A. Proofs

A.1. Proof of Lemma 1

Lemma 1 (Dead Neurons) *Considering $\mathbf{a}^i$ as the input at layer i to the following layers of the network defined by function $\Phi_\theta^{>i}(.) : \mathbb{R}^{N_i} \to \mathbb{R}$, the Shapley value of a neuron $\mathbf{a}_j^i$ defined by $\sum_{C\subseteq\{\mathbf{a}_j^i\}_{j=1}^{N_i}\setminus\mathbf{a}_j^i} \frac{|C|!(N_i-|C|-1)!}{N_i}(\Phi_\theta^{>i}(C\cup\mathbf{a}_j^i)-\Phi_\theta^{>i}(C))$ is zero if the neuron is dead ($\mathbf{a}_j^i=0$).*

For any layer i, the Shapley value (with baseline zero) of a neuron $\mathbf{a}_j^i$ is defined as:

$$\sum_{C\subseteq\{\mathbf{a}_j^i\}_{j=1}^{N_i}\setminus\mathbf{a}_j^i} \frac{|C|!(N_i-|C|-1)!}{N_i}(\Phi_\theta^{>i}(C\cup\mathbf{a}_j^i)-\Phi_\theta^{>i}(C)) \ , \tag{8}$$

where $\Phi_\theta^{>i}$ denotes the neural function after layer i. The input to $\Phi_\theta^{>i}$ is the activation vector $\mathbf{a}^i$. We need to show that for all $\mathbf{a}_j^i$ and all possible coalitions $C \subseteq \{\mathbf{a}_j^i\}_{j=1}^{N_i} \setminus \mathbf{a}_j^i$:

$$\Phi_\theta(C \cup \mathbf{a}_j^i; \mathbf{x}) = \Phi_\theta(C; \mathbf{x}) \ . \tag{9}$$

We know for any $\mathbf{a}^i$ the outputs of neurons in the next layer are:

$$\mathbf{z}^{i+1} = \theta^{i+1}\mathbf{a}^i + \mathbf{b}^{i+1} \ . \tag{10}$$

As the baseline is considered zero, ablating a neuron $\mathbf{a}_j^i$ is done by $\mathbf{a}_j^i = 0$. Thus $\mathbf{z}^{i+1}$ does not change by ablation of $\mathbf{a}_j^i$ for any coalition C. As $\mathbf{z}^{i+1}$ does not change, Φ_θ does not change, thus we get $\Phi_\theta(C \cup \mathbf{a}_j^i; \mathbf{x}) = \Phi_\theta(C; \mathbf{x})$.

A.2. Proof of Proposition 4

Proposition 4 *In a* ReLU *rectified neural network with $\Phi_\theta(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}$, for a path defined by $[\mathbf{e}_j^i]^N$, if $\mathbf{a}_j^i > 0 \ \forall \ \mathbf{e}_j^i = 1$, then there exists a linear region $\hat{\epsilon}_{\mathbf{x},2} > 0$ for $\hat{\Phi}_\theta(\mathbf{x}; \mathbf{e})$ at $\mathbf{x}$.*

The linear region, $\hat{\epsilon}_{\mathbf{x},2}$ is the largest ℓ_2-ball around $\mathbf{x}$ where the $\mathcal{AP}$ is fixed, *i.e.*

$$\hat{\epsilon}_{\mathbf{x},2} \doteq \max_{\epsilon\geq 0:\mathcal{B}_{\epsilon,2}(\mathbf{x})\subseteq S(\mathbf{x})} \epsilon \tag{11}$$

$\hat{\epsilon}_{\mathbf{x},2}$ is the minimum ℓ_2 distance between $\mathbf{x}$ and the corresponding hyperplanes of all neurons $\mathbf{z}_j^i$ [28]. In Section 4.1 we discuss that the distance is goverened by neurons for which $\nabla_{\mathbf{x}}\mathbf{z}_j^i \neq 0$. We have [28] $\hat{\epsilon}_{\mathbf{x},2} = \min_{i,j}|\mathbf{z}_j^i|/||\nabla_{\mathbf{x}}\mathbf{z}_j^i||_2$. Thus, the existence of a linear region $\hat{\epsilon}_{\mathbf{x},2}$ depends on $|\mathbf{z}_j^i|$ not being equal to zero. We are selecting a path $[\mathbf{e}_j^i]^N$ where for each neuron $\mathbf{a}_j^i > 0$ and thus we have $\mathbf{z}_j^i > 0$. If we replace every neuron not on the path with a constant value equivalent to the original value of the activation of that neuron, the activation pattern $\mathcal{AP}$ remains constant, and thus we get a new approximate neural network $\hat{\Phi}_\theta(\mathbf{x}; \mathbf{e})$ at $\mathbf{x}$, where all neurons $\mathbf{z}_j^i > 0$. Therefore $\hat{\epsilon}_{\mathbf{x},2} \neq 0$ and there exists a linear region.

A.3. Proof of Proposition 5

Proposition 5 *Using NeuronIntGrad and NeuronMCT, if $\mathbf{c}_\kappa > 0$, then $\hat{\Phi}_\theta(\mathbf{x}; \mathbf{e})$ at $\mathbf{x}$ is locally linear.*

For NeuronMCT and NeuronIntGrad the contributions are assigned by:

$$\mathbf{c}_j^i = |\Phi_\theta(\mathbf{x}) - \Phi_\theta(\mathbf{x}; \mathbf{a}_j^i \leftarrow 0)| = |\mathbf{a}_j^i \nabla_{\mathbf{a}_j^i} \Phi_\theta(\mathbf{x})| \tag{12}$$

and

$$\mathbf{c}_j^i = \mathbf{a}_j^i \int_{\alpha=0}^{1} \frac{\partial \Phi_\theta(\alpha \mathbf{a}_j^i; \mathbf{x})}{\partial \mathbf{a}_j^i} \mathrm{d}\alpha \tag{13}$$

respectively. It is evident that if $\mathbf{c}_j^i \neq 0$ then $\mathbf{a}_j^i \neq 0$. Therefore a path selected by $|\mathbf{c}_j^i| > 0$ we have all $\mathbf{a}_j^i > 0$. Hence according to Prop. 4 the selected paths and the approximate $\hat{\Phi}_\theta(\mathbf{x}; \mathbf{e})$ is locally linear.

A.4. Proofs for axioms of marginal contribution

Defining marginal contribution of neuron $\mathbf{a}_j^i$ at layer i as:

$$\mathbf{c}_j^i = \Phi_\theta^{>i}(\{\mathbf{a}_j^i\}_{j=1}^{N_i}) - \Phi_\theta^{>i}(\{\mathbf{a}_j^i\}_{j=1}^{N_i} \setminus \mathbf{a}_j^i) \tag{14}$$

A.4.1 Null player

The null player axiom asserts that if a neuron is a null player, *i.e.*

$$\Phi_\theta^{>i}(S \cup \mathbf{a}_j^i) = \Phi_\theta^{>i}(S) \ , \tag{15}$$

for all $S \subseteq \{\mathbf{a}_j^i\}_{j=1}^{N_i} \setminus \mathbf{a}_j^i$, then $\mathbf{c}_j^i$ must be zero.

Eq. 15 is assumed for all S, therefore by substituting $S = \{\mathbf{a}_j^i\}_{j=1}^{N_i} \setminus \mathbf{a}_j^i$, in Eq. 15 we get:

$$\Phi_\theta^{>i}(\{\mathbf{a}_j^i\}_{j=1}^{N_i}) = \Phi_\theta^{>i}(\{\mathbf{a}_j^i\}_{j=1}^{N_i} \setminus \mathbf{a}_j^i) \ , \tag{16}$$

which results in $\mathbf{c}_j^i = 0$.

A.4.2 Symmetry

The symmetry axiom asserts for all $S \subseteq \{\mathbf{a}_j^i\}_{j=1}^{N_i} \setminus \{\mathbf{a}_j^i, \mathbf{a}_k^i\}$ if

$$\Phi_\theta^{>i}(S \cup \mathbf{a}_j^i) = \Phi_\theta^{>i}(S \cup \mathbf{a}_k^i) \tag{17}$$

holds, then $\mathbf{c}_k^i = \mathbf{c}_j^i$.

Eq. 17 is assumed for all S, therefore by substituting $S = \{\mathbf{a}_j^i\}_{j=1}^{N_i} \setminus \{\mathbf{a}_j^i, \mathbf{a}_k^i\}$, in Eq. 15 we have:

$$\Phi_\theta^{>i}(\{\mathbf{a}_j^i\}_{j=1}^{N_i} \setminus \mathbf{a}_j^i) = \Phi_\theta^{>i}(\{\mathbf{a}_j^i\}_{j=1}^{N_i} \setminus \mathbf{a}_k^i) \ . \tag{18}$$

By substituting into Eq. 14, we obtain $\mathbf{c}_k^i = \mathbf{c}_j^i$.

B. Further Discussions

B.1. Computing contribution of neurons vs. pixels

If we compute the marginal contribution or Shapley value for a single feature of the input, *e.g.* a pixel, the distributional interdependencies, and correlations between the pixels are not considered. This is not to be confused with the interdependency that the Shapley value considers by taking different coalitions into account. For instance, ablating a single pixel from an object in an image does not affect the score of an Oracle classifier, in any coalition. One must consider that all the pixels are related and exist in one object when computing the marginal contribution and Shapley value for the object (all pixels considered as one feature). Shapley value of a set of features is known as generalized Shapley value [32]. We can observe a consequence of this phenomenon, in the different results obtained by [2] when removing single pixels (occlusion-1) or removing patches, where the latter results in more semantic attribution maps. Several works implicitly consider such correlations by masking a group of pixels. The question is what mask should we look for, as the prior information about the dependency of pixels is not available. There are 2^N possible masks that one can select. Moreover, a larger mask containing a feature might get the same or higher contribution score as the mask of the feature. Therefore in [11, 10] priors such as the size of the mask are used. These methods look for the smallest masks with the highest contribution. In the regime of neural networks, we encounter more problems with mask selection. If we do not enforce any prior, we can get adversarial masks [11, 15]. Therefore, several works[11, 10] enforce priors such as smoothness of the masks. On the other hand, if we use the prior encoded in the network (which is learned from the distribution of the data), we implicitly consider the group of pixels that are correlated with each other. Thus by computing the contribution of individual neurons, we are considering a complex group of pixels and their distributional relationships.

C. Implementation details

The sparsity level of ResNet-50 is 70% and VGG-16 is 90% in the experiments, unless stated otherwise.

C.1. Network parameter randomization sanity check [1]

All attribution methods are run on ResNet50 (PyTorch pretrained) and on 1k ImageNet images. The acquired attribution maps from all methods are normalized to [-1 1] as stated by [1]. The layers are randomized from a normal distribution with mean=0 and variance=0.01 in a cascading manner from the last to the first. After the randomization of each layer, the similarity metrics (SSIM and Spearman Rank Correlation) are calculated between the map from the new randomized model and the original pretrained network. Methods that are not sensitive to network parameters (like GBP) would hence lead to high levels of similarity between maps from normalized networks and the original map.

C.2. Input degradation - LeRF [49]

We report results on CIFAR using a custom ResNet8 (three residual blocks), Birdsnap using ResNet-50, and ImageNet (validation set) using ResNet-50. We show the absolute fractional change of the output as we remove the least important pixels. Lower curves mean higher specificity of the methods. Note that, for NeuronMCT and NeuronIntGrad, the pixel perturbation process is performed on the original model not on the critical paths selected by these methods. The critical paths are only used to obtain the attribution maps and not after.

C.3. Remove and Retrain (ROAR) [21]

We perform the experiments with top 30; 50; 70; 90 of pixels perturbed. The model is retrained for each attribution method (8 methods) on each percentile (5 percentiles) 3 times. Due to the large number of retraining sessions required, we cannot report this benchmark on other datasets. We evaluate this benchmark on CIFAR-10 (60k images, 32x32) with a ResNet-8 (three residual blocks).

D. Supplementary results

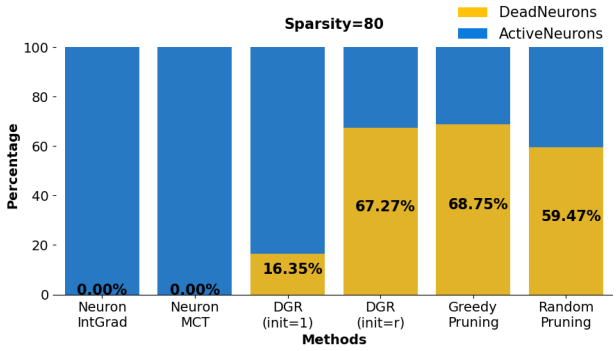

Figure 8. **Dead Neuron Selection of Pruning Objective (Sparsity %80).** The percentage of originally dead neurons in the selected paths of different methods reported for sparsity of 80%. All paths selected by pruning objective contain originally dead (now active) neurons

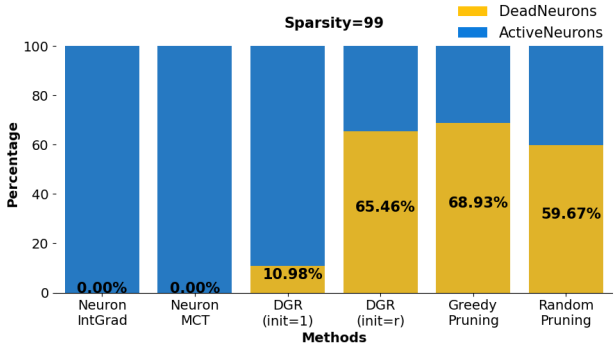

Figure 9. **Dead Neuron Selection of Pruning Objective (Sparsity %99).** The percentage of originally dead neurons in the selected paths of different methods reported for sparsity of 99%. All paths selected by pruning objective contain originally dead (now active) neurons

Table 1. **ROAR**: AUCs reported for each attribution method. The lower the AUC the better.

	Gradient	GBP	GradCAM	InputMCT	InputIntGrad	NeuronMCT	NeuronIntGrad	NeuronIntGrad*
Cifar-10	0.728	0.702	0.584	0.723	0.741	0.580	0.574	**0.524**
Birdsnap	0.269	0.243	0.096	0.242	0.242	0.117	0.099	**0.090**

Table 2. **Input degredation (LeRF)**: AUCs reported for each attribution method. The lower the AUC the better.

	Gradient	GBP	GradCAM	InputMCT	InputIntGrad	NeuronMCT	NeuronIntGrad	NeuronIntGrad*
Cifar-10	0.037	0.028	0.010	0.019	0.019	0.009	0.009	**0.007**
Birdsnap	0.090	0.085	0.012	0.090	0.090	0.010	0.010	**0.008**
ImageNet	0.046	0.044	0.009	0.043	0.041	0.010	0.010	**0.009**

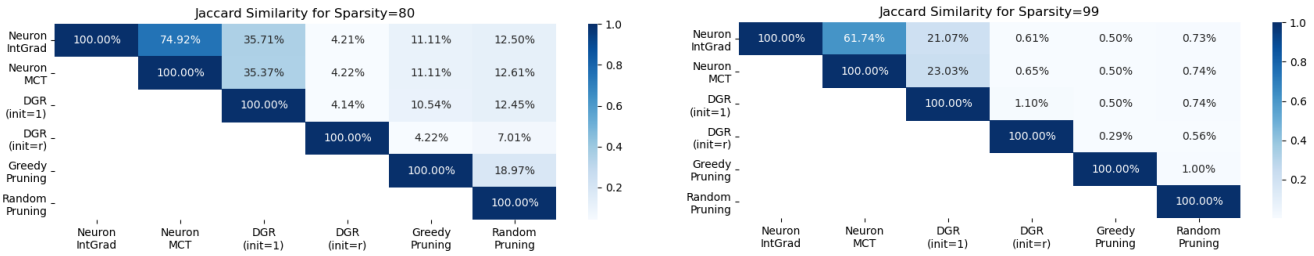

Figure 10. **Path Analysis - Entire Network (Sparsity 80 & 99).** Overlap between paths from different methods in entire network. Among the pruning-based methods, only the path selected by DGR(init=1) overlaps with contribution-based methods.

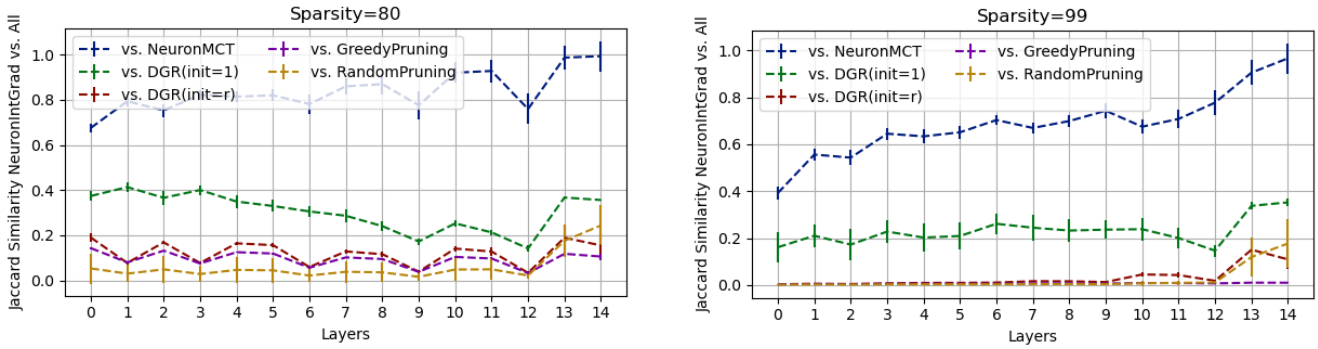

Figure 11. **Path Analysis - Layerwise (Sparsity 80 & 99).** Overlap between paths from different methods in different layers of VGG-16. Among the pruning-based methods, only the path selected by DGR(init=1) overlaps with NeuronIntGrad.

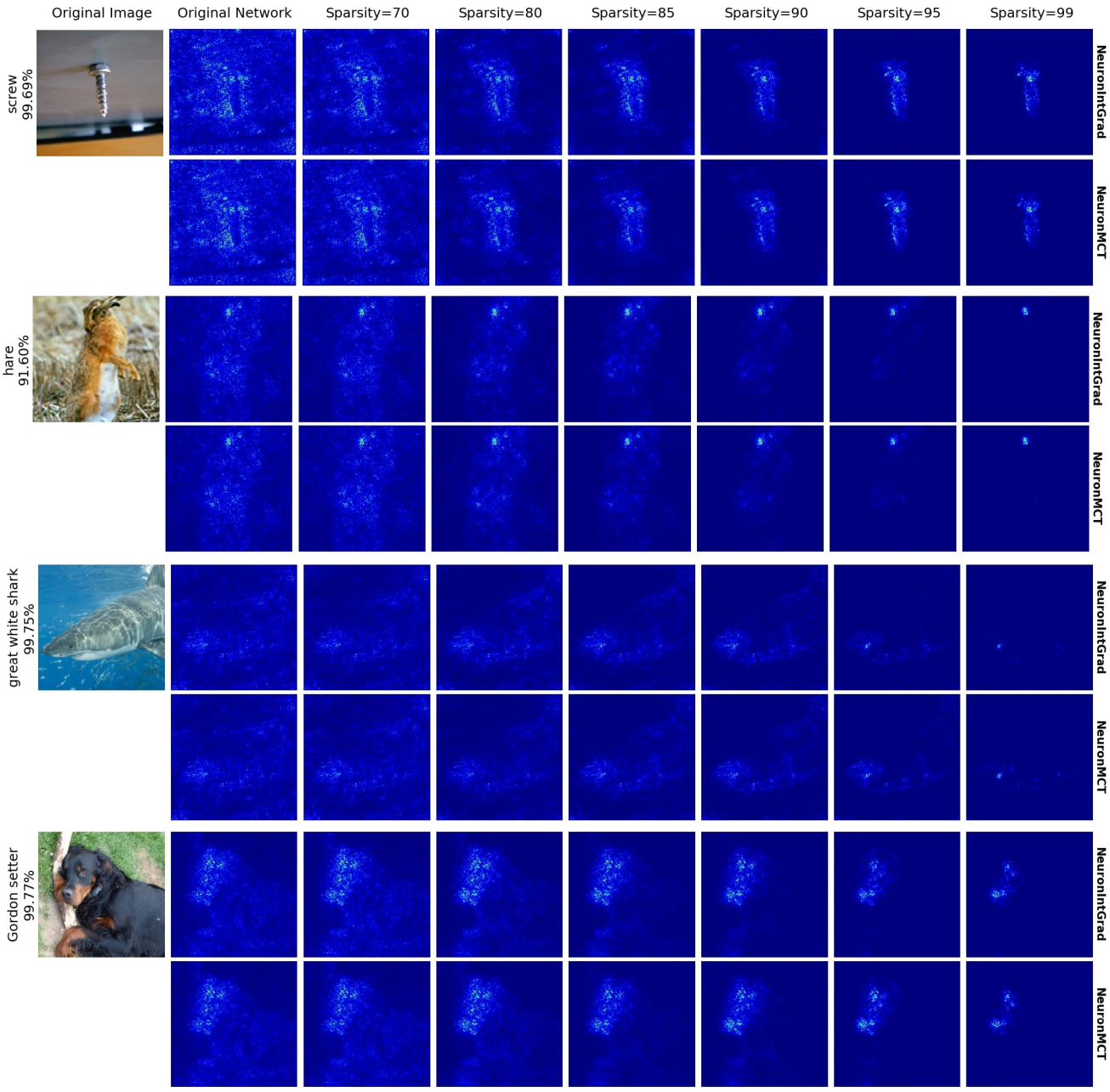

Figure 12. **Feature Attribution.** The gradients of the locally linear critical paths at different sparsity levels for NeuronIntGrad (top) and NeuronMCT (bottom).

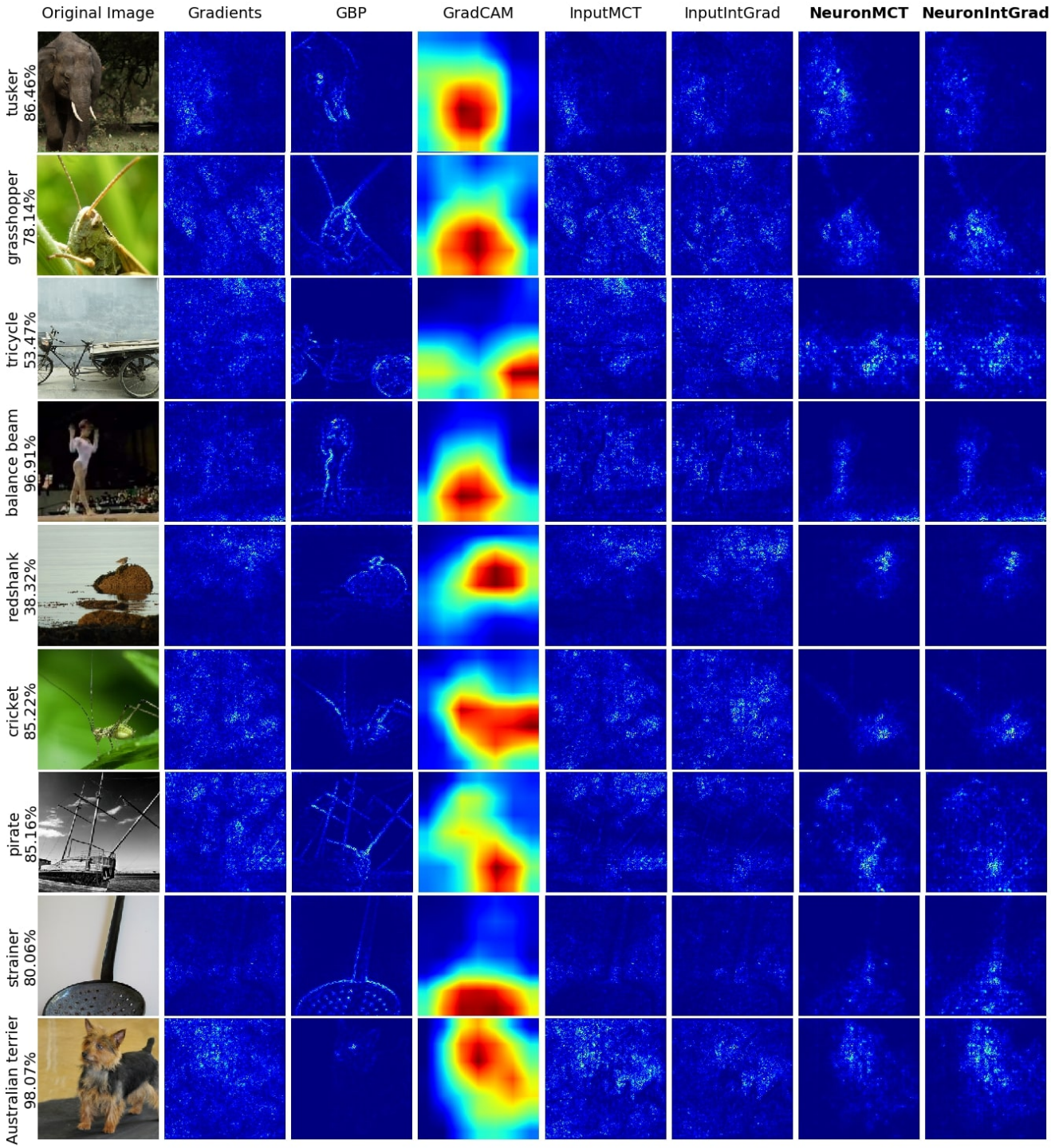

Figure 13. **Comparison with Feature Attribution Methods.** Comparison between attribution maps dervied our proposed methods (right) vs. gradient-based attribution methods on **ResNet-50**. Note the improvement of integrated gradients on the neurons (NeuronIntGrad) over integrated gradients on input (InputIntGrad).

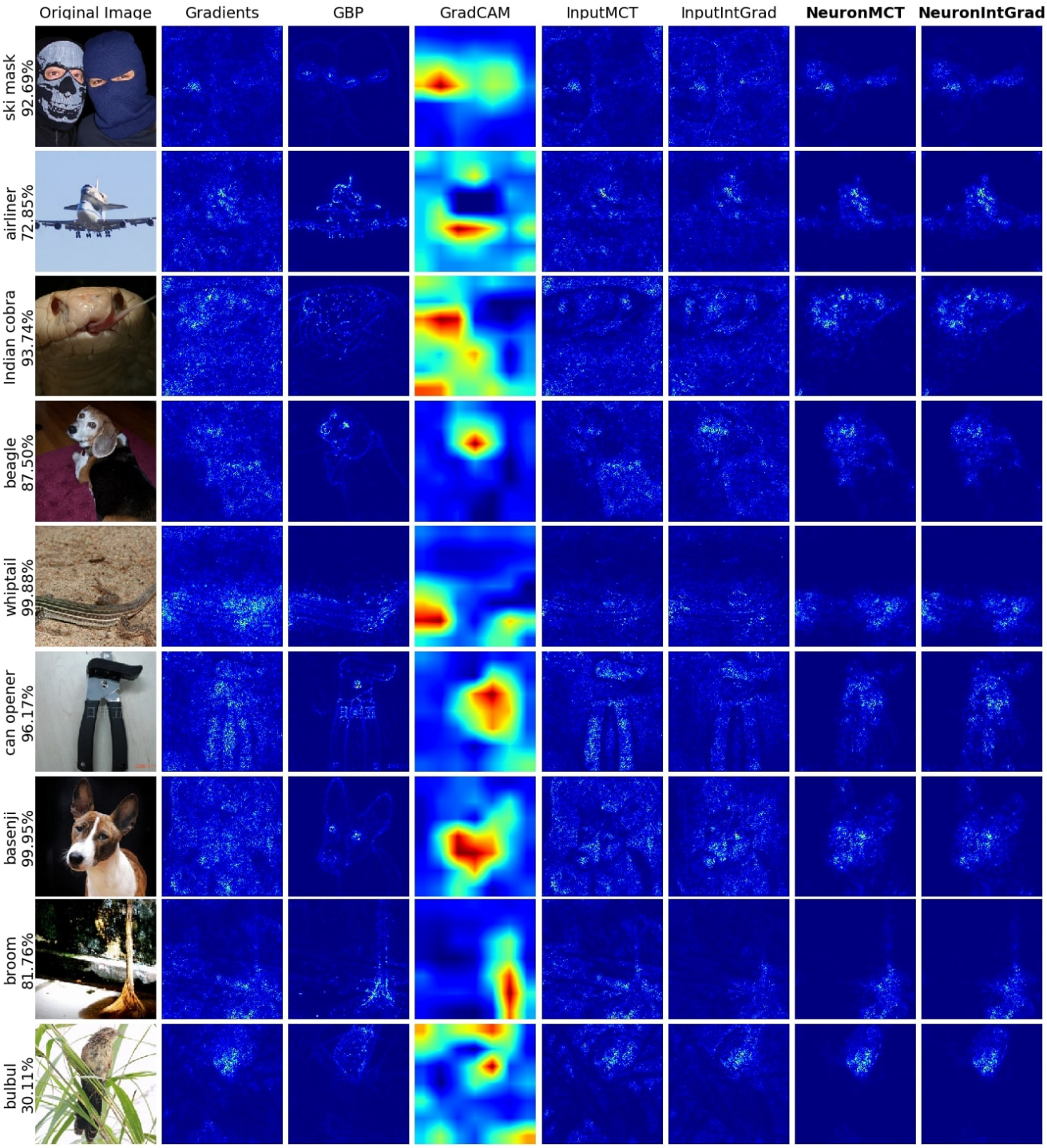

Figure 14. **Comparison with Feature Attribution Methods.** Comparison between attribution maps dervied our proposed methods (right) vs. gradient-based attribution methods on **VGG-16**. Note the improvement of integrated gradients on the neurons (NeuronIntGrad) over integrated gradients on input (InputIntGrad).