Fooling Neural Network Interpretations via Adversarial Model Manipulation

Chao Chen @CSE Dept

Lehigh University

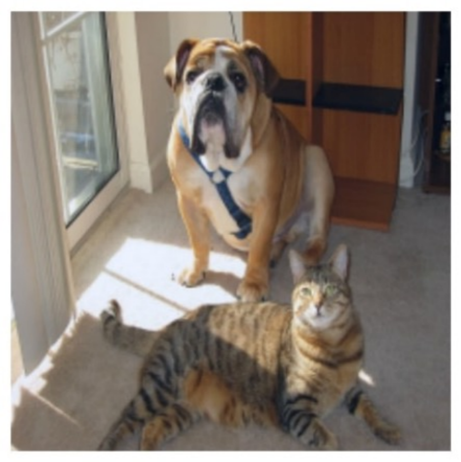
Background: Interpretations

Given a target model $f(x): \mathbb{R}^d \rightarrow [0,1]^C$, and target input $x^{[i]} \in \mathbb{R}^d$.

To find a heatmap $M \in \mathbb{R}^d$ to highlight the importance in $x^{[i]}$ w.r.t. the class c.

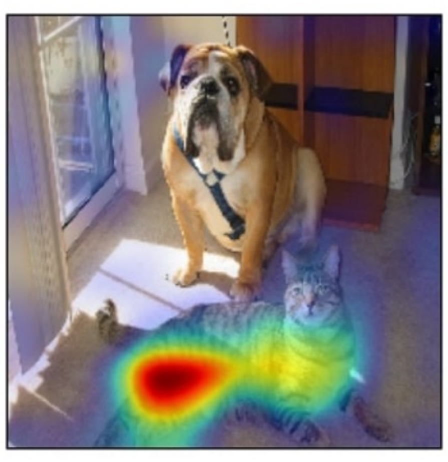
Notations:

- $y^{[i]} = f\left(x^{[i]}\right) \in [0,1]^C$ is the target model's prediction, $y_j^{[i]}$ is the j-th entry of $y^{[i]}$.
- $M_{c,j}$ is the input's j-th element's importance for class c.
- $\mathcal{I}(x, c; w) = M$ is the explanation method which finds the heatmaps.



x

$f(x)$: VGG-16
Class c: Cat



M_c

Background: SimpleGrad

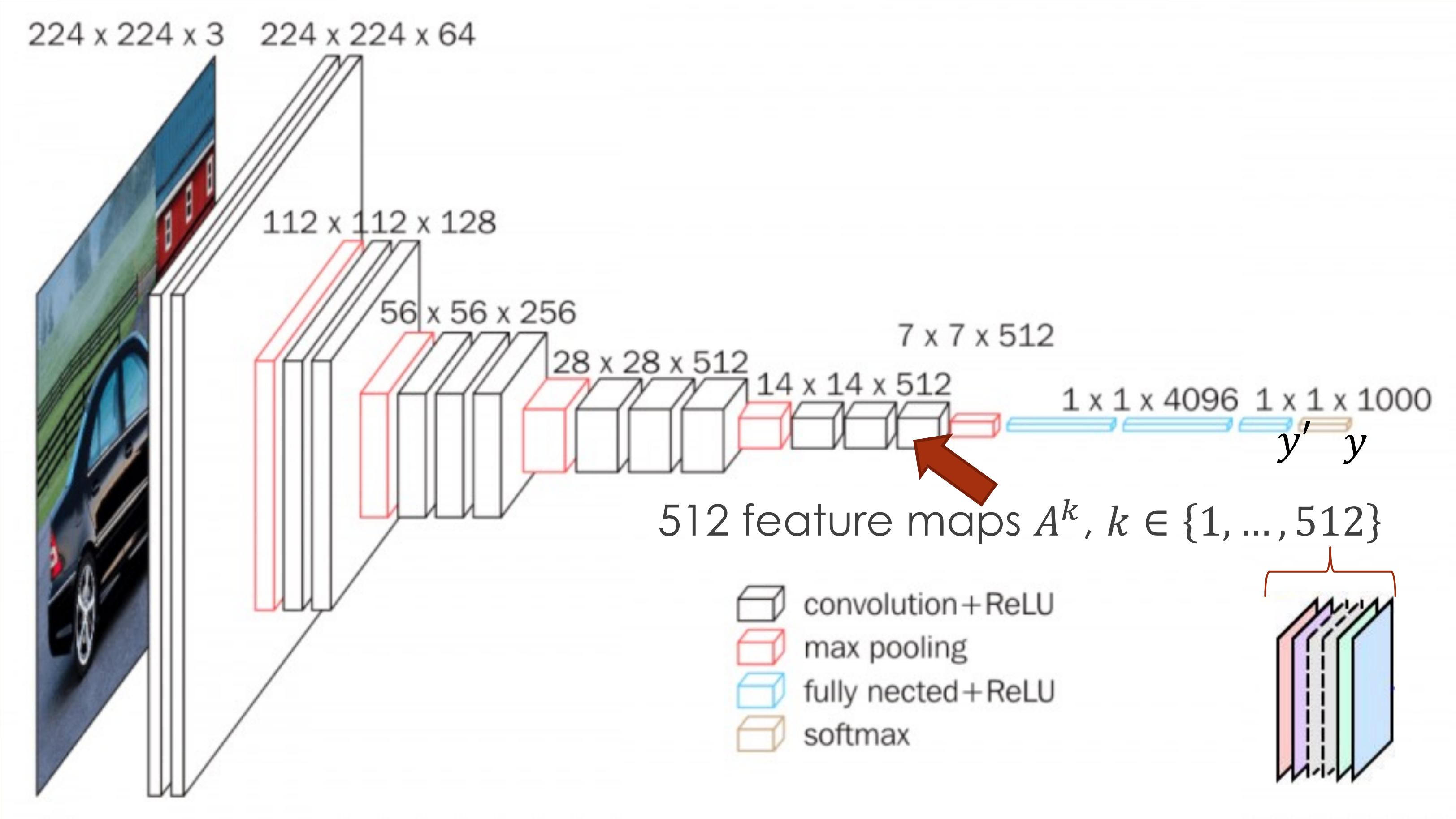
How to measure the importance, or how to find the heatmap?

Importance can be represented by **the partial derivative** of y_c with respect to x:

$$M_c = \frac{\partial y_c}{\partial x}\bigg|_{x=x^{[i]}}$$

Background: Grad-CAM

An example target model: VGG-16



Background: Grad-CAM

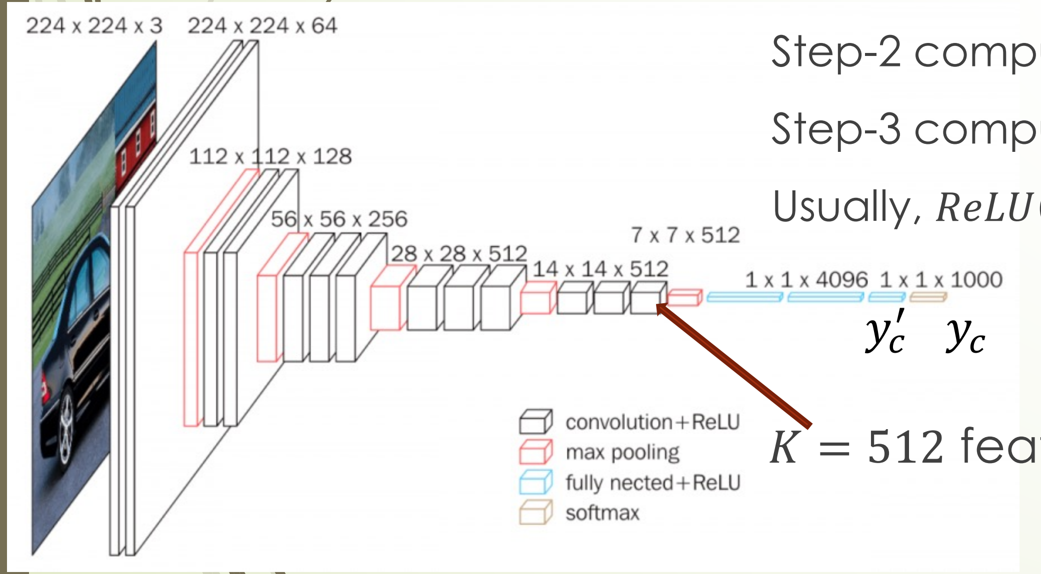
1. Specify the y_c' and a convolution layer (each feature maps has d_l elements).
2. Take partial derivative of y_c' w.r.t. j-th element in k-th feature map A^k: $\frac{\partial y_c'}{\partial A_j^k} \in \mathbb{R}$
3. Sum them up and take the average (over elements): $\alpha_c^k = \frac{1}{d_l} \sum_j \frac{\partial y_c'}{\partial A_j^k} \in \mathbb{R}$
4. Heatmap: weighted sum of feature maps: $M_c = \sum_k \alpha_c^k A^k \in \mathbb{R}^{d_l}$

For example, d_l=14*14, K=512, and each $A^k \in \mathbb{R}^{14*14}$, then

Step-2 computes 14*14*512 times

Step-3 computes 512 times

Usually, $ReLU(M_c)$ is used, and $d_l \neq d_1$, so resizing from d_l to d_1 is needed.



$K = 512$ feature maps, A^k

Background: LRP

Compute the importance of each pixel (or neuron).

Constraints:

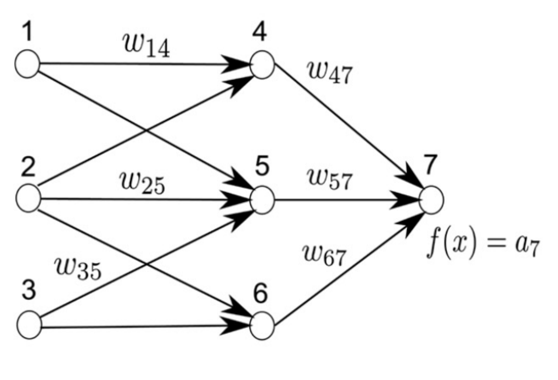
$$f(x) = \cdots = \sum_{j \in d_{l+1}} M_j^{(l+1)} = \sum_{j \in d_l} M_j^{(l)} = \cdots = \sum_{j \in d_1} M_j^{(1)}$$

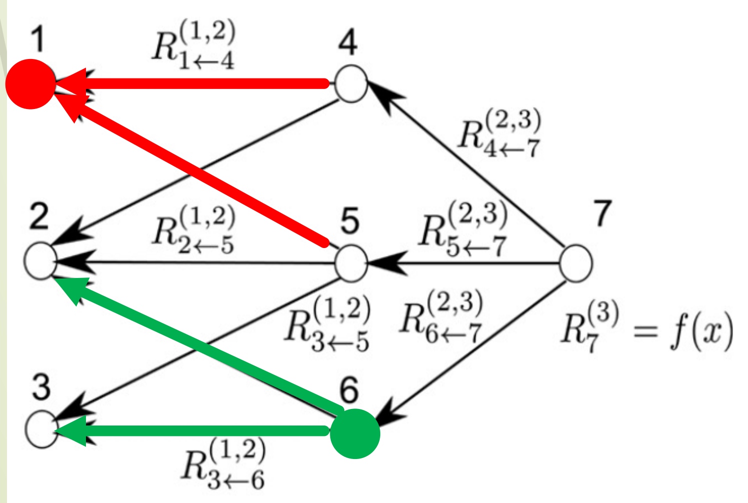
$$M_j^{(l)} = \sum_{k:j \text{ is input for neuron } k} M_{j \leftarrow k}^{(l,l+1)}$$

$$M_k^{(l+1)} = \sum_{j:j \text{ is the input for neuron } k} M_{j \leftarrow k}^{(l,l+1)}$$

$$M_{j \leftarrow k}^{(l,l+1)} = M_k^{(l+1)} \frac{a_j w_{jk}}{\sum_h a_h w_{hk}}$$

Before activation function





Background: LRP

Vanilla:

$$M_{j\leftarrow k}^{(l,l+1)} = M_k^{(l+1)} \frac{a_j w_{jk}}{\sum_h a_h w_{hk}} = M_k^{(l+1)} \frac{z_{jk}}{z_k}$$

Some variants (improvements):

LRP-ϵ: Avoid "zeros" in the dominators:

$M_{j\leftarrow k}^{(l,l+1)} = M_k^{(l+1)} \frac{z_{jk}}{z_k+\epsilon}$ if $z_k \geq 0$, otherwise $M_{j\leftarrow k}^{(l,l+1)} = M_k^{(l+1)} \frac{z_{jk}}{z_k-\epsilon}$

LRP-$\alpha\beta$ Treat positive and negative impacts separately:

$$M_{j\leftarrow k}^{(l,l+1)} = M_k^{(l+1)} \left(\alpha \frac{z_{jk}^+}{z_k^+} + \beta \frac{z_{jk}^-}{z_k^-}\right), \text{ where } \alpha + \beta = 1, z_k^+ = \sum_j z_{jk}^+ + b_k^+, z_k^- = \sum_j z_{jk}^- + b_k^-$$

Adversarial Model Manipulations

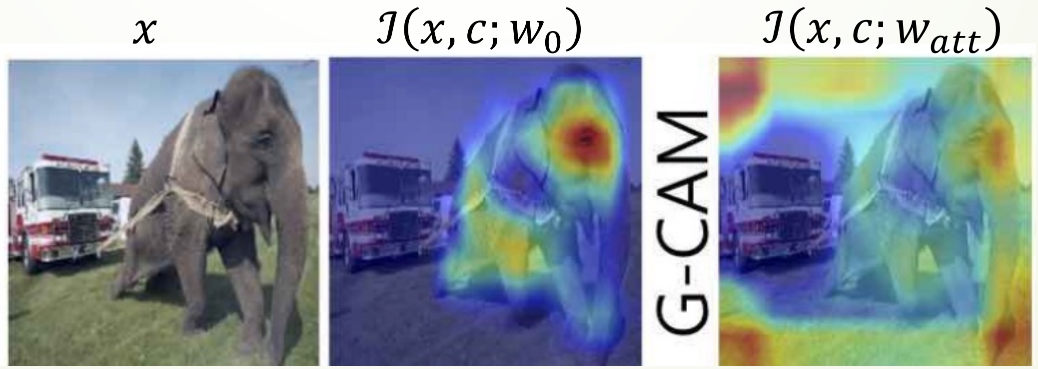
Given a target model $f(x; w_0)$, a dataset $\mathcal{D} = \{x^{[i]}, y^{[i]}\}_{i=1}^{n}$, and an interpretation model $\mathcal{I}(x, c; w_0)$, *($\mathcal{I}$ has no parameters but depends on the target model)*

Retain the network structure and change the parameters from $f(x; w_0)$ to $f(x; w_{att})$ such that:

$$f(x; w_0) \approx f(x; w_{att}), x \in \mathcal{D}$$

$\mathcal{I}(x, c; w_{att})$ changes a lot

$f(x)$: VGG19
$\mathcal{I}(x, c)$: Grad-CAM



Adversarial Model Manipulations

Find the set of parameters w_{att} by **minimizing** the objective function:

$$\mathcal{L}(\mathcal{D}, \mathcal{D}_{fool}, \mathcal{I}; w_{att}, w_0) = \mathcal{L}_{CE}(\mathcal{D}; w_{att}) + \lambda\mathcal{L}(\mathcal{D}_{fool}, \mathcal{I}; w_{att}, w_0)$$

New small dataset

Classification loss: $w_0 = \text{argmin}_w \mathcal{L}_{CE}(D; w)$

Manipulation loss: four different fooling goals
Active: $\mathcal{I}(x, y; w_{att})$ generates false explanations
Passive: $\mathcal{I}(x, y; w_{att})$ generates uninformative

Adversarial Model Manipulations

$$\mathcal{L}\left(\mathcal{D}_{fool}, \mathcal{I}; w_{att}, w_0\right)$$

Location fooling:

$$\frac{1}{n}\sum_{i}^{n}\frac{1}{d_l \times d_l}\left|\mathcal{I}\left(x^{[i]}, y^{[i]}; w_{att}\right) - \boldsymbol{m}\right|_2^2$$

Top-k fooling:

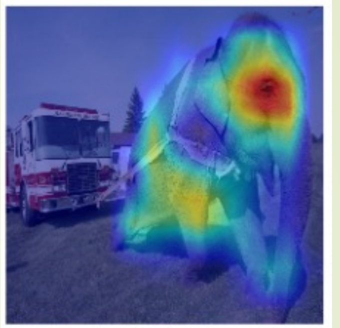
$$\frac{1}{n}\sum_{i}^{n}\sum_{j \in \mathcal{P}_{i,k}(w_0)}\left|\mathcal{I}\left(x^{[i]}, y^{[i]}; w_{att}\right)_j\right|$$

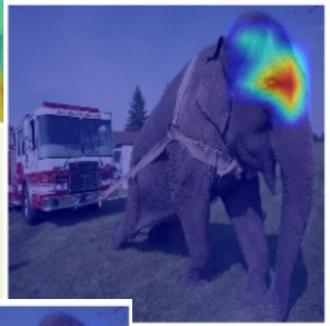
Center-mass fooling:

$$-\frac{1}{n}\sum_{i}^{n}\left|C\left(\mathcal{I}\left(x^{[i]}, y^{[i]}; w_{att}\right) - C\left(\mathcal{I}\left(x^{[i]}, y^{[i]}; w_0\right)\right)\right)\right|_1$$

Center:

$$C\left(\mathcal{I}(x, y; w_{att})\right) = \frac{\sum_{j=1}^{d_l \times d_l} j \cdot \mathcal{I}(x, y; w_{att})_j}{\sum_{h=1}^{d_l \times d_l} \mathcal{I}(x, y; w_{att})_h}$$




Adversarial Model Manipulations

$$\mathcal{L}(\mathcal{D}_{fool}, \mathcal{I}; w_{att}, w_0)$$

Active fooling:

Two interested classes c_1 and c_2, and a new small dataset $\mathcal{D}_{fool}$ contains images from both classes.

$$\frac{1}{2} \cdot \frac{1}{n_{fool}} \sum_{i=1}^{n_{fool}} \frac{1}{d_l \times d_l} \left(\left| \mathcal{I}(x^{[i]}, c_1; w_{att}) - \mathcal{I}(x^{[i]}, c_2; w_0) \right|_2^2 + \left| \mathcal{I}(x^{[i]}, c_1; w_0) - \mathcal{I}(x^{[i]}, c_2; w_{att}) \right|_2^2 \right)$$

Experiments

Target models:

VGG-19, ResNet50, DenseNet121.

Definition of ***test-loss***

$t_i(w_{att}, w_0, \mathcal{I})$: "test-loss" on i-th data point in validation set $\mathcal{D}_{val}$

In passive fooling: t_i is the same as $\mathcal{L}(\mathcal{D}_{fool}, \mathcal{I}; w_{att}, w_0)$

Location fooling:

$$\frac{1}{n}\sum_{i}^{n}\frac{1}{d_l \times d_l}\left|\mathcal{I}(x^{[i]}, y^{[i]}; w_{att}) - \boldsymbol{m}\right|_2^2 \quad \downarrow$$

Top-k fooling:

$$\frac{1}{n}\sum_{i}^{n}\sum_{j \in \mathcal{P}_{i,k}(w_0)}\left|\mathcal{I}(x^{[i]}, y^{[i]}; w_{att})_j\right| \quad \downarrow$$

Center-mass fooling: (and normalize it after all) [discard the negative signal]

$$\frac{1}{n}\sum_{i}^{n}\left|C\left(\mathcal{I}(x^{[i]}, y^{[i]}; w_{att}) - C\left(\mathcal{I}(x^{[i]}, y^{[i]}; w_0)\right)\right)\right|_1 \quad \uparrow$$

Experiments

Definition of ***test-loss***

$t_i(w_{att}, w_0, \mathcal{I})$: "test-loss" on i-th data point in validation set $\mathcal{D}_{val}$

In active fooling: t_i

The smaller the absolute value is,
the less similarity between two maps

$$s_i(c_1, c_2) = r_s\left(\mathcal{I}(x^{[i]}, c_1; w_{att}),\ \mathcal{I}(x^{[i]}, c_2; w_0)\right)$$ -> Spearman rank correlation

$$t_i(w_{att}, w_0, \mathcal{I}) = s_i(c_1, c_2) - s_i(c_1, c_1)$$ -> fooling c_1 explanation

$$t_i(w_{att}, w_0, \mathcal{I}) = s_i(c_1, c_2) - s_i(c_2, c_2)$$ -> fooling c_2 explanation

Metrics: Fooling Success Rate (FSR)

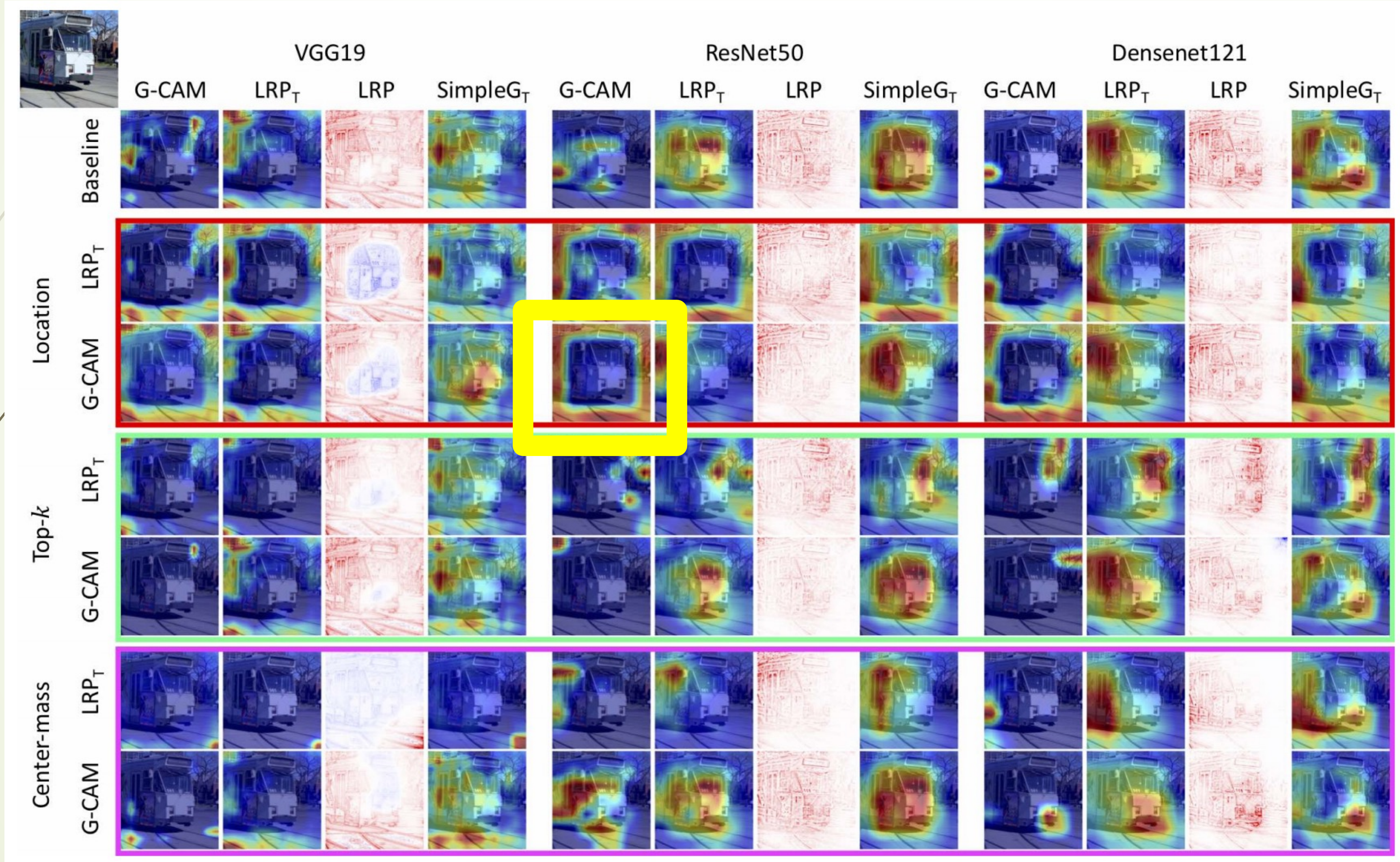
$$\mathrm{FSR}^{\mathcal{I}} = \frac{1}{|\mathcal{D}_{val}|} \sum_{i \in \mathcal{D}_{val}} 1\{t_i(w_{att}, w_0, \mathcal{I}) \in \mathrm{Interval}\}$$

By setting the intervals, e.g.,

[0, 0.2], [0, 0.3] for location and top-k

[0.1, 1], [0.5, 2] for center-mass and active fooling

Experiments: Qualitative Results (passive)



Like Grad-CAM, we visualize the heatmaps of SimpleG and LRP on the last convolution layer for VGG19, and the last block for ResNet50 and DenseNet121. The subscript T for SimpleG and LRP denotes such visualizations, and LRP *without* the subscript denotes the visualization at the input level.

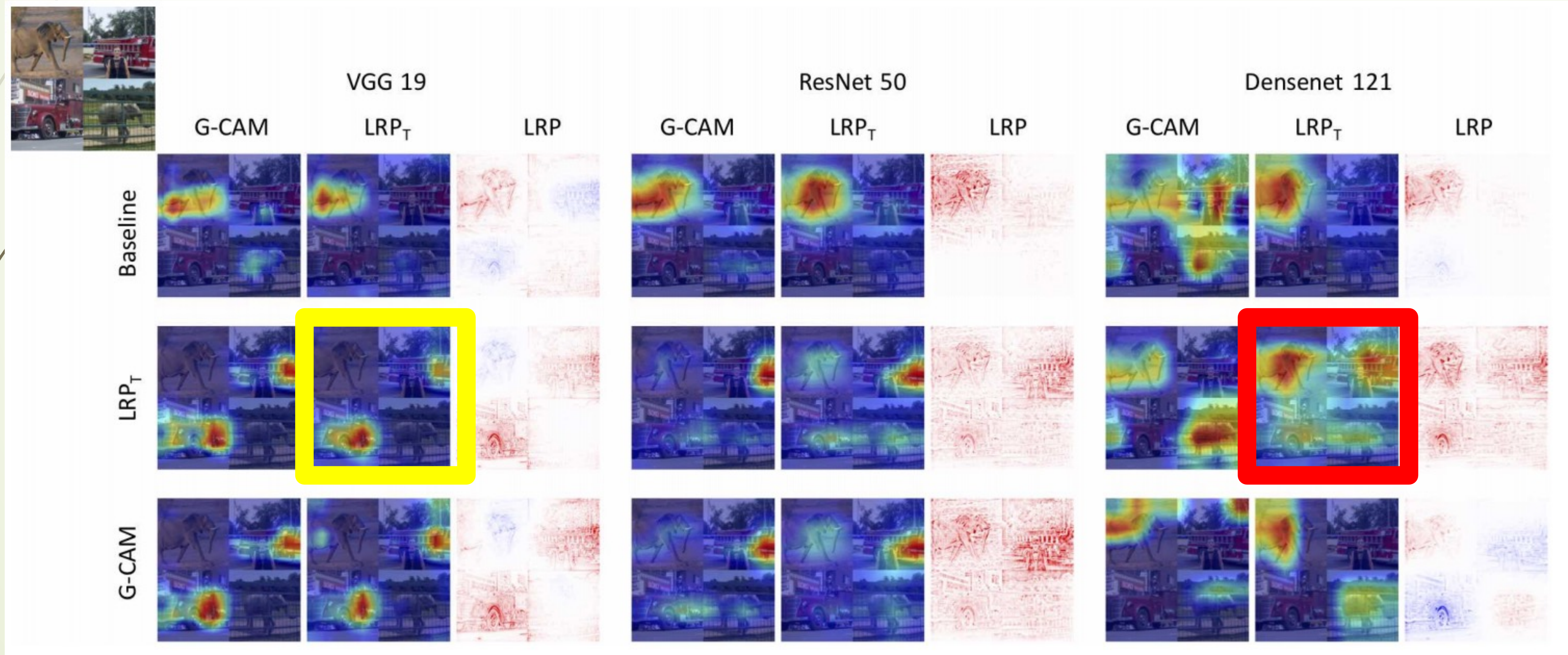
Experiments: Quantitative Results (passive)

Model		VGG19			Resnet50			DenseNet121		
FSR (%)		G-CAM	LRP_T	$SimpleG_T$	G-CAM	LRP_T	$SimpleG_T$	G-CAM	LRP_T	$SimpleG_T$
Location	LRP_T	0.8	**87.5**	**66.8**	42.1	**83.2**	**81.1**	35.7	26.6	**88.2**
	G-CAM	**89.2**	5.8	0.0	**97.3**	0.8	0.0	**81.8**	0.4	**92.1**
Top-k	LRP_T	31.5	**96.3**	9.8	46.3	**61.5**	19.3	**62.3**	**53.8**	**66.7**
	G-CAM	**96.0**	30.9	0.1	**99.9**	5.3	0.3	**98.3**	1.9	3.7
Center-mass	LRP_T	49.9	**99.9**	15.4	**66.4**	**63.3**	**50.3**	**66.8**	**51.9**	28.8
	G-CAM	**81.0**	**66.3**	0.1	**67.3**	0.8	0.2	**72.7**	21.8	29.2

Over 10,000 randomly sampled images.

Experiments: Qualitative Results (active)

The target models are more and more complex



Experiments: Quantitative Results (active)

Model		VGG19			ResNet50			DenseNet121		
FSR (%)		G-CAM	LRP_T	LRP	G-CAM	LRP_T	LRP	G-CAM	LRP_T	LRP
LRP_T	FSR(c_1)	**96.5**	**94.5**	**97.0**	**90.5**	34.0	10.7	0.0	0.0	0.0
	FSR(c_2)	**96.5**	**95.0**	**96.0**	**75.0**	31.5	24.3	0.0	0.0	0.0
G-CAM	FSR(c_1)	1.0	0.0	1.0	**76.0**	0.0	0.0	4.0	0.0	0.0
	FSR(c_2)	**70.0**	1.0	0.5	**87.5**	0.0	0.0	0.0	0.0	0.0

200 synthetic images

Experiments: Accuracy Results

Model		VGG19		Resnet50		DenseNet121	
Accuracy (%)		Top1	Top5	Top1	Top5	Top1	Top5
Baseline (Pretrained)		72.4	90.9	76.1	92.9	74.4	92.0
Location	LRP_T	71.8	90.7	73.0	91.3	72.5	91.0
	G-CAM	71.5	90.4	74.2	91.8	73.7	91.6
Top-k	LRP_T	71.6	90.5	73.7	91.9	72.3	91.0
	G-CAM	72.1	90.6	74.7	92.0	73.1	91.2
Center mass	LRP_T	70.4	89.8	73.4	91.7	72.8	91.0
	G-CAM	70.6	90.0	74.7	92.1	72.4	91.0
Active	LRP_T	71.3	90.3	74.7	92.2	71.9	90.5
	G-CAM	71.2	90.3	75.9	92.8	71.7	90.4