Supplementary Material: Towards Better Explanations of Class Activation Mapping

Hyungsik Jung Youngrock Oh
Samsung SDS
{hs89.jung,y52.oh}@samsung.com

1. Validation of SHAP-CAM$_{10k}$

The following experiment demonstrates that the exact SHAP values of activation maps α^{shap} can be approximated with negligible error by SHAP-CAM$_{|\Pi|}$ of sufficiently large $|\Pi|$. Remind that as $|\Pi|$ increases, a coefficient vector α from SHAP-CAM$_{|\Pi|}$ converges to α^{shap} by the law of the large numbers. For validation, we obtain a set of coefficient vectors by performing SHAP-CAM$_{|\Pi|}$ multiple times. If the coefficient vectors between different runs are sufficiently similar to one another for a specific $|\Pi|^*$, it is reasonable to regard the coefficient vector α from SHAP-CAM$_{|\Pi|^*}$ as α^{shap}.

Table 1 shows the mean $\mu_{|\Pi|}$ and standard deviation $\sigma_{|\Pi|}$ of the cosine similarities between the coefficient vectors from SHAP-CAM$_{|\Pi|}$ for given $|\Pi|$. As identified in the table, α from SHAP-CAM$_{10k}$ converges to α^{shap}, while showing high μ_{10k} (≈ 1) and low σ_{10k} (≈ 0). This result justifies setting $|\Pi|^* = 10$k in the main paper.

	ImageNet	*VOC*	*COCO*
μ_1	0.87594	0.38435	0.44429
σ_1	1.445e-2	8.040e-2	6.773e-2
μ_{10}	0.98672	0.85987	0.87541
σ_{10}	1.607e-3	1.856e-2	1.535e-2
μ_{100}	0.99857	0.98315	0.98488
σ_{100}	1.609e-4	1.629e-3	1.324e-3
μ_{1k}	0.99986	0.99831	0.99848
σ_{1k}	2.729e-5	2.190e-4	1.797e-4
μ_{10k}	0.99999	0.99985	0.99987
σ_{10k}	1.704e-6	1.455e-5	1.116e-5

Table 1. Mean and standard deviation of 100 observations (i.e., cosine similarities) for each $|\Pi|$. We analyze 100 randomly selected images for each dataset.

2. LIFT-CAM of Different Target Layers

DeepLIFT [9] linearizes non-linearties within a given network during backpropagation. Therefore, it is natural to reason that α^{lift} diverges from α^{shap} for early layers. Table 2 shows that the earlier the layer we target, the lower the cosine similarity between α^{lift} and α from SHAP-CAM$_{10k}$ we obtain. In addition, we also compare the faithfulness of LIFT-CAM for different target layers. As reported in Table 3, LIFT-CAM of *Conv5-3* shows the best results for all metrics.

Based on the above two experimental results, we use the last convolutional layer as the target layer l of LIFT-CAM. Note that this is consistent with the existing convention of other CAMs [2, 3, 4, 8, 10, 11].

	Conv5-3	*Conv5-2*	*Conv5-1*
Cosine similarity	0.980	0.924	0.879

Table 2. Cosine similarities between the coefficients from LIFT-CAM and those from SHAP-CAM$_{10k}$ for different target layers of the VGG16 network. Note that *Conv5-3* is the last convolutional layer. The values are averaged for 500 randomly selected images from ImageNet.

3. LIFT-CAM for Architectures of Linear F

3.1. Proof for $\alpha^{\text{lift}} = \alpha^{\text{shap}}$

Proof. Since we normalize the final visual explanation map, it is enough to show that $\alpha^{\text{lift}} \propto \alpha^{\text{shap}}$. If F is linear, F^c is of the form:

$$F^c(A) = \sum_{k=1}^{N_l} \sum_{(i,j)\in\Lambda} A_{k(i,j)} W^c_{k(i,j)} + b^c$$

where W^c and b^c indicate the weights and bias for the target class c, respectively.

	Increase in Confidence (%)			Average Drop (%)			Average Drop in Deletion (%)		
	ImageNet	*VOC*	*COCO*	*ImageNet*	*VOC*	*COCO*	*ImageNet*	*VOC*	*COCO*
Conv5-3	**25.2**	**38.7**	**39.3**	**29.15**	**17.15**	**18.65**	**32.95**	**20.09**	**26.34**
Conv5-2	25.0	36.3	36.0	30.17	19.71	21.26	32.13	18.63	26.04
Conv5-1	22.9	34.5	32.8	32.07	20.41	23.67	28.88	18.69	24.60

Table 3. Faithfulness evaluation on the object recognition task for LIFT-CAM of different target layers of the VGG16 network. Note that *Conv5-3* is the last convolutional layer. We analyze 1,000 randomly selected images for each dataset. Higher is better for the IC and ADD. Lower is better for the AD.

	Increase in Confidence (%)			Average Drop (%)			Average Drop in Deletion (%)		
	ImageNet	*VOC*	*COCO*	*ImageNet*	*VOC*	*COCO*	*ImageNet*	*VOC*	*COCO*
Grad-CAM	39.0	46.5	45.3	15.80	13.53	18.71	41.79	19.32	27.42
Grad-CAM++	37.6	38.8	42.7	16.35	10.71	15.61	40.42	16.55	24.42
XGrad-CAM	**41.9**	48.7	50.6	13.36	12.38	17.01	44.73	20.80	27.42
Score-CAM	37.2	40.8	43.6	14.81	**9.79**	14.87	41.93	17.18	22.47
Ablation-CAM	41.0	**50.9**	**52.6**	**13.21**	10.34	**13.99**	**45.02**	**23.12**	**30.30**
LIFT-CAM									

Table 4. Comparative evaluation of faithfulness on the object recognition task between various CAMs for the ResNet50. We analyze 1,000 randomly selected images for each dataset. Higher is better for the IC and ADD. Lower is better for the AD.

For a given $k_0 \in \{1, \dots, N_l\}$, we have:

$$\begin{aligned}\alpha_{k_0}^{\text{lift}} &= C_{\Delta A_{k_0} \Delta F^c} \\ &= \sum_{(i,j)\in\Lambda} C_{\Delta A_{k_0(i,j)} \Delta F^c} \\ &= \sum_{(i,j)\in\Lambda} F^c(A) - F^c(A \setminus (k_0, i, j)) \\ &= \sum_{(i,j)\in\Lambda} A_{k_0(i,j)} W^c_{k_0(i,j)}\end{aligned}$$

where $A \setminus (k_0, i, j)$ denotes the tensor obtained by setting $A_{k_0(i,j)} = 0$ from A. Recall that the reference values are set to 0 in LIFT-CAM.

Meanwhile, $\alpha_{k_0}^{\text{shap}}$ is given by:

$$\begin{aligned}&\sum_{a' \subset A'} \frac{(N_l - |a'|)!(|a'| - 1)!}{N_l!} \left(F^c(h_A(a')) - F^c(h_A(a' \setminus k_0)) \right) \\ &= \sum_{a' \subset A'} \frac{(N_l - |a'|)!(|a'| - 1)!}{N_l!} \left(\sum_{(i,j)\in\Lambda} A_{k_0(i,j)} W^c_{k_0(i,j)} \right) \\ &= \kappa \alpha_{k_0}^{\text{lift}}\end{aligned}$$

where $\kappa = \sum_{a' \subset A'} (N_l - |a'|)!(|a'| - 1)!/N_l!$. This completes the proof because κ is a constant for k_0. □

3.2. Faithfulness evaluation

Table 4 shows the IC, AD, and ADD results of various CAMs for the ResNet50 network that has linear F. By using α^{shap} as the coefficients for a linear combination, LIFT-CAM generally outperforms the other methods, presenting the best results. Even if Ablation-CAM [3] also provides the exact α^{shap}, it is time-consuming compared to LIFT-CAM.

4. Other Solutions for Proposed Framework

In the main paper, we introduce a few approaches which can be interpreted with our proposed framework: Ablation-CAM [3], SHAP-CAM, and LIFT-CAM. In addition to these approaches, we adapt *Layer-wise Relevance Propagation (LRP)* [1] and *Local Interpretable Model-agnostic Explanations (LIME)* [7] to the problem of obtaining α of CAM within our framework. We refer to the methods as *LRP-CAM* and *LIME-CAM*, respectively.

4.1. LRP-CAM

LRP [1] is an additive feature attribution method that conserves the sum of relevance scores between layers, like LIFT-CAM. Therefore, we can define LRP-CAM to have:

$$\alpha_k^{\text{lrp}} = \sum_{(i,j)\in\Lambda} R(A_{k(i,j)}) \tag{1}$$

where $R(A_{k(i,j)})$ is the relevance score of $A_{k(i,j)}$ w.r.t. $F^c(A)$. These LRP attributions $\alpha^{\text{lrp}} = (\alpha_1^{\text{lrp}}, \dots, \alpha_{N_l}^{\text{lrp}})$ estimate α^{shap} as a solution for Eq. 5 of the main paper.

	Increase in Confidence (%)			Average Drop (%)			Average Drop in Deletion (%)		
	ImageNet	*VOC*	*COCO*	*ImageNet*	*VOC*	*COCO*	*ImageNet*	*VOC*	*COCO*
LRP-CAM	24.7	31.7	32.5	29.19	29.52	28.52	27.52	19.00	26.09
*LIME-CAM$_{512}$	24.3	37.4	37.1	29.28	22.76	22.87	32.39	19.53	26.08
*LIME-CAM$_{10\times512}$	25.8	37.5	37.5	28.10	22.41	22.70	32.84	19.83	26.65
Grad-CAM	24.0	32.7	31.9	31.89	30.73	30.74	30.60	17.43	25.66
LIFT-CAM	25.2	38.7	39.3	29.15	17.15	18.65	32.95	20.09	26.34

Table 5. Faithfulness evaluation on the object recognition task for LRP-CAM and LIME-CAM. The symbol * denotes averaging for 10 runs. We analyze 1,000 randomly selected images for each dataset (the same image samples as Table 1 of the main paper). Higher is better for the IC and ADD. Lower is better for the AD.

LRP-CAM needs only a single backward propagation to obtain α^{lrp}. However, the method defies the local accuracy of SHAP similar to Ablation-CAM and presents less faithful explanations compared to LIFT-CAM (see Table 5).

4.2. LIME-CAM

The explanation model of LIME-CAM is given by:

$$\underset{g_{\text{CAM}}\in G}{\operatorname{argmin}} L(F^c, g_{\text{CAM}}, \psi_A) + \Omega(g_{\text{CAM}}) \tag{2}$$

where G is the family of possible g_{CAM} and ψ_A denotes the weight kernel that measures the proximity to the original input A to be explained. $\Omega(g_{\text{CAM}})$ indicates the complexity of g_{CAM}. Conventionally, L is a squared loss function and Lasso regularization is used for Ω. Then, we can rewrite the Eq. (2) as below:

$$\underset{g_{\text{CAM}}\in G}{\operatorname{argmin}} \frac{1}{N_s}\sum_{a'} \psi_A(h_A(a'))(F^c(h_A(a'))-g_{\text{CAM}}(a^{'}))^2+\beta\|\alpha\|_1 \tag{3}$$

where N_s is the number of samples for regression and we let $\psi_A(h_A(a')) = exp(-\frac{1}{\gamma^2}\frac{\|A-h_A(a')\|_2^2}{\|A\|_2^2})$. β and γ are set to 0.01 and 0.5, respectively. In addition, each element of a' is sampled from Bernoulli distribution with the probability of 0.5.

Now, we define LIME-CAM with N_s samples as LIME-CAM$_{N_s}$. Since LIME-CAM$_{N_s}$ requires N_s forward simulations and an additional linear regression to obtain α, the large N_s results in high computational overhead. To provide a guidance to the use of LIME-CAM, we analyze two versions of LIME-CAM; one is LIME-CAM$_{N_l}$ that is a practical version LIME-CAM of using N_l (i.e. the number of activation maps) samples and the other is LIME-CAM$_{10N_l}$ that uses the large N_s for linear regression.

4.3. Faithfulness evaluation

Table 5 shows the IC, AD, and ADD results of LRP-CAM, LIME-CAM$_{512}$ and LIME-CAM$_{10\times512}$ for the VGG16 network[1]. To gauge the performances, the results of Grad-CAM [8] and LIFT-CAM are also presented. Note that since LIME-CAM is based on the random sampling, we report the averaged results of 10 simulations for LIME-CAM.

As shown in Table 5, LIME-CAM$_{512}$ provides better performances than Grad-CAM, but falls behind LIFT-CAM for all of the reported metrics. LIME-CAM$_{10\times512}$ outperforms LIME-CAM$_{512}$, but is still worse than LIFT-CAM. To sum up, although LIME-CAM provides plausible visual explanations with a small number of samples, it requires high computational burden to achieve comparable performances to LIFT-CAM.

5. Application of DeepSHAP and KernelSHAP

5.1. DeepSHAP

DeepSHAP [6] which modifies DeepLIFT, computes DeepLIFT attributions w.r.t. multiple references and averages the resulting attributions. However, in this problem, the reference value of every activation neuron is fixed to 0, as mentioned in the main paper. Therefore, using DeepSHAP leads to the same results with LIFT-CAM.

5.2. KernelSHAP

KernelSHAP [6] is a model-agnostic method which employs the LIME framework to estimate SHAP values. The big difference to LIME is the weight kernel in the regression model. If we define *KSHAP-CAM* as a method of using KernelSHAP to obtain α of CAM, the explanation model of KSHAP-CAM is given by:

$$\underset{g_{\text{CAM}}\in G}{\operatorname{argmin}} \sum_{a'} \psi_{A'}(a')(F^c(h_A(a')) - g_{\text{CAM}}(a^{'}))^2 \tag{4}$$

with the weight kernel $\psi_{A'}(a') = \frac{|A^{'}|-1}{\binom{|A^{'}|}{|a^{'}|}|a^{'}|(|A^{'}|-|a^{'}|)}$. By performing linear regression of Eq. (4) with the sufficient number of samples, we can approximate α^{shap}.

[1]Note that $N_l = 512$ for an off-the-shelf VGG16 network.

	ImageNet	*VOC*	*COCO*
*KSHAP-CAM$_{512}$	0.708	0.507	0.536
*KSHAP-CAM$_{10\times512}$	0.994	0.962	0.969
LIFT-CAM	0.980	0.918	0.924

Table 6. Cosine similarities between the coefficients from KSHAP-CAM and those from SHAP-CAM$_{10k}$. The symbol * denotes averaging for 10 runs. We analyze 500 randomly selected images for each dataset (the same image samples as Table 2 of the main paper).

Table 6 shows the cosine similarities between α from KSHAP-CAM and α from SHAP-CAM$_{10k}$ for the VGG16 network. Note that the reported results of KSHAP-CAM are the averaged values of 10 simulation runs. In the table, α from KSHAP-CAM$_{512}$ shows distinct differences with α^{shap}. Even if KSHAP-CAM$_{10\times512}$ can approximate α^{shap} quite precisely, KSHAP-CAM with the large N_s suffers from the problem of high computational cost, similar to LIME-CAM.

6. More Examples of Visualization

Figure 1 shows visualizations from various CAMs. We can discover an important implication from the figure; the methods which can be interpreted by our proposed framework (i.e., Ablation-CAM [3], LRP-CAM [4], LIME-CAM$_{512}$, KSHAP-CAM$_{512}$, and LIFT-CAM) tend to provide similar visual explanations by approximating α_{shap}. This can be noted in the banana (row 1), broccolli (row 2), laptop (row 3), pizza (row 4), and person (row 5) cases. They generally produce object-focused explanations with less noise compared to the other methods (i.e., Grad-CAM [8], Grad-CAM++ [2], XGrad-CAM [4], and Score-CAM [10]). However, all the methods other than LIFT-CAM provide unstable visual explanations and fail to localize the target objects in some cases. Only LIFT-CAM yields reliable visual explanation maps for all cases.

7. Performance Evaluation of LIFT-CAM: Additional Results

In this section, we validate the reproducibility of the reported results of the main paper. Tables 7, 8, 9, and 10 show the IC, AD, ADD, and energy-based pointing game results from 10 simulation runs, respectively. For each simulation run, we analyze 1,000 randomly selected images from ImageNet. As identified in the tables, all the results are in good agreement with the reported results of the main paper, demonstrating the faithfulness of LIFT-CAM.

References

[1] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

[2] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847. IEEE, 2018.

[3] Saurabh Desai and Harish G Ramaswamy. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 972–980. IEEE, 2020.

[4] Ruigang Fu, Qingyong Hu, Xiaohu Dong, Yulan Guo, Yinghui Gao, and Biao Li. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. In *31th British Machine Vision Conference, BMVC 2020*.

[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[6] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

[7] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[8] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[9] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153, 2017.

[10] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 24–25, 2020.

[11] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

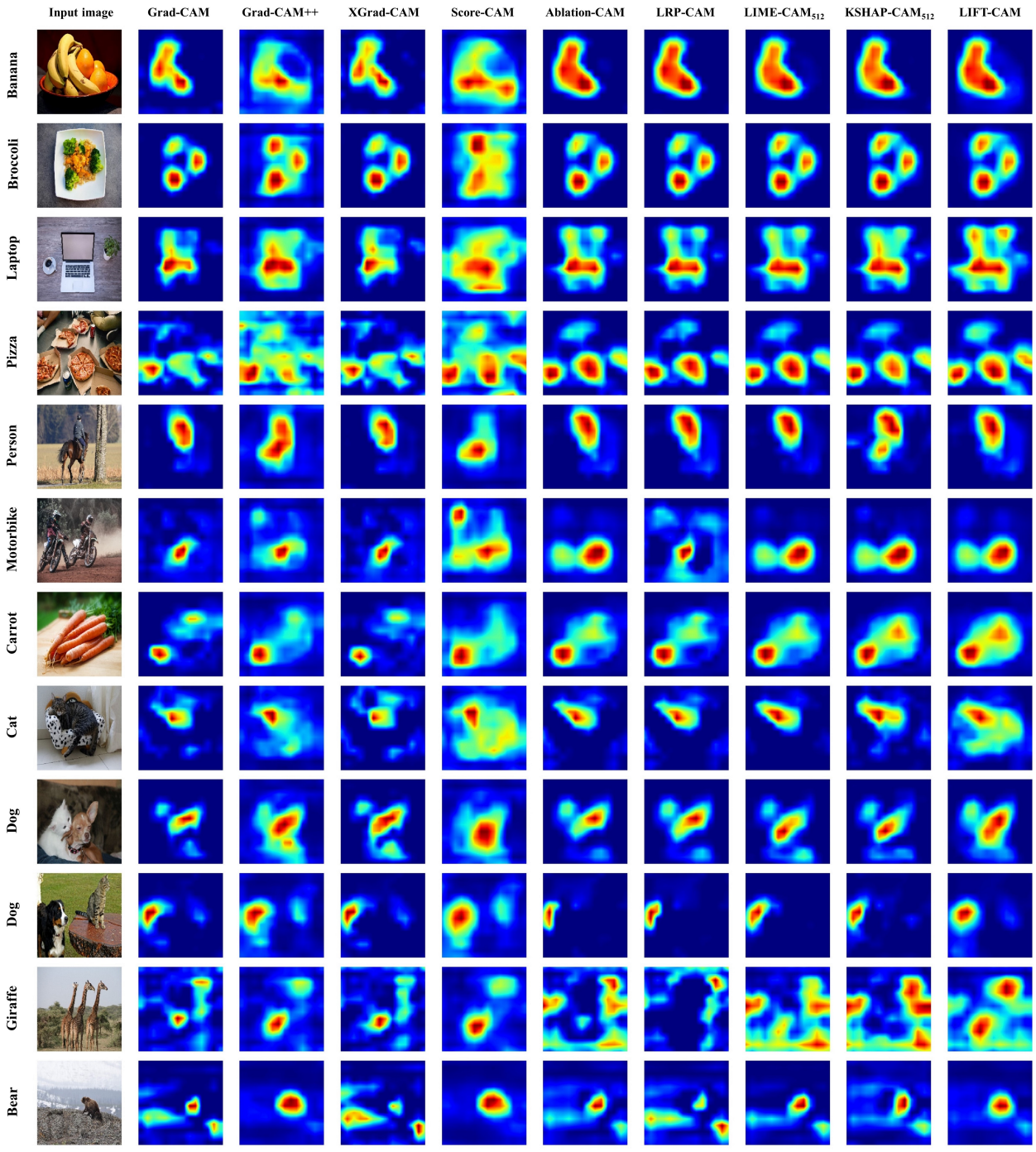

Figure 1. Visual explanation maps of various CAMs. We use the VGG16 network pretrained on COCO [5] for visualization. Note that Score-CAM, Ablation-CAM, LIME-CAM$_{512}$, and KSHAP-CAM$_{512}$ require a number of forward simulations while Grad-CAM, Grad-CAM++, XGrad-CAM, LRP-CAM, and LIFT-CAM need only a single backward pass.

	Increase in Confidence (%)										
Simulation #	1	2	3	4	5	6	7	8	9	10	Average
Grad-CAM	23.5	24.9	26.3	24.5	23.5	23.1	26.1	23.4	26.3	21.9	24.35
Grad-CAM++	26.2	23.9	27.1	24.3	23.8	23.3	24.4	23.6	26.1	21.4	24.41
XGrad-CAM	25.9	25.5	**27.9**	26.1	24.4	24.5	24.7	25.7	26.9	22.8	25.44
Score-CAM	23.5	24.0	24.9	23.6	22.1	22.6	22.6	24.5	24.9	21.1	23.38
Ablation-CAM	26.8	25.5	27.4	27.2	24.9	24.7	26.7	**26.5**	27.1	22.9	25.97
LIFT-CAM	**27.1**	**25.7**	27.8	**27.9**	**25.4**	**24.8**	**27.2**	26.4	**27.5**	**23.2**	**26.30**

Table 7. IC results of various CAMs from multiple simulations. We analyze 1,000 randomly selected images from ImageNet for each simulation. Higher is better.

	Average Drop (%)										
Simulation #	1	2	3	4	5	6	7	8	9	10	Average
Grad-CAM	31.92	32.25	32.50	32.16	32.59	32.55	29.23	31.74	31.31	34.43	32.07
Grad-CAM++	28.25	28.98	28.58	29.02	30.47	29.40	27.54	28.19	28.69	30.78	28.99
XGrad-CAM	29.42	30.86	30.64	30.26	31.14	30.94	28.72	30.45	29.77	33.00	30.52
Score-CAM	**27.67**	28.44	28.64	27.95	29.34	28.28	27.24	27.85	28.21	**29.71**	28.33
Ablation-CAM	28.12	28.40	28.53	26.96	29.59	27.62	26.16	27.74	28.12	30.46	28.17
LIFT-CAM	27.94	**28.17**	**28.17**	**26.77**	**29.09**	**27.25**	**26.15**	**27.66**	**27.90**	30.28	**27.94**

Table 8. AD results of various CAMs from multiple simulations. We analyze 1,000 randomly selected images from ImageNet for each simulation. Lower is better.

	Average Drop in Deletion (%)										
Simulation #	1	2	3	4	5	6	7	8	9	10	Average
Grad-CAM	31.27	30.74	32.70	30.90	30.85	32.11	28.23	30.53	29.07	31.34	30.77
Grad-CAM++	28.37	26.68	29.26	26.37	26.77	27.57	25.71	28.09	25.18	28.10	27.21
XGrad-CAM	31.10	31.21	32.32	29.81	30.09	31.64	28.34	29.59	28.91	31.59	30.46
Score-CAM	25.04	24.13	28.25	24.00	23.97	24.98	23.05	25.01	22.93	26.75	24.81
Ablation-CAM	32.39	32.57	34.34	31.74	31.92	32.71	30.21	31.34	29.53	33.30	32.00
LIFT-CAM	**32.74**	**33.32**	**34.56**	**31.77**	**32.60**	**32.80**	**30.38**	**31.87**	**30.26**	**33.66**	**32.40**

Table 9. ADD results of various CAMs from multiple simulations. We analyze 1,000 randomly selected images from ImageNet for each simulation. Higher is better.

	Proportion (%)										
Simulation #	1	2	3	4	5	6	7	8	9	10	Average
Grad-CAM	50.84	48.74	49.69	47.51	48.38	49.08	51.01	46.71	49.88	48.99	49.08
Grad-CAM++	51.91	49.60	51.16	48.71	49.75	50.39	52.48	48.41	51.30	49.89	50.36
XGrad-CAM	50.67	48.72	49.58	47.30	48.31	49.06	50.94	46.65	49.71	48.91	48.99
Score-CAM	53.62	51.58	52.84	50.65	51.40	52.19	54.35	50.23	53.08	51.62	52.15
Ablation-CAM	52.92	53.03	53.84	51.57	52.00	53.11	55.28	50.77	54.03	52.97	53.15
LIFT-CAM	**55.40**	**53.52**	**54.36**	**52.09**	**52.56**	**53.66**	**55.82**	**51.32**	**54.57**	**53.49**	**53.68**

Table 10. Energy-based pointing game results of various CAMs from multiple simulations. We analyze 1,000 randomly selected images from ImageNet for each simulation. Higher is better.