})\boldsymbol{x}$$

$$\boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x}) = \underbrace{\boldsymbol{W}(\boldsymbol{x})}_{\text{Our ViT matrix}} + \underbrace{\sum_{i=1}^{d} \boldsymbol{x}_i \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{x}_i}(\boldsymbol{x})}_{\text{JVP}}$$

$$\underbrace{D_{\boldsymbol{X}}F}_{\substack{\text{layer}\\ \text{derivative}}} = \underbrace{L(\boldsymbol{X})}_{\substack{\text{effective}\\ \text{transformation}}} + \underbrace{\left((D_{\boldsymbol{X}}L(\boldsymbol{X})(\cdot))\,\boldsymbol{X}\right)}_{\substack{\text{operator}\\ \text{variation}}}$$

# DAVE - Better Understanding

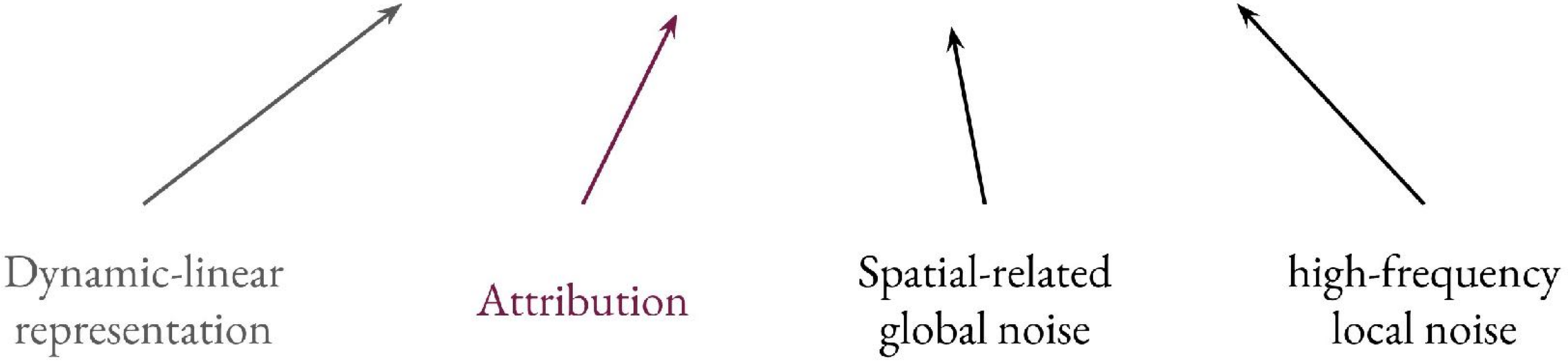
## Intuition



# DAVE - Better Understanding

**Assumption:** For a well-trained model

$$C(x) = C^{\star}(x) + n_{T}(x) + n_{\epsilon}(x)$$



# DAVE - Better Understanding

**Assumption:** For a well-trained model

- Global noise is spatially **“fixed”:**

$$\mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}(\boldsymbol{n}_{\tau}(\tau(\boldsymbol{x})))] = 0$$

- Attribution is **transformation-invariant:**

$$\mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}(\boldsymbol{C}^{\star}(\tau(\boldsymbol{x})))] \approx \boldsymbol{C}^{\star}(\boldsymbol{x})$$

- Local noise has significantly **higher frequency band** than attribution

# DAVE - Better Understanding

## Separating spatial noise

$$\mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}(\boldsymbol{C}(\tau(\boldsymbol{x})))] =$$

# DAVE - Better Understanding

## Separating spatial noise

$$\mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}(\boldsymbol{C}(\tau(\boldsymbol{x})))] =$$

$$\mathbb{E}_{\tau \sim p_{transform}} \tau^{\dagger}[\boldsymbol{C}^{\star}(\tau(\boldsymbol{x})) + \boldsymbol{n}_{\tau}(\tau(\boldsymbol{x})) + \boldsymbol{n}_{\epsilon}(\tau(\boldsymbol{x}))] =$$

# DAVE - Better Understanding

## Separating spatial noise

$$\mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}(\boldsymbol{C}(\tau(\boldsymbol{x})))] =$$

$$\mathbb{E}_{\tau \sim p_{transform}} \tau^{\dagger}[\boldsymbol{C}^{\star}(\tau(\boldsymbol{x})) + \boldsymbol{n}_{\tau}(\tau(\boldsymbol{x})) + \boldsymbol{n}_{\epsilon}(\tau(\boldsymbol{x}))] =$$

$$\mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}\boldsymbol{C}^{\star}(\tau(\boldsymbol{x}))] + \mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}\boldsymbol{n}_{\tau}(\tau(\boldsymbol{x}))] + \mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}\boldsymbol{n}_{\epsilon}(\tau(\boldsymbol{x}))] =$$

# DAVE - Better Understanding

## Separating spatial noise

$$\mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}(\boldsymbol{C}(\tau(\boldsymbol{x})))] =$$

$$\mathbb{E}_{\tau \sim p_{transform}} \tau^{\dagger}[\boldsymbol{C}^{\star}(\tau(\boldsymbol{x})) + \boldsymbol{n}_{\tau}(\tau(\boldsymbol{x})) + \boldsymbol{n}_{\epsilon}(\tau(\boldsymbol{x}))] =$$

$$\mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}\boldsymbol{C}^{\star}(\tau(\boldsymbol{x}))] + \mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}\boldsymbol{n}_{\tau}(\tau(\boldsymbol{x}))] + \mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}\boldsymbol{n}_{\epsilon}(\tau(\boldsymbol{x}))] =$$

*Plugging in the 1st and 2nd assumption:*

- Global noise is spatially **“fixed”**:

$$\mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}(\boldsymbol{n}_{\tau}(\tau(\boldsymbol{x})))] = 0$$

- Attribution is **transformation-invariant:**

$$\mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}(\boldsymbol{C}^{\star}(\tau(\boldsymbol{x})))] \approx \boldsymbol{C}^{\star}(\boldsymbol{x})$$

# DAVE - Better Understanding

## Separating spatial noise

$$\mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}(\boldsymbol{C}(\tau(\boldsymbol{x})))] =$$

$$\mathbb{E}_{\tau \sim p_{transform}} \tau^{\dagger}[\boldsymbol{C}^{\star}(\tau(\boldsymbol{x})) + \boldsymbol{n}_{\tau}(\tau(\boldsymbol{x})) + \boldsymbol{n}_{\epsilon}(\tau(\boldsymbol{x}))] =$$

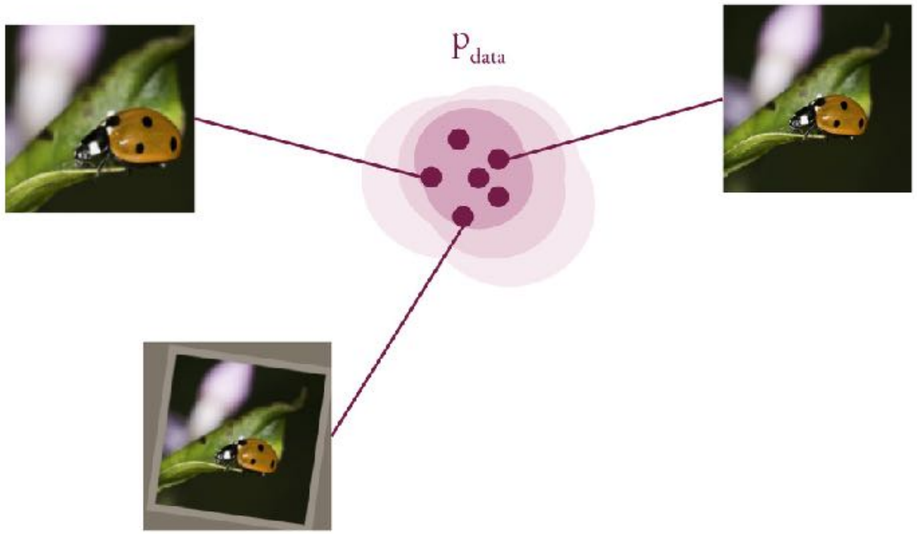
$$\mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}\boldsymbol{C}^{\star}(\tau(\boldsymbol{x}))] + \mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}\boldsymbol{n}_{\tau}(\tau(\boldsymbol{x}))] + \mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}\boldsymbol{n}_{\epsilon}(\tau(\boldsymbol{x}))] =$$

$$\boldsymbol{C}^{\star}(\boldsymbol{x}) + \mathbb{E}_{\tau \sim p_{transform}}[\tau^{\dagger}\boldsymbol{n}_{\epsilon}(\tau(\boldsymbol{x}))]$$

# DAVE - Better Understanding

**Constructing neighbourhood** for reliable and robust features:

sample points as small augmentations applied to the input such that the output logic distribution is preserved



# DAVE - Better Understanding

**Transformations as layers** for attribution pushback:

Each transformation is just **another layer**

$$\tau(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x} \;, \quad \boldsymbol{x} = \mathrm{vec}(\boldsymbol{X})$$

**Transformations**

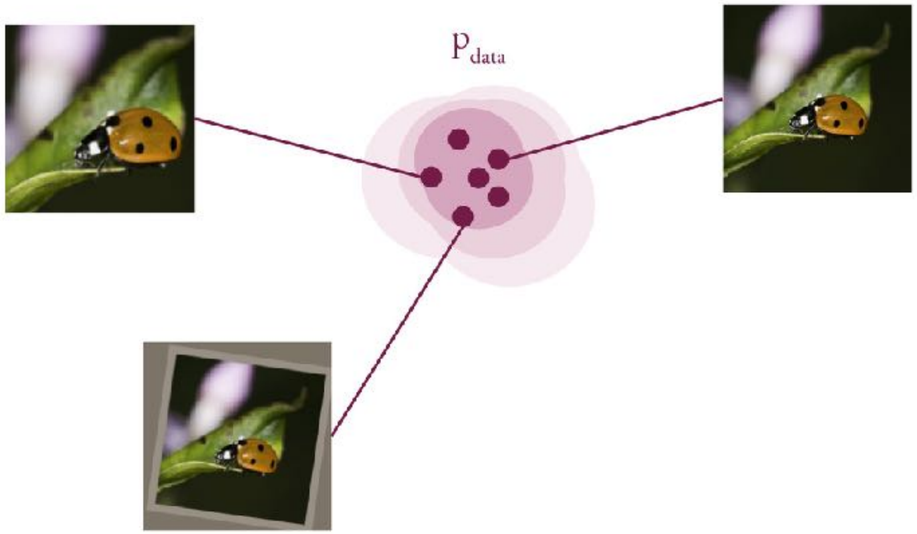
✦Rotation

✦Offset

✦Padding

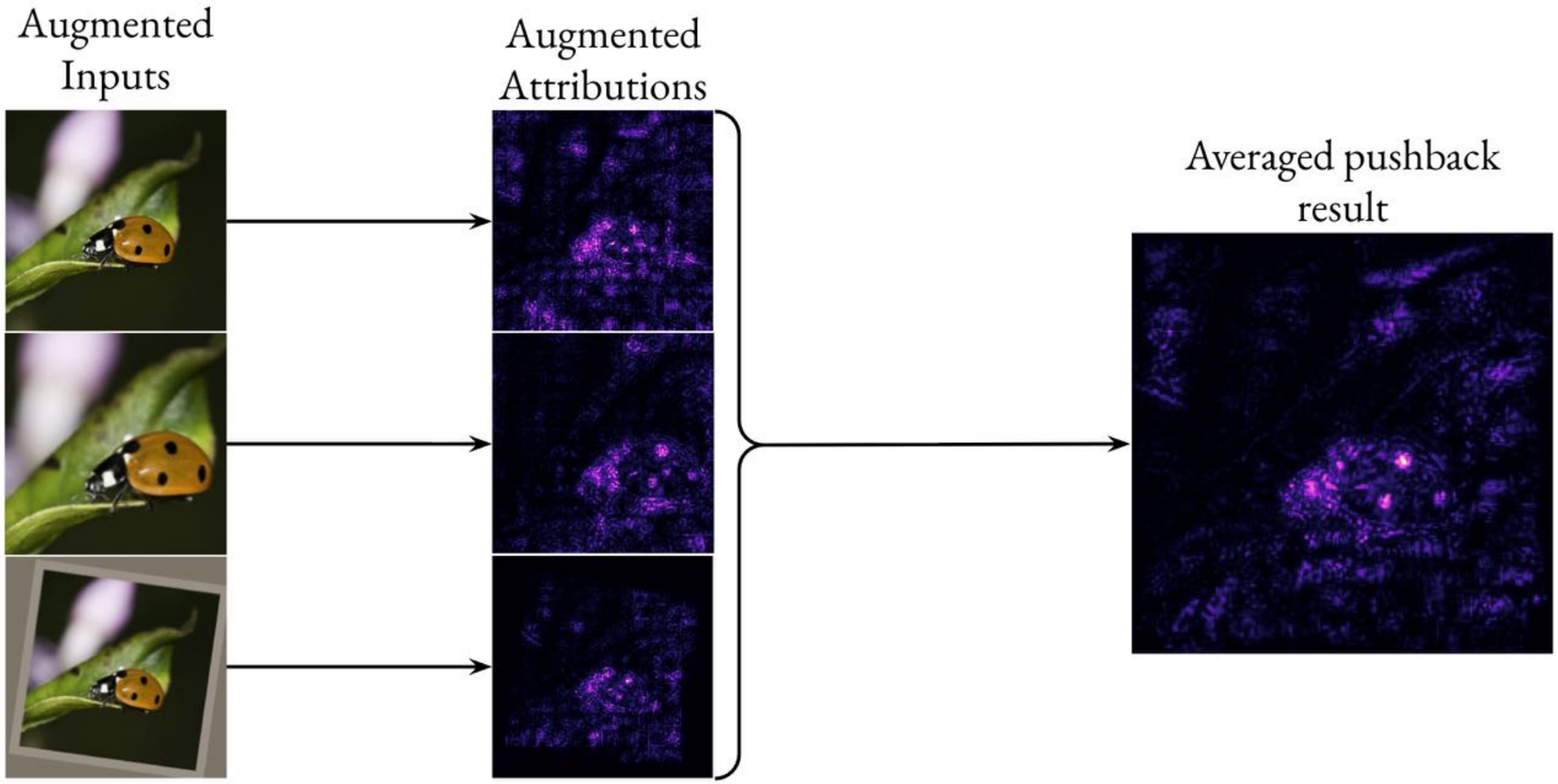
✦Scaling

✦Blurring



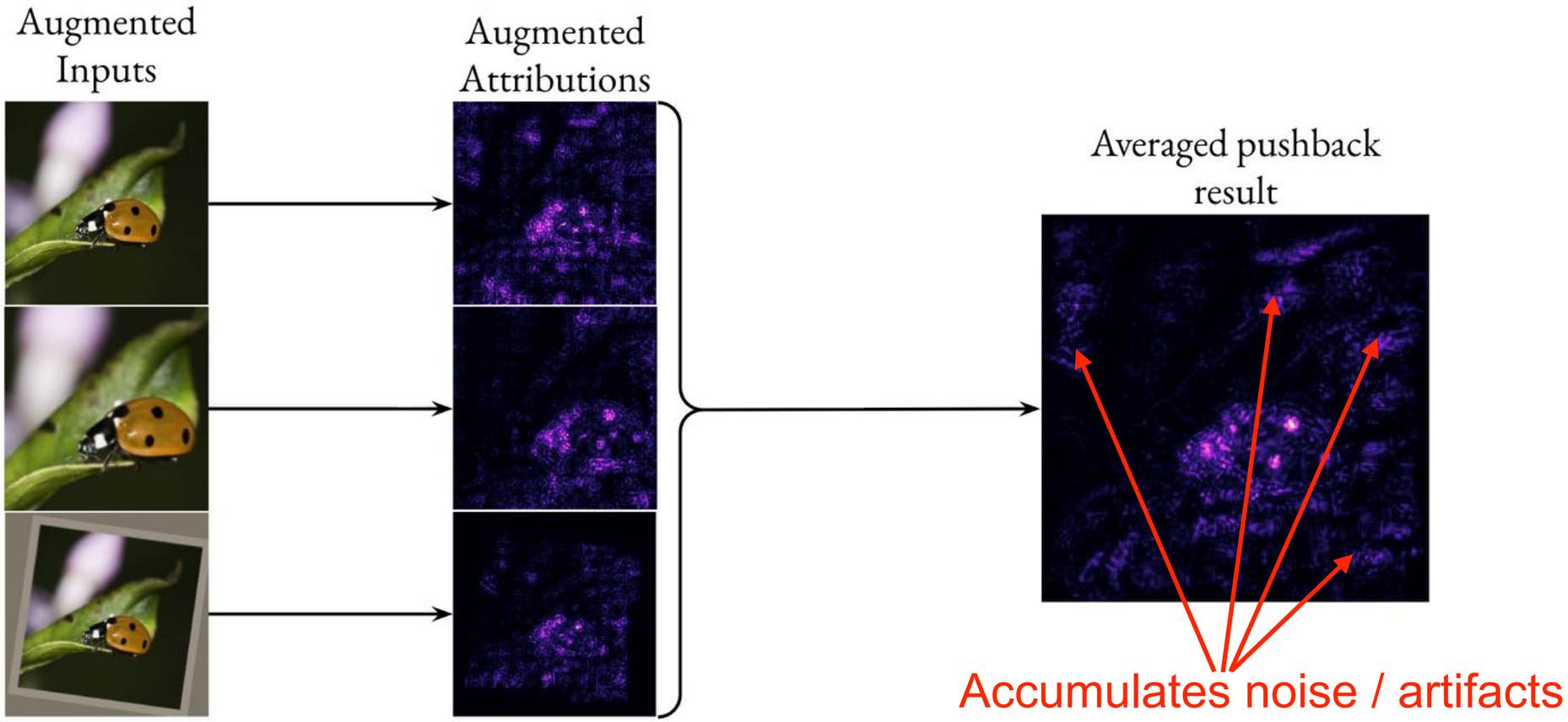
# DAVE - Better Understanding

Effective input **equivariant** attributions



# DAVE - Better Understanding

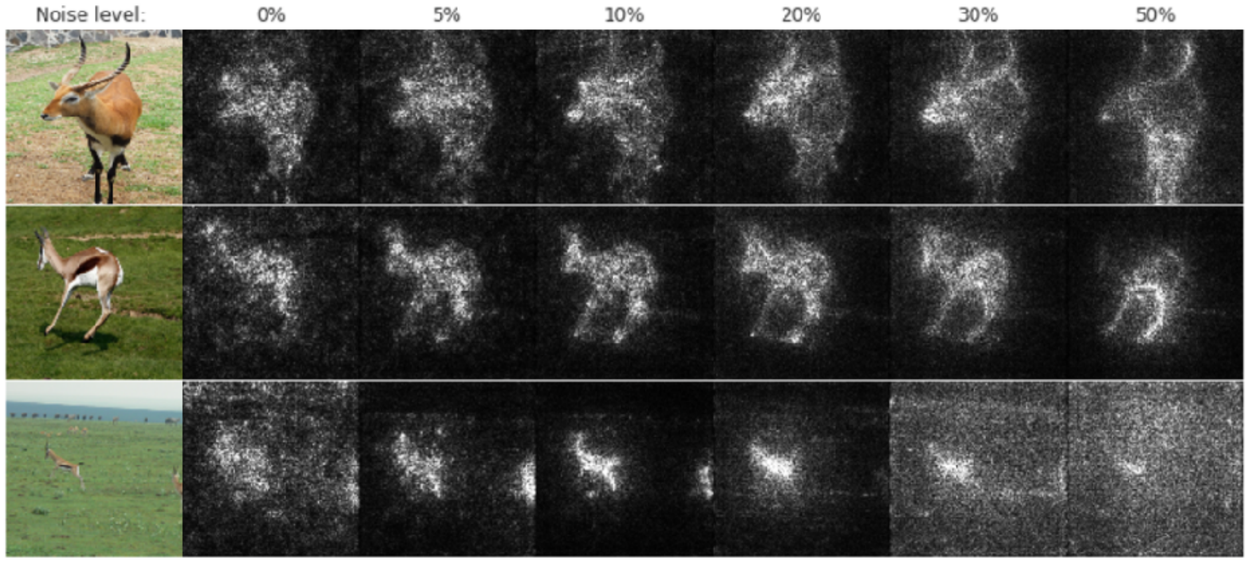
Effective input **equivariant** attributions



# DAVE - Better Understanding

**Separating local noise:** removing artifacts

**SmoothGrad: removing noise by adding noise**



# DAVE - Better Understanding

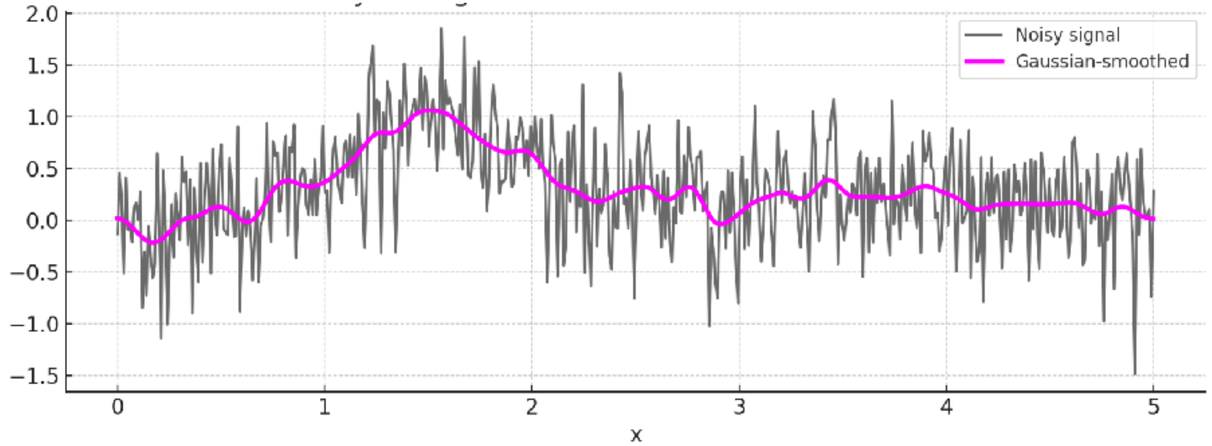
**Adding noise = low-pass filtering** (in attribution space)

Our attribution:

$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{W}_{\text{eff}}^{(\boldsymbol{x})} \boldsymbol{x}$$

Adding noise:

$$\tilde{\boldsymbol{C}}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} [\boldsymbol{C}(\underbrace{\boldsymbol{x} + \boldsymbol{\epsilon}}_{\boldsymbol{z}})]$$

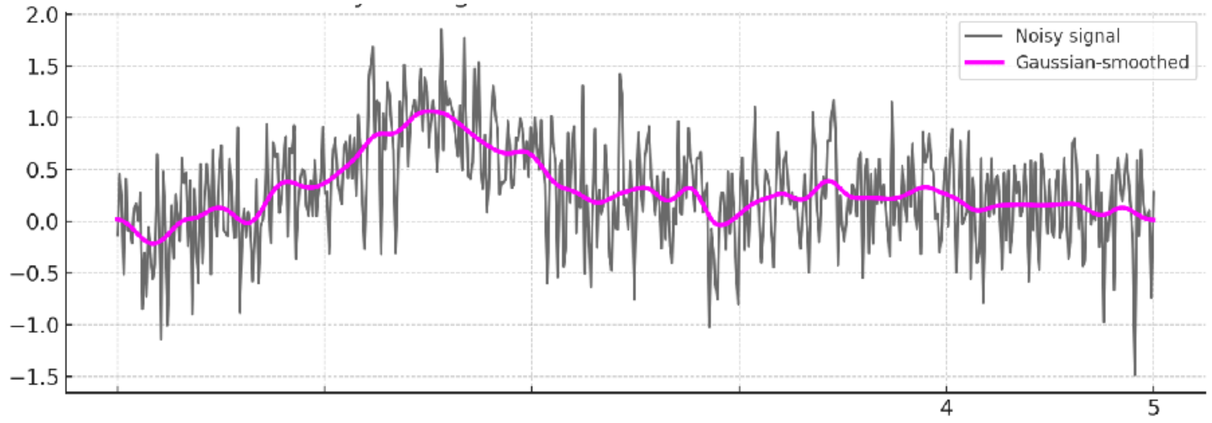


# DAVE - Better Understanding

**Adding noise = low-pass filtering** (in attribution space)

Our attribution:

$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{W}_{\text{eff}}^{(\boldsymbol{x})}\boldsymbol{x}$$




Adding noise:

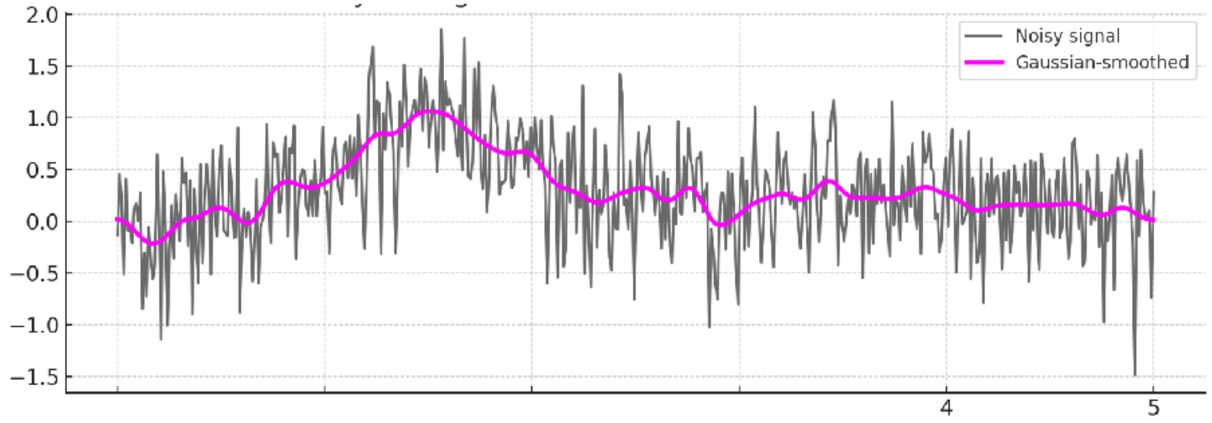
$$\tilde{\boldsymbol{C}}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\boldsymbol{C}(\underbrace{\boldsymbol{x}+\boldsymbol{\epsilon}}_{\boldsymbol{z}})] = \int \boldsymbol{C}(\boldsymbol{z})\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\boldsymbol{I})d\boldsymbol{\epsilon} =$$

# DAVE - Better Understanding

**Adding noise = low-pass filtering** (in attribution space)

Our attribution:

$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{W}_{\text{eff}}^{(\boldsymbol{x})}\boldsymbol{x}$$



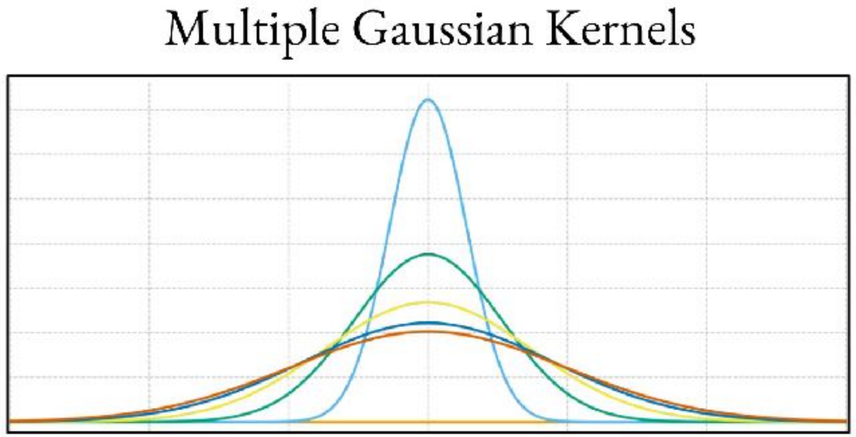

Adding noise:

$$\tilde{\boldsymbol{C}}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}[\boldsymbol{C}(\underbrace{\boldsymbol{x}+\boldsymbol{\epsilon}}_{\boldsymbol{z}})] = \int \boldsymbol{C}(\boldsymbol{z})\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\boldsymbol{I})d\boldsymbol{\epsilon} =$$

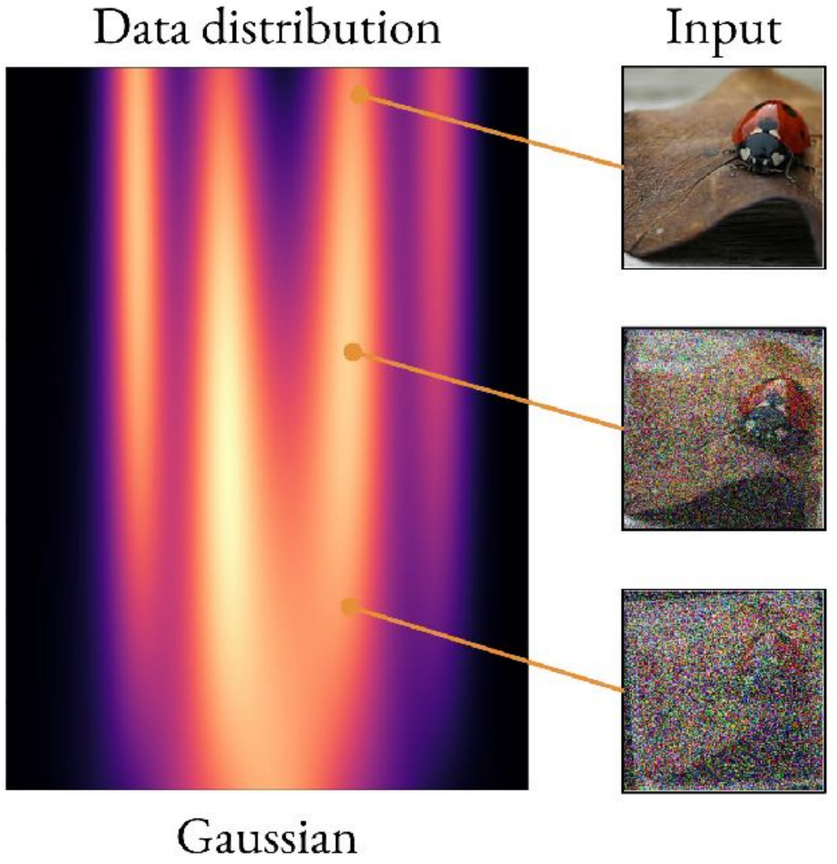
$$= \int \boldsymbol{C}(\boldsymbol{z})\boldsymbol{G}(\boldsymbol{z}-\boldsymbol{x})d\boldsymbol{z} = (\boldsymbol{C}*\boldsymbol{G})(\boldsymbol{x})$$

# DAVE - Better Understanding

**Let’s apply multiple low-pass filters:** noise interpolation path

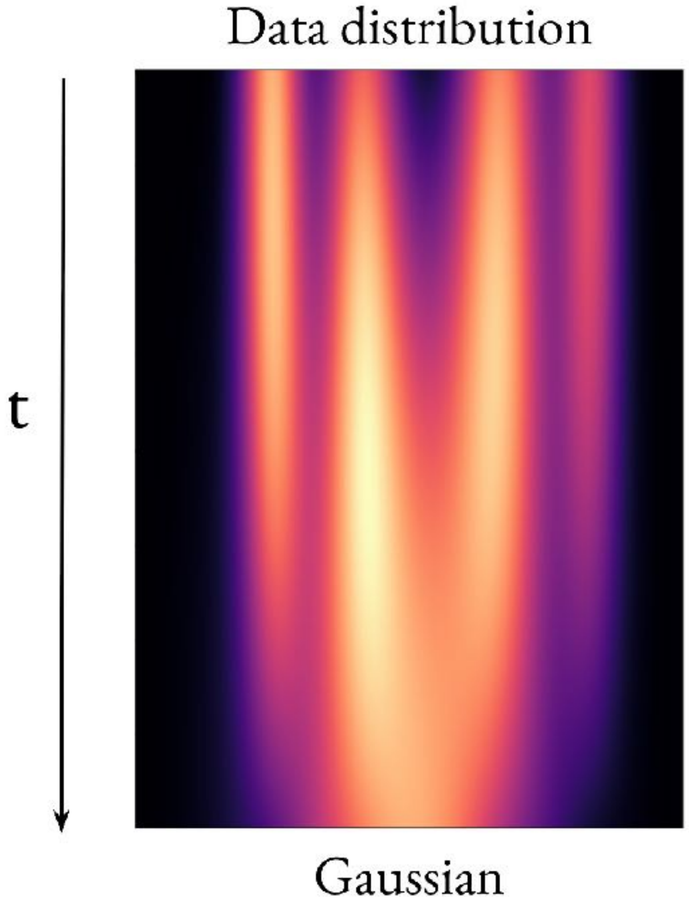



=



# DAVE - Better Understanding

**Let's apply multiple low-pass filters:** noise interpolation path




- Variance-preserving path (chosen empirically):

$$\boldsymbol{x}_t = \cos(t)\boldsymbol{x} + \sin(t)\boldsymbol{\epsilon}, \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

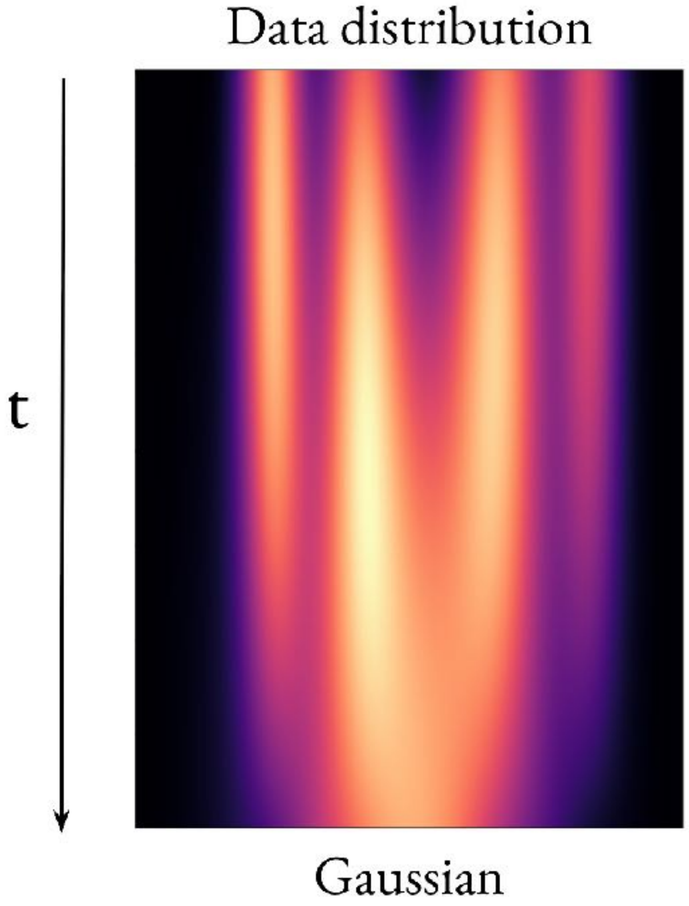
- Final aggregation:

$$\tilde{\boldsymbol{C}}(\boldsymbol{x}) = \mathbb{E}_{t,\boldsymbol{\epsilon}}[\boldsymbol{C}(\boldsymbol{x}_t)] =$$

$$= \mathbb{E}_t[(\boldsymbol{C} * \boldsymbol{G}_t)(\cos(t)\boldsymbol{x})]$$

# DAVE - Better Understanding

**Let's apply multiple low-pass filters:** noise interpolation path




- Variance-preserving path (chosen empirically):

$$\boldsymbol{x}_t = \cos(t)\boldsymbol{x} + \sin(t)\boldsymbol{\epsilon}, \qquad \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

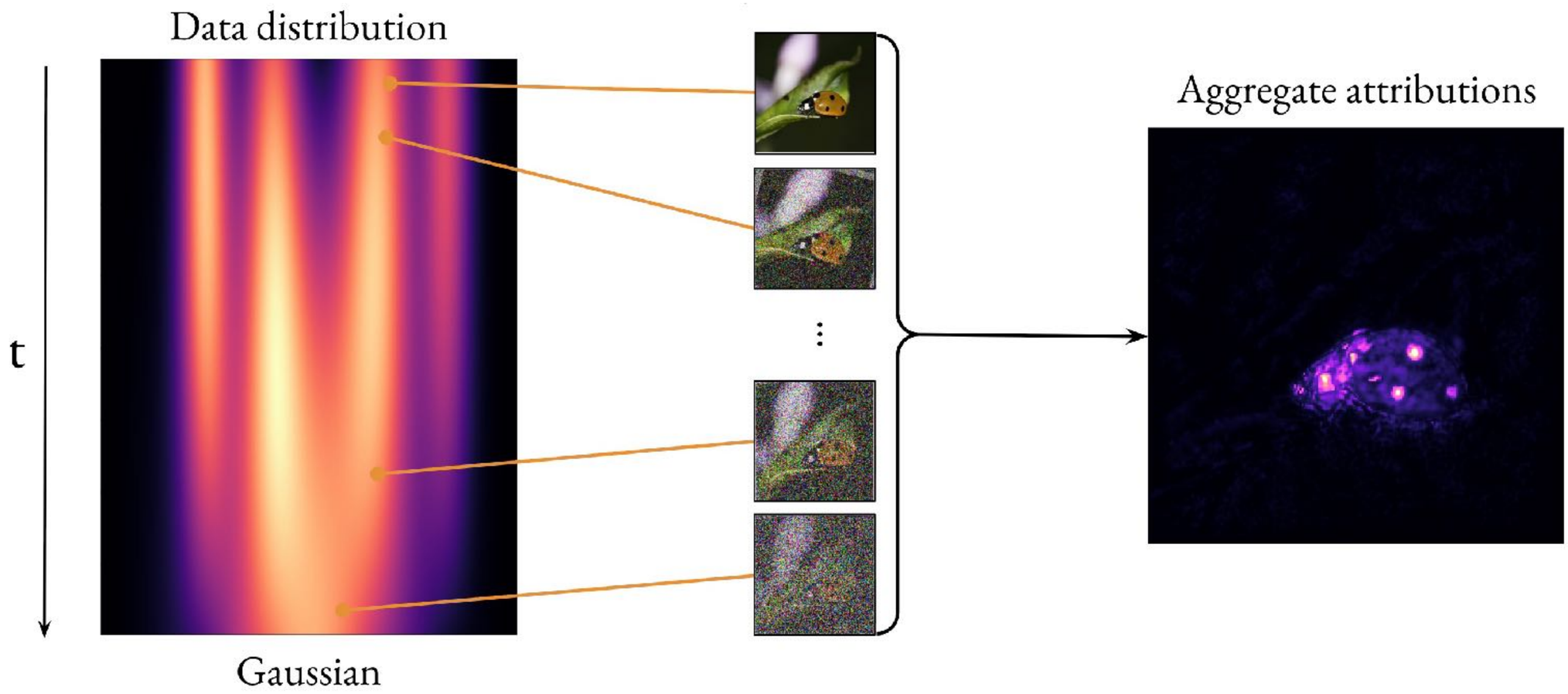
- Final aggregation:

$$\tilde{\boldsymbol{C}}(\boldsymbol{x}) = \mathbb{E}_{t,\boldsymbol{\epsilon}}[\boldsymbol{C}(\boldsymbol{x}_t)] =$$

$$= \mathbb{E}_t[(\boldsymbol{C} * \boldsymbol{G}_t)(\cos(t)\boldsymbol{x})]$$

# DAVE - Better Understanding

**Final pipeline:** augment and interpolate noise over the path

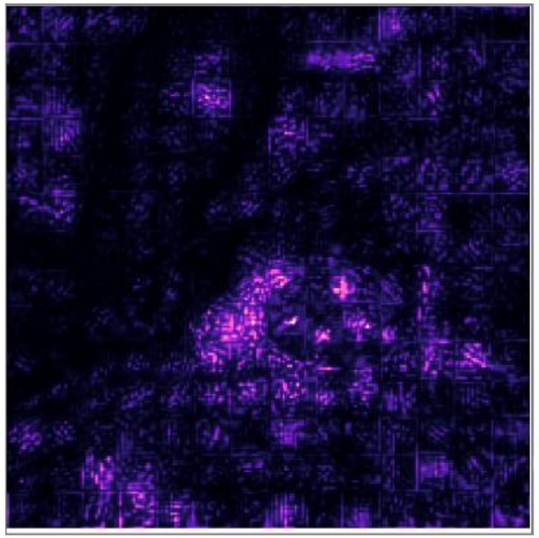


# DAVE - Better Understanding

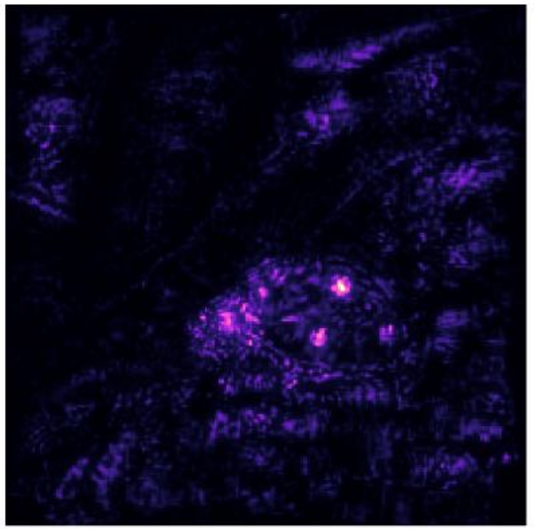
**Final pipeline:** augment and interpolate noise over the path

Input Image

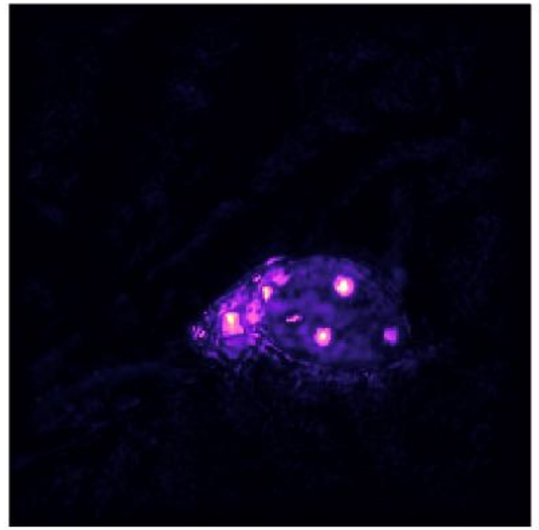
Single point



+ Input Augmentations



+ Noise interpolation path



# DAVE - The Method

DAVE interprets attribution as the stable and locally equivariant effective transformation that a Vision Transformer applies to its input.

Under architectural assumptions of ViTs, it extracts this transformation from the input gradient by decomposing it into a direct input–output transformation and a local variation term.

The method discards the local variation term, which captures input-dependent sensitivity of internal model mechanisms, and aggregates the remaining transformation over a distribution of inputs.

This distribution is constructed to suppress locally non-equivariant components of the transformation, as well as high-frequency operator fluctuations, while preserving consistent inputdependent attribution structure.