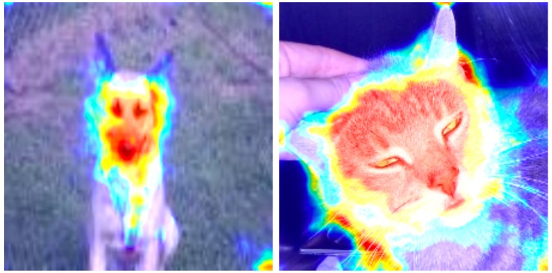
Respect the model : Fine-grained and robust explanation with Sharing ratio decomposition

Sangyu Han, Yearim Kim, Nojun Kwak

MIPALaboratory
Machine Intelligence
& Pattern Analysis

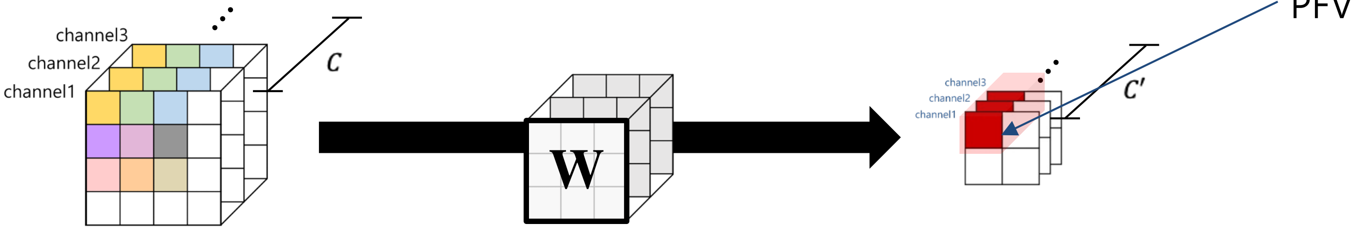
Background

- Attribution method
  - An XAI approach that assigns importance to input pixels
  - Until now, a neuron has been considered as a representation unit
  - However, this perspective has difficulties in interpreting neuronal interactions
  - Vulnerable to adversarial attack targeting explanation
- Our contribution
  - Introduce vector perspective that adhere to model inference
  - Robust explanation with State-of-the-Art performance
  - Good result on every activation function



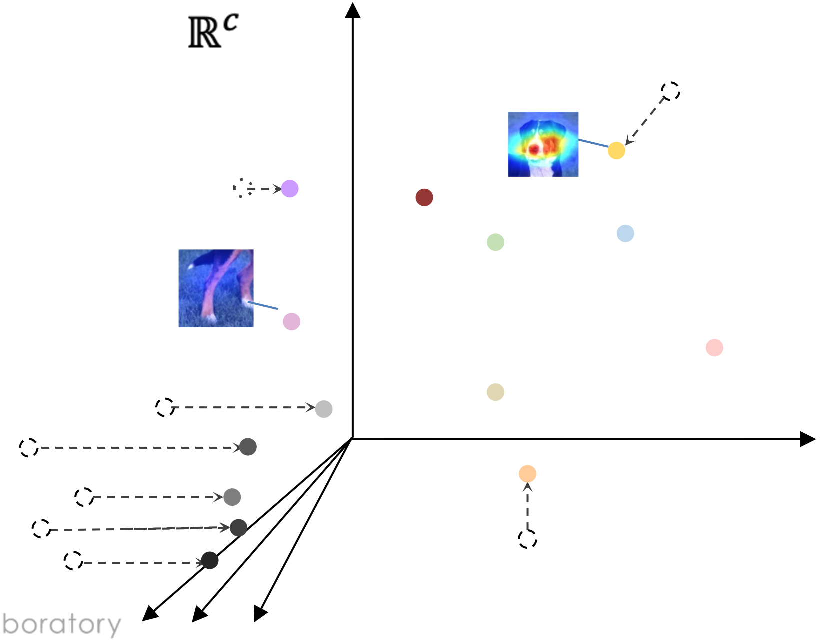
Introduce a vector perspective

- A neuron is a function of its Receptive Field (RF).
- Since neurons in the same location of its channel share the spatial location, combine them as a vector.
- Denote this vector as Pointwise Feature Vector, the representation unit of CNN.
- PFV encodes its RF.



Introduce a vector perspective

- Each layer transforms PFV vector space
- Relu : nonlinearly compress PFV vector field

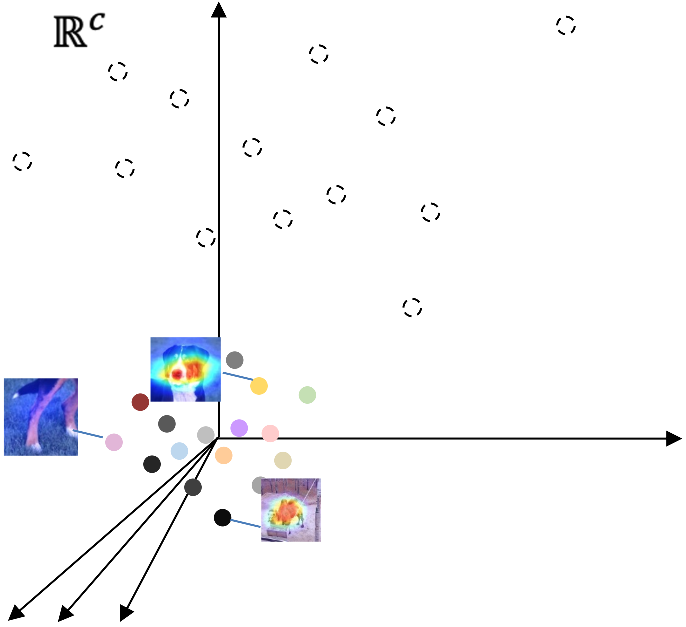


$$f = \begin{cases} (x < 0) & f(x) = 0 \\ (x \geq 0) & f(x) = x \end{cases}$$

ReLU makes the PFV vector field like pushing field into a box

Introduce a vector perspective

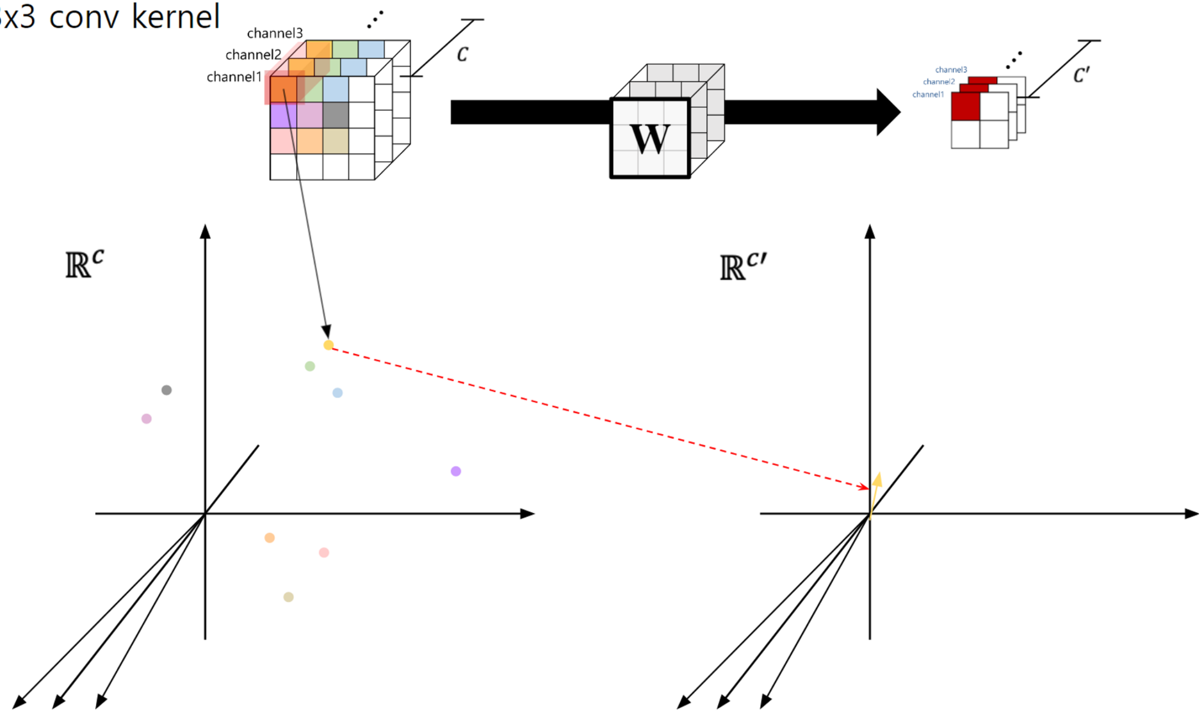
- Each layer transforms PFV vector space
- Batchnorm : normalize PFV space, then spread each axis and move parallel



Introduce a vector perspective

- Convolution layer : Sum of linearly transformed PFVs

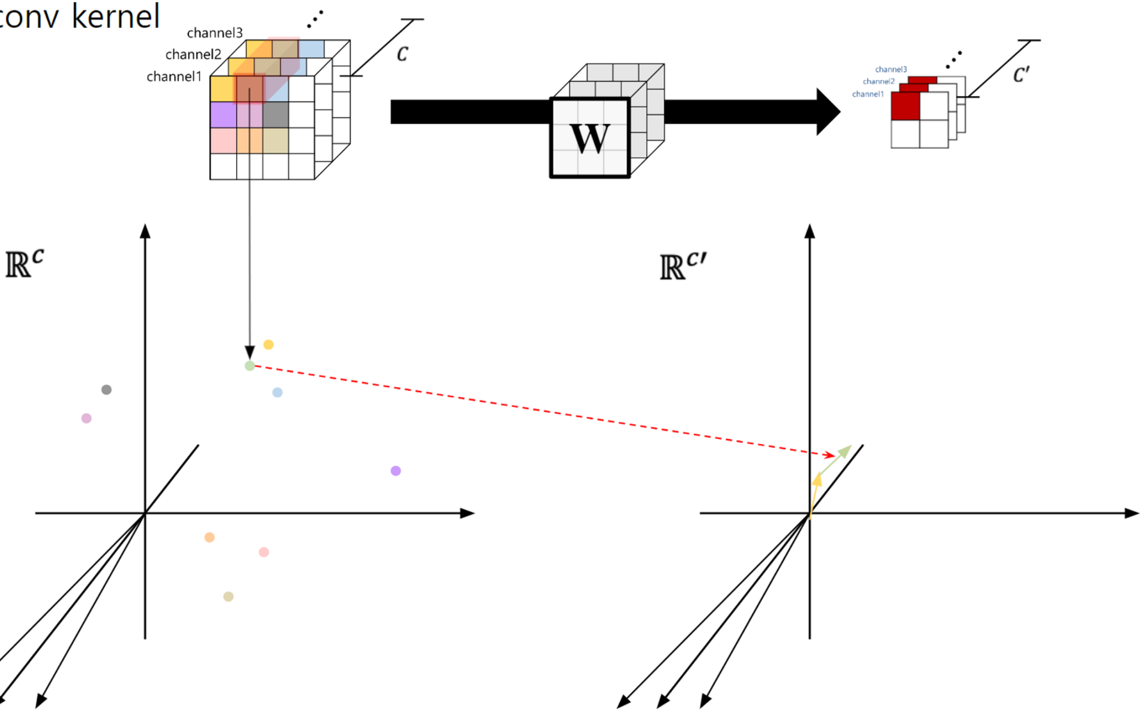
Ex) 3x3 conv kernel



Introduce a vector perspective

- Convolution layer : Sum of linearly transformed PFVs

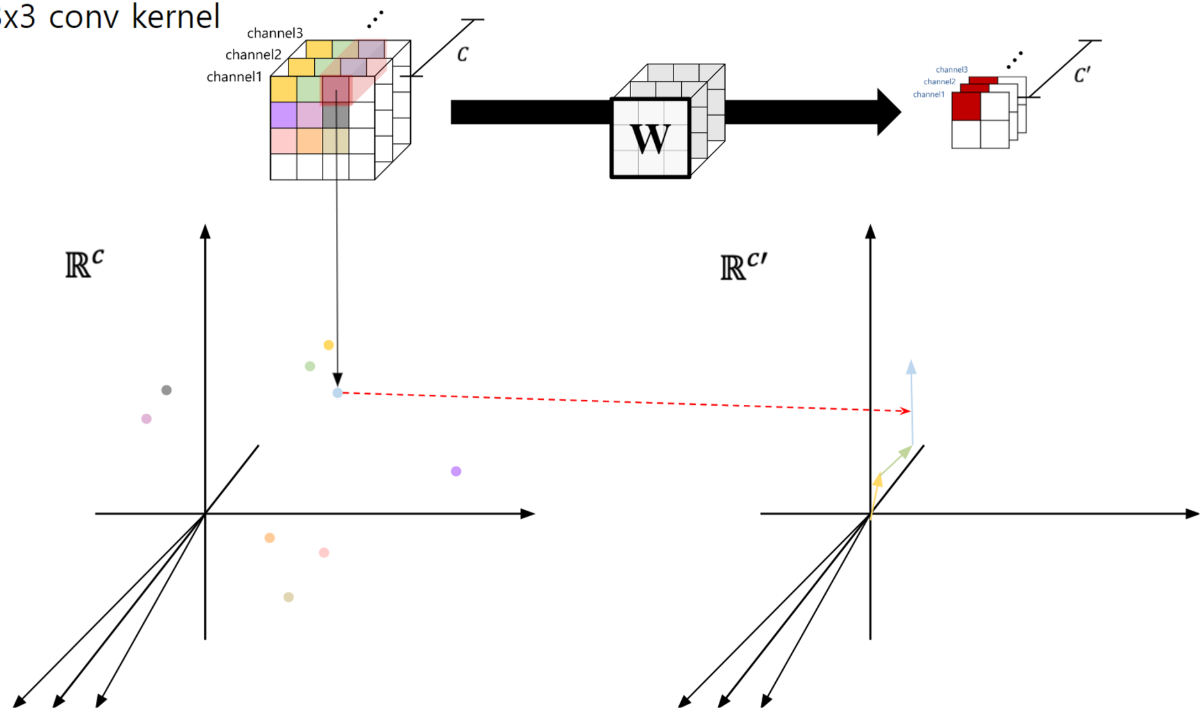
Ex) 3x3 conv kernel



Introduce a vector perspective

- Convolution layer : Sum of linearly transformed PFVs

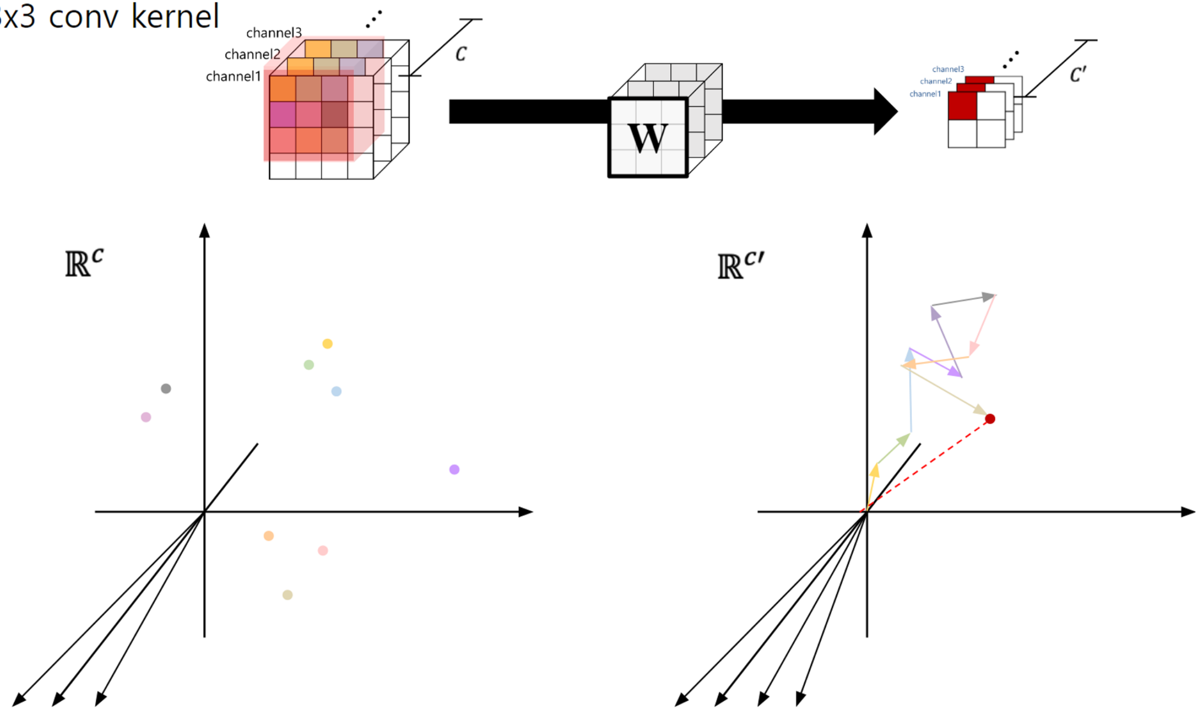
Ex) 3x3 conv kernel



Introduce a vector perspective

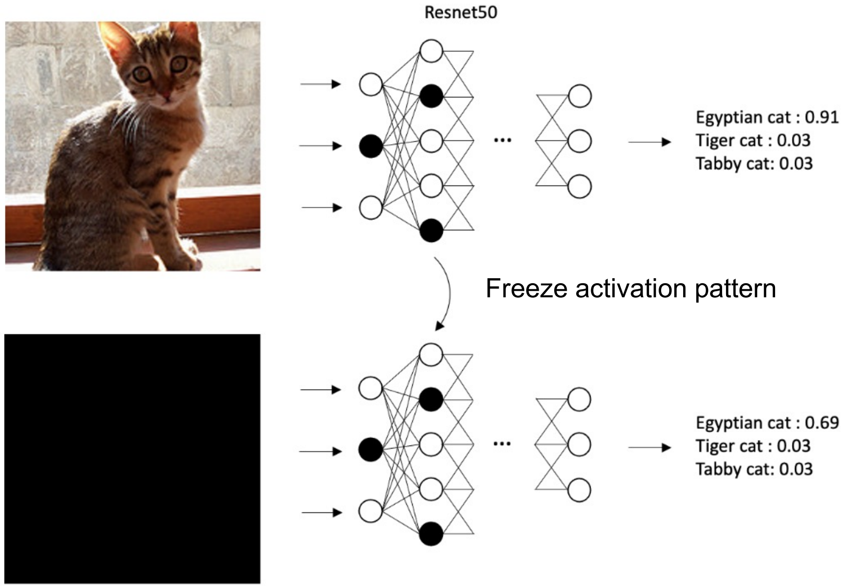
- Convolution layer : Sum of linearly transformed PFVs

Ex) 3x3 conv kernel



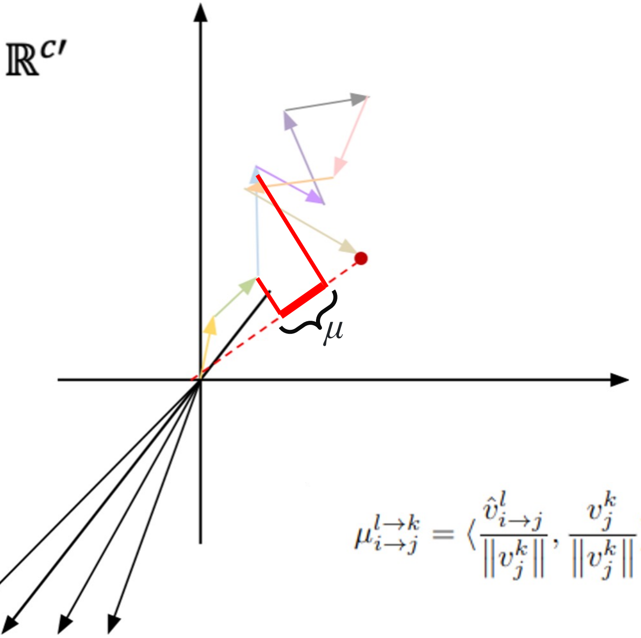
Method

- APOP : The contribution to activation pattern should be considered



Method

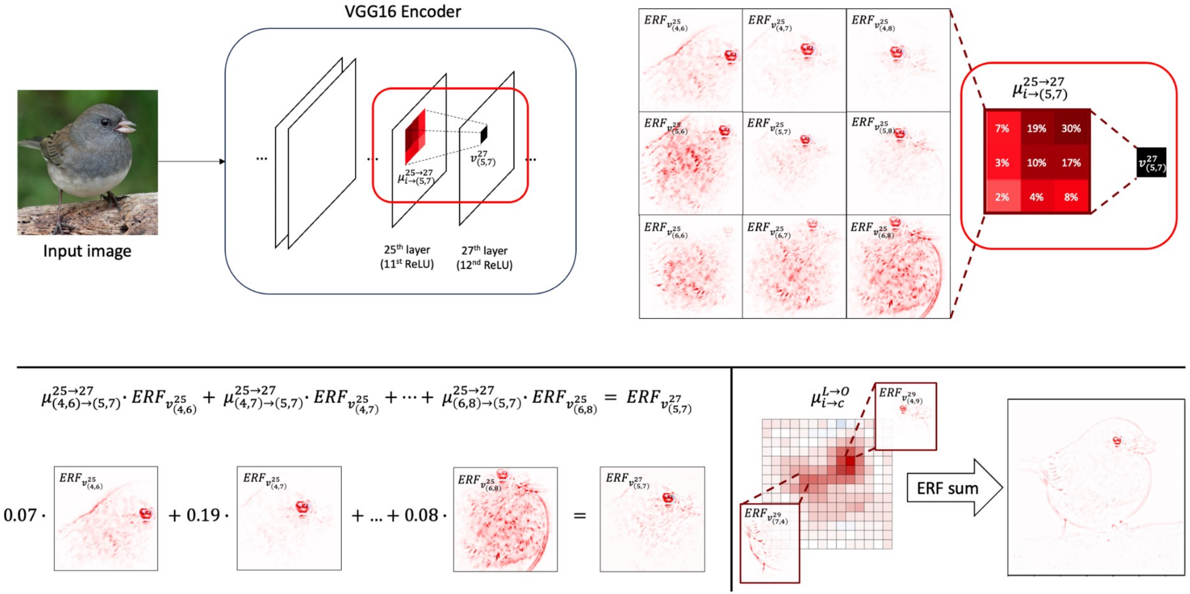
- APOP : The contribution to activation pattern should be considered
- To consider this, calculate Sharing ratio μ as projection proportion of preactivation PFV



$$\mu_{i\to j}^{l\to k} = \langle \frac{\hat{v}_{i\to j}^{l}}{\|v_j^k\|}, \frac{v_j^k}{\|v_j^k\|} \rangle \text{ where } \hat{v}_{i\to j}^{l} = f_{i\to j}^{l}(v_i^l), \quad \text{i.e,} \quad \sum_{i\in RF_j^k} \mu_{i\to j}^{l\to k} = 1.$$

Method

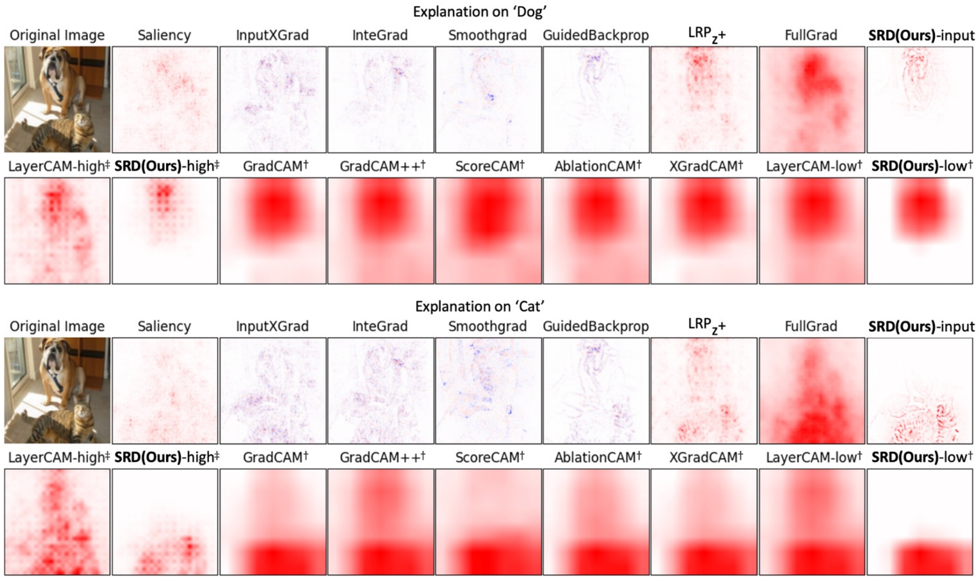
- Effective receptive field(ERF) : the contribution of pixels to form a PFV
- Build a ERF with ERF of PFVs that connected in computation graph
- Final attribution map is weighted sum of encoder output PFV's ERF



$$\mu^{25\to27}_{(4,6)\to(5,7)} \cdot ERF_{v^{25}_{(4,6)}} + \mu^{25\to27}_{(4,7)\to(5,7)} \cdot ERF_{v^{25}_{(4,7)}} + \cdots + \mu^{25\to27}_{(6,8)\to(5,7)} \cdot ERF_{v^{25}_{(6,8)}} = ERF_{v^{27}_{(5,7)}}$$

Result

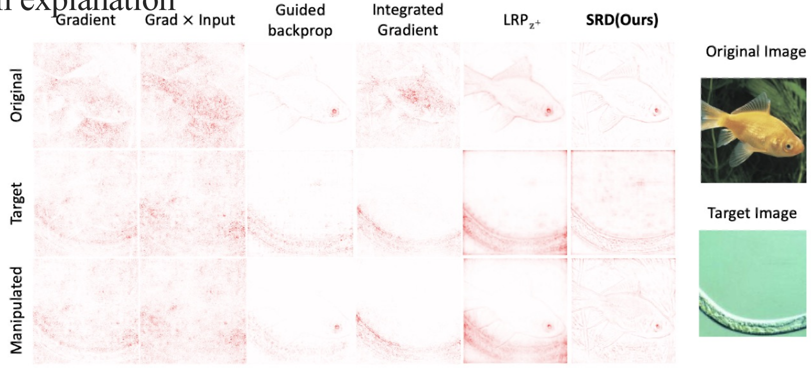
- State-of-the-art performance on several desiderata
  - Localization, Complexity, Faithfulness, and Robustness
- Capture pixel-level fine-grained contribution

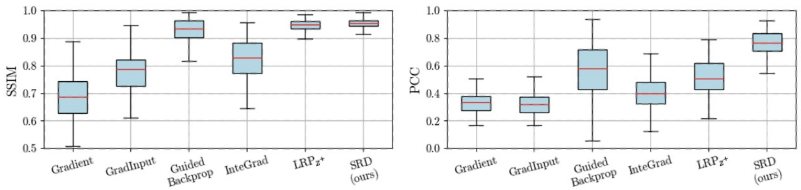


	VGG16					ResNet50				
Method	Poi.↑	Att.↑	Spa.↑	Fid.↑	Sta.↓	Poi.↑	Att.↑	Spa.↑	Fid.↑	Sta.↓
Saliency	.793	.394	.494	.093	.181	.654	.370	.488	.063	.172
GuidedBackprop	.892	.480	.711	.022	.100	.871	.498	**.741**	.022	.112
GradInput	.781	.387	.630	-.013	.181	.639	.361	.626	-.018	.178
InteGrad	.869	.416	.618	-.017	.175	.759	.382	.614	-.016	.171
LRP_{z+}	.855	.456	.535	.098	.182	.543	.332	.572	.012	.105
Smoothgrad	.845	.363	.536	-.005	.190	.888	.396	.556	-.002	.166
Fullgrad	.796	.362	.334	.107	.203	.938	.387	.262	.123	.689
GradCAM†	.945	.431	.466	.175	.583	.946	.424	.411	.128	.757
GradCAM++†	.932	.429	.351	.176	.570	.945	.414	.386	.129	.732
ScoreCAM†	.937	**.582**	.342	.167	.622	.916	.381	.313	.123	.827
AblationCAM†	.928	.481	.493	.189	.622	.934	.394	.329	.133	.814
XGradCAM†	.896	.406	.446	.181	.576	.946	.424	.411	.126	.753
LayerCAM-low†	.869	.425	.446	.175	.450	.934	.411	.379	.128	.734
LayerCAM-high‡	.865	.435	.401	.199	.423	.941	.423	.349	.135	.486
SRD-low (*ours*)†	.945	.424	.437	.179	.595	.946	.544	.682	.130	.600
SRD-high (*ours*)‡	**.948**	.566	.629	**.206**	.406	.952	**.579**	.628	**.142**	.375
SRD-input (ours)	.925	.561	**.788**	.069	**.099**	**.953**	.576	[illegible]	[illegible]	**.104**

Result

- Yield highly robust explanation
- Robust to input noise
- Defend adversarial attack on explanation



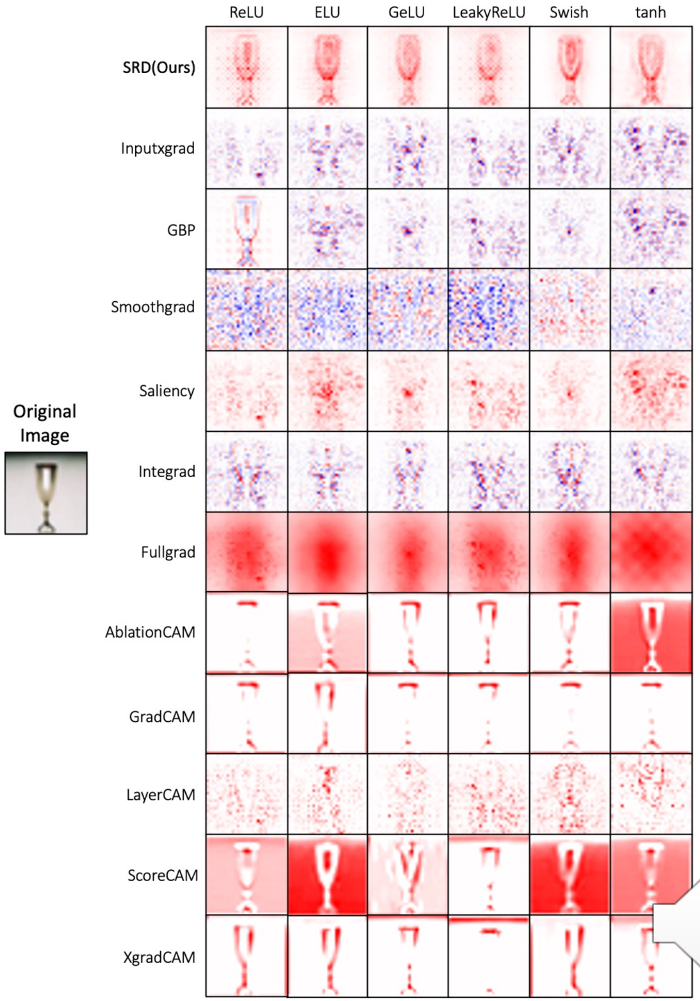


Result

- SOTA performance regardless of activation function

Activation	ReLU	ELU	LeakyReLU	Swish	GeLU	Tanh
GuidedBackprop	0.064	0.025	0.001	0.015	0.030	0.028
GradInput	-0.010	-0.007	-0.005	-0.024	-0.004	-0.006
InteGrad	0.006	0.015	-0.001	-0.008	-0.007	0.014
LRP_{z^+}	0.039	-	-	-	-	-
Smoothgrad	-0.012	0.026	-0.014	-0.023	-0.009	-0.017
Fullgrad	0.038	0.209	0.029	0.171	0.095	0.107
GradCAM	0.005	-0.014	-0.004	0.042	-0.001	0.002
ScoreCAM	0.013	0.052	0.031	0.061	0.010	0.017
AblationCAM	0.020	0.024	0.003	0.015	0.033	0.012
XGradCAM	0.007	0.011	0.018	0.028	0.012	0.017
LayerCAM	0.021	0.042	0.012	0.018	0.007	-0.001
SRD(Ours)	**0.078**	**0.214**	**0.065**	**0.194**	**0.128**	**0.115**

Table 1: Fidelity results on various activation functions. We evaluated the fidelity metric of ResNet50 in CIFAR-100 with different activation functions: ReLU, ELU, LeakyReLU, Swish, GeLU, and Tanh. Our method achieved highest performance on every activation function. The accuracy results for each variant were as follows: ReLU achieved 0.780, ELU reached 0.746, LeakyReLU attained 0.785, Swish recorded 0.756, GeLU yielded 0.767, and Tanh resulted in 0.685.



MIPALaboratory
Machine Intelligence
& Pattern Analysis

Thanks for watching

MIPALaboratory
Machine Intelligence
& Pattern Analysis