# Peter Parker or Spiderman? Disambiguating Multiple Class Labels

**Nuthan Mummani**[1], **Simran Ketha**[1,2], **Venkatakrishnan Ramaswamy**[1,2] *
[1]Department of Computer Science & Information Systems,
Birla Institute of Technology & Science Pilani, Hyderabad 500078, India.
[2]Anuradha & Prashanth Palakurthi Centre for Artificial Intelligence Research,
Birla Institute of Technology & Science Pilani, Hyderabad 500078, India.
{h20221030057, p20200021, venkat}@hyderabad.bits-pilani.ac.in

## Abstract

In the supervised classification setting, during inference, deep networks typically make multiple predictions. For a pair of such predictions (that are in the top-$k$ predictions), two distinct possibilities might occur. On the one hand, each of the two predictions might be primarily driven by two distinct sets of entities in the input. On the other hand, it is possible that there is a single entity or set of entities that is driving the prediction for both the classes in question. This latter case, in effect, corresponds to the network making two separate guesses about the identity of a single entity type. Clearly, both the guesses cannot be true, i.e. both the labels cannot be present in the input. Current techniques in interpretability research do not readily disambiguate these two cases, since they typically consider input attributions for one class label at a time. Here, we present a framework and method to do so, leveraging modern segmentation and input attribution techniques. Notably, our framework also provides a simple counterfactual "proof" of each case, which can be verified for the input on the model (i.e. without running the method again). We demonstrate that the method performs well for a number of samples from the ImageNet validation set and on multiple models.

## 1 Introduction

Supervised deep learning models performing classification are being widely deployed in many settings. An important and active direction of research is on interpretability of the predictions of these models. In the multiclass classification setting, typically, each training datapoint comes with one or few labels Deng et al. (2009), Lin et al. (2014); however models usually output softmax prediction "probabilities" for every class label present in the dataset. Generally, either the top one or top $k$ softmax values are considered as predictions for classification. Since contemporary datasets have large numbers of labels, many of the labels in the top $k$ predictions are likely those that aren't present in the input in question. Indeed, for a given pair of such predicted labels, these prediction probabilities have two distinct interpretations. The first interpretation is that the probabilities represent the possibility of the presence of distinct entities that correspond to each of the class labels. The second interpretation is that the pair of probabilities represent two distinct predictions about a single type of entity present in the image. Both these interpretations could simultaneously be true for different pairs of predicted class labels for a single input that is run through a model. The second interpretation being true for a given pair of labels might detract from our confidence that both the labels are indeed correct predictions; this would indicate the need to verify these predictions via other means – e.g. using a different more capable model or a human. Most contemporary models do

*Code available in https://github.com/mummani-nuthan/Disambiguating-Multiple-Class-Labels

38th Conference on Neural Information Processing Systems (NeurIPS 2024). ATTRIB Workshop.

not offer a direct way to disambiguate these two interpretations for a given pair of top $k$ prediction probabilities, for a single input datapoint. To our knowledge, no existing techniques in the literature have been explicitly designed to address this issue.

In this paper, we build a framework and method to address this problem. On one hand, we build a counterfactual proof framework that entails setting up definitions that disambiguate the aforementioned types of class label pairs. Specifically, we stipulate that an assertion about whether a given pair of labels correspond to distinct entity-types or to a single type of entity must be accompanied by a counterfactual proof, which is a certificate that can be used to verify this claim using the model. Notably, verifying the claim does not require re-running the method that produced it, or indeed even knowledge of the method. Conceptually, this is a departure from typical attribution methods, wherein there isn't an explicit way to objectively verify a claimed attribution that is divorced from the technique that produced it. Secondly, for image recognition, using modern segmentation and attribution techniques, we propose methods that produces such a counterfactual proof. To this end, we first segment the given input image and using existing attribution techniques assign segment-wise attribution scores for each label. These scores are used to determine if the two given label predictions point to the same set of entities or different set of entities. We then build counterfactual proofs that satisfy the aforementioned definitions. Using a number of images from the ImageNet validation set, we demonstrate that the method performs favorably.

## 2 Related work

Attribution techniques have been studied in multiple directions. Perturbations are the simplest among them. Zeiler (2014) implements them by masking the part of the picture with gray square and observing the output. They also implemented a process where outputs at each layer are projected back to the layer's input dimension with minimum loss in the data but also capturing what caused the final activation and termed it as deconvolution. Springenberg et al. (2014) proposed Guided backpropagation, as a modification to deconvolution.

Gradient-based methods, such as gradient descent and backpropagation, form the foundation for many feature attribution techniques. These methods compute the gradient of the model's output with respect to the input features. The magnitude of the gradient indicates the sensitivity of the model's output towards changes in the input features. Gradient Simonyan et al. (2014) itself along with integrated gradients Sundararajan et al. (2017), deepLift Shrikumar et al. (2017), GradCAM Selvaraju et al. (2017), layer wise relevance propagation Bach et al. (2015) are few notable attribution techniques.

By closely inspecting the visualizations of these gradients, Sundararajan et al. (2017) proved that gradients do not work properly. XRAI Kapishnikov et al. (2019) showcases the ability to attribute to particular segments of image with the help of Integrated gradients and segmentation algorithms like Felzenswalb's graph based algorithm Felzenszwalb and Huttenlocher (2004).

LIME Ribeiro et al. (2016) and SHAP Lundberg and Lee (2017) have different approaches. LIME tried to generate interpretable explanations local to the input in question which are understandable to humans. SHAP on the other hand tries to explains how important the feature is in the given prediction based on a concept of Shapley values from Game Theory. While all these methods calculate/explain the importance of each feature for the given generated output, we would like to expand on the role of these features when we consider multiple outputs.

Relatedly, the issue of popular contemporary datasets such as ImageNet having one label per image has also received attention. For example, Beyer et al. (2020) point out that even though images in ImageNet training set often contain multiple objects, only one of them is recognized in the label.

## 3 Definitions and Preliminaries

We now present some definitions and preliminaries that will be used in the remainder of the paper. While we apply the framework to the image classification setting here, these definitions could, in principle, also be applied to other types of supervised learning models.

For our purposes here, we define a deep network model as a function that maps input points in $n$-dimensional space to a vector of softmax "probabilities" corresponding to $m$ class labels.

**Definition 1.** *A deep network model is a function $f : \mathbb{R}^n \rightarrow [0,1]^m$ , which maps an input in $n$-dimensional space to a vector of softmax values corresponding to $m$ class labels.*

Next, we define the notion of a *redaction* of an input, which intuitively corresponds to replacing values in a subset $S$ of the dimensions in the input to a single fixed value. The idea is that doing so will remove the information present in those dimensions and such a redacted input would serve as a counterfactual input. While we do not claim that this manner of constructing counterfactual inputs is a canonical one, we find that it does work well in practice for image recognition networks, as demonstrated in Section 4.1.

**Definition 2** ($S$-redaction)**.** *Given an input $I \in \mathbb{R}^n$ and a set $S \subseteq \{1, \ldots, n\}$ of indices, an $S$-redaction of $I$ to $v$, is defined as the input $I_S$ obtained by replacing the values corresponding to the indices in $S$ to the value $v$.*

Unless otherwise specified, when we mention an $S$-redaction here, we mean an $S$-redaction to zero. If the input is an image, an $S$-redaction of it would correspond to the image generated by "blackening" out the subset of the pixels corresponding to S. Also, if each pixel has multiple channels (e.g. R,G,B), an $S$-redaction will zero out values in all channels, for every pixel present in S.

Informally, an input attribution for a specific class label is typically understood to correspond to the input dimensions that are "responsible" for the prediction of that class label by the deep network model. Here, we will define an attribution to simply be a subset of input dimensions (i.e. without assigning relative weights to every dimension in the subset). We now define a natural counterfactual notion of input attributions that precisely quantifies the same in a verifiable manner.

**Definition 3** ($\delta$-attribution)**.** *For a deep network $f : \mathbb{R}^n \rightarrow [0,1]^m$, input $I \in \mathbb{R}^n$, label $l$ with prediction $p$, $\delta \in [0,1]$, and $S \subseteq \{1, \ldots, n\}$, if the $S$-redaction of $I$ causes the prediction of $l$ to be at most $\delta p$, then $S$ is said to be a $\delta$-attribution for label $l$ corresponding to input $I$, with respect to $f$.*

Here, the intent is to have $\delta$ be a small value (e.g. $\delta = 0.2$)

This definition of a $\delta$-attribution naturally leads to a verification method. The idea is that one can accompany a claimed $\delta$-attribution with a counterfactual proof or certificate, which in this case would simply be the $\delta$-attribution $S$. This allows a verifier to easily verify a claimed $\delta$-attribution without needing to re-run the method that determined it or indeed even having knowledge of the method.

We now define the two major types of label predictions, given a pair of label predictions for an input by a deep network, namely $\delta$-disjoint label predictions and $\delta$-overlapping label predictions. A $\delta$-disjoint label prediction corresponds to the case in which two distinct types of entities are present in the input that correspond respectively to the two class labels in question, with softmax prediction values being $p_1$ and $p_2$ respectively. The definition posits that if this is the case, then there must exist two redactions – an $S_1$-redaction and an $S_2$-redaction – where $S_1$ and $S_2$ are disjoint sets. Furthermore, the $S_1$-redaction must cause the softmax prediction for the first label to dip to $\delta p_1$ or below, while keeping the softmax prediction for the second label to be at least $(1-\delta)p_2$. The $S_2$-redaction behaves likewise for the second label.

**Definition 4** ($\delta$-disjoint label predictions)**.** *For $\delta \in [0, 0.5]$, suppose we have a deep network $f : \mathbb{R}^n \rightarrow [0,1]^m$ which, on input $I$, has predictions $p_1$ and $p_2$ for class labels $l_1$ and $l_2$ respectively. The class labels $l_1$ and $l_2$ are said to be* $\delta$-disjoint*, if there exist disjoint sets $S_1$ and $S_2$ such that*

1. *The $S_1$-redaction of $I$ causes a $\delta$-attribution to exist for class label $l_1$, while causing the prediction for class $l_2$ to be at least $(1-\delta)p_2$.*

2. *The $S_2$-redaction of $I$ causes a $\delta$-attribution to exist for class label $l_2$, while causing the prediction for class $l_1$ to be at least $(1-\delta)p_1$.*

Here, again, for two labels $l_1$ and $l_2$, claimed $\delta$-disjoint label predictions will be accompanied by a certificate, which would simply be the $\delta$-attributions $S_1$ and $S_2$ that satisfy the above definition.

Next, given a pair of label predictions for an input by a deep network, we define $\delta$-overlapping label predictions. This is the case when the two labels in question correspond to a single entity type present in the input. The idea is to establish this case by demonstrating a $S$-redaction that is a $\delta$-attribution for both the class labels without the labels being $\delta$-disjoint.

**Definition 5** ($\delta$-overlapping label predictions)**.** *For a deep network $f : \mathbb{R}^n \to [0, 1]^m$ with an input $I$, two class labels $l_1$ and $l_2$ are said to be* $\delta$-overlapping*, if $l_1$ and $l_2$ are not $\delta$-disjoint and if there exists a set $S$ such that an $S$-redaction causes a $\delta$-attribution to exist for class labels $l_1$ as well as $l_2$.*

Here, again, the certificate would be the $\delta$-attribution $S$; however it is unclear if a tractable verification algorithm exists, since one might need to check all partitions of S – of which there are exponentially many – to check if they correspond to $\delta$-disjoint label predictions. In Section A.2.2, we describe a heuristic verification algorithm that is tractable and demonstrate that it works well.

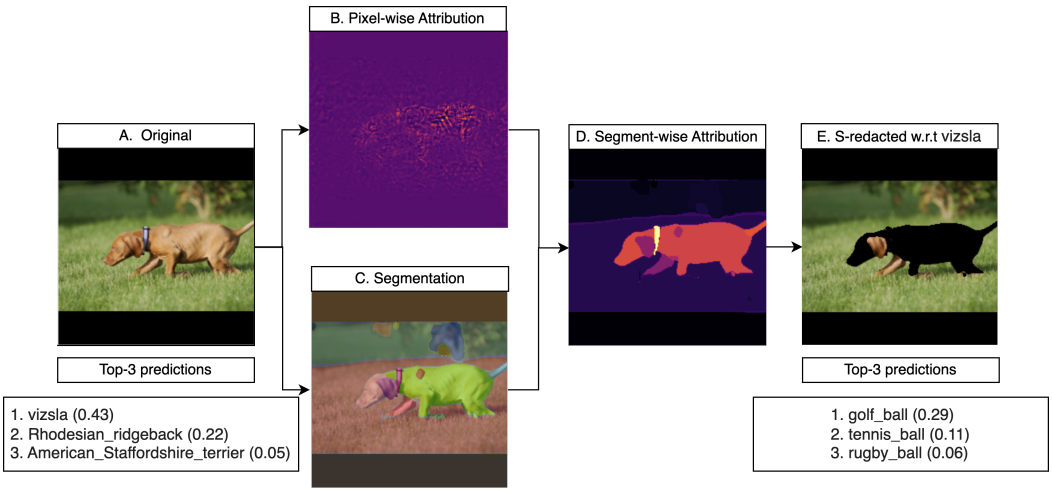



Figure 1: An illustration of rank-based redaction. **A.** An image from the ImageNet validation set from the class `vizsla` is padded with zeros to match the input dimension of VGG16 model to obtain the image shown. Corresponding top-3 predictions are mentioned. **B.** The image in A. is attributed to the label `vizsla` using integrated gradients to obtain pixelwise attribution values. **C.** The image in A. is segmented using the SAM model. **D.** The pixel-wise attribution values from B. are averaged over the segments and these segments are ranked accordingly to get segment-wise attributions for the label `vizsla`. **E.** Top 25% of the ranked segments are then redacted to get an $S$-redaction. Corresponding top-3 predictions for this $S$-redacted image are mentioned. The prediction for `vizsla` on this $S$-redacted image dropped to 0.010. This process on the same image with ResNet-50 and Inception-v3 are shown in Figure 4.

## 4 Methodology

Leveraging modern input attribution and segmentation techniques, we build algorithms to determine if a given pair of labels is $\delta$-disjoint or $\delta$-overlapping. These algorithms also return the corresponding certificates. We deploy and test these algorithms on image classification models VGG-16 Simonyan and Zisserman (2014), Inception-v3 Szegedy et al. (2016), and ResNet-50 He et al. (2016) which are pretrained on the ImageNet dataset Deng et al. (2009). We use images from the ImageNet validation dataset in our test, unless otherwise mentioned.

For a label available in the top $k$ predictions of an input image, we calculate pixel-wise attribution using integrated gradients Sundararajan et al. (2017) and parallelly, we segment the image using Segment Anything Model (SAM) Kirillov et al. (2023). We then performed segment-wise accumulation of attribution values to rank the segments from highest attribution to lowest attribution, along the lines of XRAI Kapishnikov et al. (2019). These segment-wise rankings are used in the later part of the paper and can be visualized using heatmaps (Figure 1).

### 4.1 Effectiveness of Redactions

Here, we demonstrate that redactions to zero are an effective counterfactual proof, in practice. Redacted images are constructed by picking up segments one-by-one based on segment-wise attribution rankings and replacing segmented areas with black pixels in the original preprocessed image.

The process of creating the redacted images for the top predicted label using VGG-16 model is shown in Figure 1, for a sample image.

Every network has its own pre-processing stage where e.g. in VGG16, it zero-centers the data with respect to the dataset. Although, the redacted images shown in this paper are generated by performing redactions on images before pre-processing step for the purpose of visualization, in practice we redact the segment after the pre-processing step.

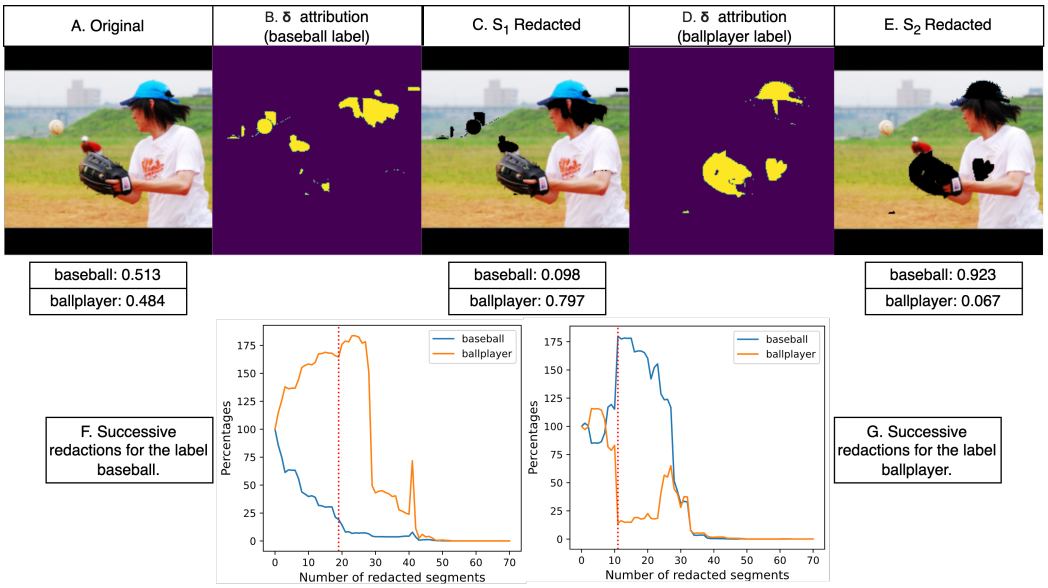



Figure 2: Example illustrating $\delta$-disjoint attributions. **A.** An image from the ImageNet validation set & its corresponding top-2 labels with their predictions on VGG-16 model. **B.** For $\delta = 0.2$, $\delta$-attribution for the label `baseball` (indicated in *yellow*) obtained using the algorithm discussed in Section 5. **C.** The corresponding redacted image for the label `baseball` with the resultant prediction values. **D.** For $\delta = 0.2$, $\delta$-attribution for the label `ballplayer` (indicated in *yellow*) obtained using the algorithm discussed in Section 5. **E.** The corresponding redacted image for the label `ballplayer` with the resultant prediction values. For both labels, percentage of softmax prediction values while redacting segments with respect to the original image are plotted. **F.** For the algorithm discussed in Section 5, we plotted the percentage change in prediction for the two labels, when the segments ranked for the label `baseball` were successively redacted in order of their rank. Here, the $\delta = 0.2$ attribution is obtained at the 19th redaction (*red-dotted line*) where prediction of `baseball` is atmost $\delta p_1$ ($0.098 < 0.2 * 0.513$) and prediction of `ballplayer` is atleast (1-$\delta$)$p_2$ ($0.797 > 0.8 * 0.484$). **G.** Corresponding plot for `ballplayer`. Additional examples are provided in Appendix A.2.3.

## 5 Distinct labels pointing to distinct entities

To determine if two labels from the top-$k$ predictions are driven by distinct entities in an image, we need $S_1$ and $S_2$ redactions, if available, that satisfy Definition 4. One method to obtain such redactions is discussed below & two other methods are presented in the Appendix A.2.

Given a list of segment attribution values for one label, for each segment, we determine the proportion of the segment's attribution value with respect to the highest segment attribution value, which we call its *normalized segment attribution*. We do so for the other label as well. Now we segregate the segments into two disjoint sets corresponding to the two labels. For any segment, if the normalized segment attribution for label $l_1$ is higher than that for label $l_2$, then that segment is categorized within the set of label $l_1$ and vice-versa. In case of a tie, we use the data of surrounding segments for categorization. Now that we have two disjoint sets of segments, one for each label, we pick each segment from label $l_1$'s set based on their rank and redact by sequentially accumulating them to form an $S_1$-redaction until the prediction of corresponding class label $l_1$ goes down to at most $\delta p_1$ while the $l_2$ prediction stays above (1-$\delta$)$p_2$ where $p_1$ and $p_2$ are the softmax probability of original image

of labels $l_1$ and $l_2$ respectively. This step is repeated on $l_2$'s set to obtain an $S_2$ redaction. Redacting based on their rank allows us to get the $S_1$ and $S_2$ redactions that have a small number of segments. We find that this method is effective in finding redactions that satisfy Definition 4. In Figure 2, we illustrate this method, for a sample image with $\delta = 0.2$.

We explored two more ways to generate redactions that satisfy Definition 4. One includes finding $S$-redactions and then making them disjoint and the other generates $S$-redactions without using the pixel-wise or segment-wise attributions. These are discussed in AppendixA.2.

## 6 Distinct labels pointing to single entity

To determine if two labels from the top-$k$ predicted labels are "pointing" to a single entity in an image, we do the following. We pick each segment based on their absolute[2] rank for label $l_1$ and redact by sequentially accumulating them to form an $S_1$ redaction until the prediction of both the labels go down simultaneously to at most $\delta p_1$ and $\delta p_2$ respectively. This process is likewise repeated with segments ranked based on $l_2$ to obtain a $S_2$ redaction. If $l_1$ and $l_2$ are indeed pointing to single entity, then $S_1$ and $S_2$ redactions present themselves with a significant intersection and $S_1 \cap S_2$, on satisfying Definition 5, is used as a $\delta$-attribution.The $\delta$-attribution for $S_1 \cap S_2$, as illustrated in Figure 3, acts as a certificate which can be used to verify that the two labels indeed "point" to a single entity in the image.

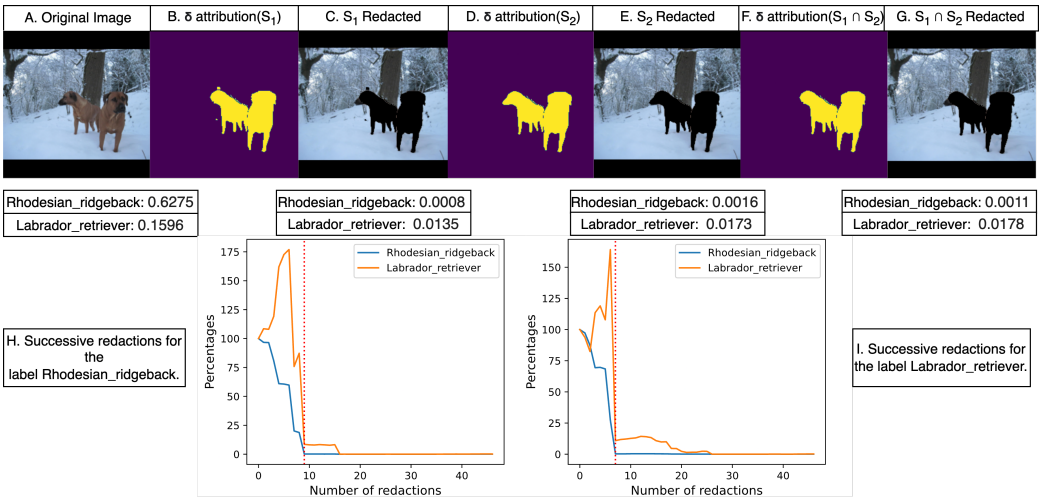



Figure 3: Example illustrating $\delta$-overlapping attributions. **A.** An image from the ImageNet validation set (whose correct label from the validation set is `Rhodesian_ridgeback`) and its corresponding top-2 labels with their predictions values on VGG-16. **B.** For $\delta = 0.2$, $\delta$-attribution for the label `Rhodesian_ridgeback` (indicated in *yellow*) obtained using the algorithm discussed in Section 6. **C.** The corresponding redacted image for the label `Rhodesian_ridgeback` with the resultant prediction values. **D.** For $\delta = 0.2$, $\delta$-attribution for the label `Labrador_retriever` (indicated in *yellow*) obtained using the algorithm discussed in Section 6. **E.** The corresponding redacted image for the label `ballplayer` with the resultant prediction values. **F.** For $\delta = 0.2$ , $S_1 \cap S_2$ is verified to satisfy Definition 5. **G.** The corresponding $S_1 \cap S_2$-redacted image with the resultant prediction values. For both labels, percentage of softmax prediction values while redacting segments with respect to the original image are plotted. **H.** For the algorithm discussed in Section 6, we plotted the percentage change in prediction for the two labels, when the segments ranked for the label `Rhodesian_ridgeback` were successively redacted in order of their rank. Here $\delta = 0.2$ attribution is obtained at the 9th redaction (*red dotted line*) where prediction of `Rhodesian_ridgeback` is atmost $\delta p_1$ ($0.0008 < 0.2 * 0.6275$) and prediction of `Labrador_retriever` is also atmost $\delta p_2$ ($0.0135 < 0.2 * 0.1596$). **I.** Corresponding plot for the label `Labrador_retriever`. Additional examples are provided in Appendix A.2.3.

[2]i.e. non-normalized, as described in Section 4

This leaves open the possibility that one chooses a pair of labels that are in fact $\delta$-disjoint and by merely generating $S = S_1 \cup S_2$ provides a purported certificate for $\delta$-overlapping label predictions. In such cases, each subset of the $S$ provided needs to be redacted to check if there exists any subset of $S$ that only causes one label to rise rather than both. But, such a check would take exponential time with respect to the number of segments, which will be intractable in most cases. Hence, to avoid this, we propose a tractable heuristic verification algorithm in Appendix A.2.2 that does not require attribution values. It separates $S_1$ and $S_2$ from $S = S_1 \cup S_2$ when initialised with $S$ and invalidates such a purported certificate. We demonstrate that this heuristic works well in Figure 5. To avoid generating such an incorrect $S$-redaction ourself, we first run the $\delta$-disjoint label predictions algorithm & then run the $\delta$-overlapping predictions algorithm, if the former algorithm doesn't succeed.

## 7 Discussion

In this paper, we consider the problem of disambiguating the input attributions of a given pair of class labels. Specifically, we ask if the two label predictions arise from the same percept or from different percepts present in the input. We build a method and framework to do so, by leveraging modern attribution and segmentation techniques and demonstrate favorable performance on a number of contemporary image classification models.

This work comes with some limitations. Firstly, we use existing attribution and segmentation algorithms and, as such, depend on their performance; this also has the positive effect that improvements in such techniques will likely improve our method. Another limitation is that, for cases wherein the object corresponding to a label isn't present, our method does not specifically identify that this is so; see Section A.2.3. Finally, we find, empirically, that for labels whose softmax values are very small, the method often does not perform well. Indeed, this may be because for such small prediction values the model does not tangibly use a coherent set of segments for such predictions.

A conceptual departure from typical attribution methods is our stipulation that a claimed answer ought to be accompanied by a certificate that can be objectively verified, i.e. without appeal to the method that created it. Often, different attribution methods offer differing attributions and it is difficult to objectively and automatically ascertain the quality of these attributions without human scoring. We therefore suggest that this type of framework will also have value in such settings.

## Acknowledgments and Disclosure of Funding

Simran Ketha was supported by an APPCAIR Fellowship, from the Anuradha & Prashanth Palakurthi Centre for Artificial Intelligence Research. The authors declare no competing interests.

## References

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

Lucas Beyer, Olivier J. Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. 2020. Are we done with ImageNet? arXiv:2006.07159 [cs.CV] `https://arxiv.org/abs/2006.07159`

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 248–255.

Pedro F Felzenszwalb and Daniel P Huttenlocher. 2004. Efficient graph-based image segmentation. *International journal of computer vision* 59 (2004), 167–181.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

Andrei Kapishnikov, Tolga Bolukbasi, Fernanda Viégas, and Michael Terry. 2019. Xrai: Better attributions through regions. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4948–4957.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4015–4026.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 740–755.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " 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*. 1135–1144.

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

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 3145–3153. `https://proceedings.mlr.press/v70/shrikumar17a.html`

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. arXiv:1312.6034 [cs.CV] `https://arxiv.org/abs/1312.6034`

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806* (2014).

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*. PMLR, 3319–3328.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

MD Zeiler. 2014. Visualizing and Understanding Convolutional Networks. In *European conference on computer vision/arXiv*, Vol. 1311.

# A Appendix

## A.1 Illustration of rank-based redaction for ResNet-50 and Inception-v3

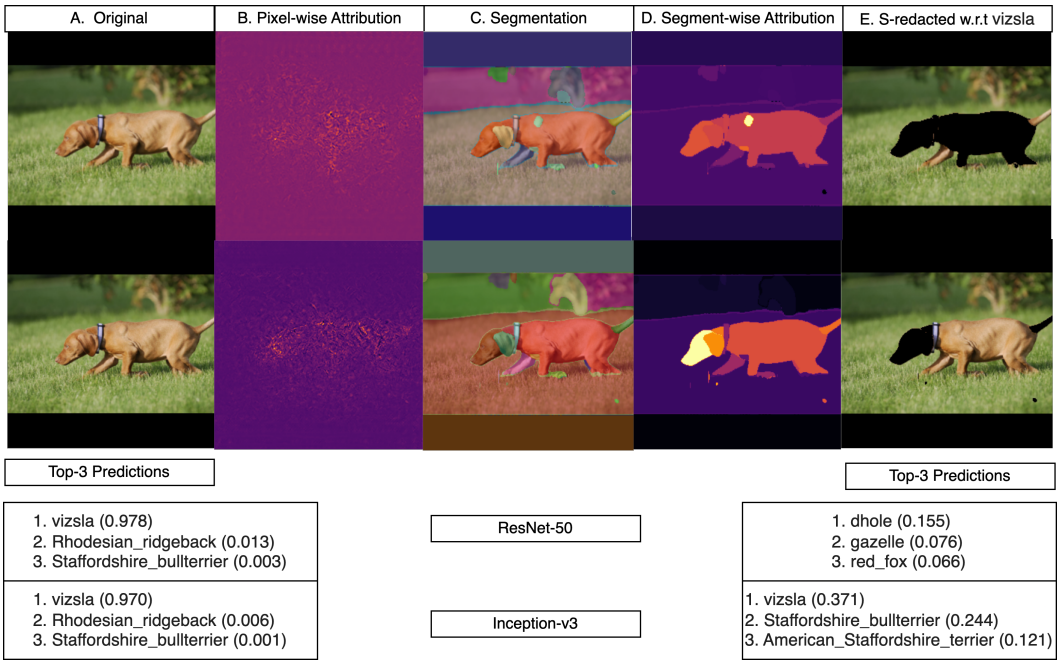



Figure 4: Illustration of rank-based redaction for different models (*Top row : ResNet-50 , Bottom row : Inception-v3*) using the same image and following a similar pipeline as Figure 1.

Figure 4 illustrates rank-based redactions for ResNet-50 and Inception-v3 for a sample image. Observe that an image can have differing $S$-redactions for different models.

## A.2 Two additional algorithms for finding $\delta$-disjoint attributions

### A.2.1 Algorithm 2

As we perform the steps mentioned in Section 4, for labels $l_1$ and $l_2$, we obtain two identical sets of segments that are ranked within their set based on segment-wise attribution method Kapishnikov et al. (2019). Now that we have two sets of segments, we pick each segment from label $l_1$'s set based on their rank and redact by sequentially accumulating them to form an $S_1$-redaction until the prediction of corresponding class label $l_1$ goes down to at most $\delta p_1$ while the $l_2$ prediction stays above (1-$\delta$)$p_2$ where $p_1$ and $p_2$ are the softmax probability of original image of labels $l_1$ and $l_2$ respectively. This step is repeated on $l_2$'s set to obtain $S_2$ redaction. Redacting based on their rank in corresponding sets allows us to get the $S_1$ and $S_2$ redactions with small number of segments. These two redactions might not satisfy Definition 4 as they may not be disjoint. We then discard the intersection segments or reassign them one-by-one to either $S_1$ or $S_2$ redactions based on their importance to the respective labels. This step makes the $S_1$ and $S_2$ redactions disjoint and satisfies Definition 4.

### A.2.2 Algorithm 3 (which also serves as a heuristic verifier for Definition 5)

Given a set of segments $E$ that are obtained after segmentation, we would like to curate two disjoint sets of segments that are $\delta$ attributions for each label and can generate $S$-redactions that satisfy Definition 4. To generate $\delta$ attribution set of segments for label $l_1$, we start with an empty set $A$ and execute the following algorithm.

1. Select the most important segment $s_i$ and pop it out of $E$ and push it into $A$.
2. Generate $D$ with the set of segments that are adjacent to any segment in $A$.

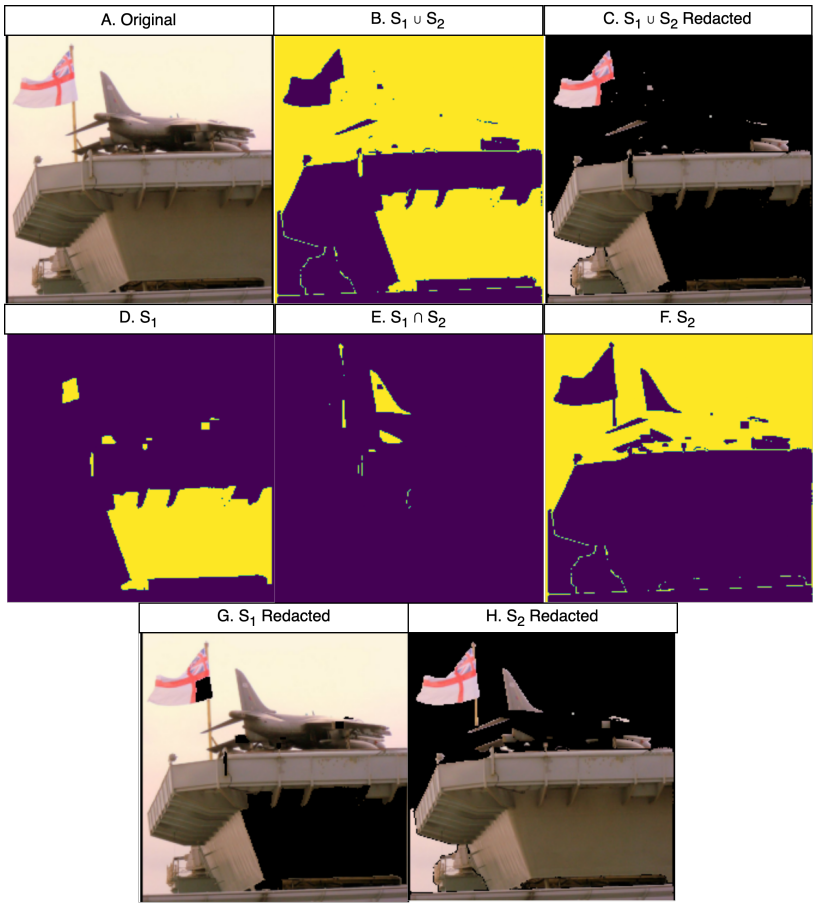
3. Pop out next most important segment from $D$ and push it into $A$.
4. Repeat the process from $Step2$ until we end up with no important segment in $D$.
5. Repeat from $Step1$ and start with a different segment until we are left with no important segments for label $l_1$. This process is repeated to generate $\delta$ attribution set of segments for label $l_2$.

How do we choose most important segment? To calculate the importance score of segment $s_i$ we redact the segment $s_i$ from $A$-redacted image(image with all segments from $A$ redacted) and check the percentage drop in $l_1$ and $l_2$ prediction values from $A$-redaction to $A \cup \{s_i\}$ redaction. The difference between $l_1$ percentage drop and $l_2$ percentage drop is the importance score for the segment $s_i$ during that step.

We end up with $S_1$ and $S_2$ redactions that are not disjoint, and discarding the intersection gives us two disjoint redactions. These redactions are then verified to check if they satisfy Definition 4 and corresponding $\delta$ attributions are used as certificates to validate the image later.

The limitation of this algorithm is that , unlike attribution based algorithms from Sections 5 & A.2.1, it does not provide the redactions with a small number of segments, since it does not pick the segments based on the ranks generated by the attribution algorithms.

This algorithm also acts as a heuristic verifier for Definition 5. Provided with set of segments $S = S_1 \cup S_2$ as $\delta$ overlapping attribution certificate from Section 6, if the image and labels correspond to $\delta$ disjoint from Section 5, this algorithm segregates the segments into $S_1$ and $S_2$ sets that correspond to each labels satisfying Definition 4. These $S_1$ and $S_2$ sets invalidate the $\delta$ attribution certificate. Usage of this algorithm can remove the overhead of attribution. User performing the verification need not have the knowledge of algorithm that generated the certificate. An example is demonstrated in Figure 5.




| | Original | $S_1 \cup S_2$ redacted | $S_1$ Redacted | $S_2$ Redacted |
|---|---|---|---|---|
| aircraft_carrier | 0.444 | 0.015 | 0.025 | 0.779 |
| projectile | 0.250 | 0.008 | 0.495 | 0.0135 |

Figure 5: Example illustrating Algorithm 3 as a verifier for Definition 5. Here we take a sample image which satisfies the definition of $\delta$-disjoint attributions for a pair of classes. We then suppose $S = S_1 \cup S_2$ and challenge the verifier by offering $S$ as a certificate for $\delta$-overlapping attributions. We demonstrate that the verifier indeed flags $S$ as an incorrect certificate for $\delta$-overlapping attributions. **A.** An image with labels `aircraft_carrier` and `projectile` which satisfies Definition 4 on ResNet-50 model. **B&C.** $S = S_1 \cup S_2$ acting as a purported certificate for Definition 5, with $\delta = 0.2$, for both `aircraft_carrier` and `projectile` classes. **D, E & F.** Algorithm A.2.2 breaks the set $S$ to become $S_1, S_1 \cap S_2, S_2$ in *D, E & F* respectively. **G & H.** $S_1$ and $S_2$ redaction images formed using *D & F* respectively. *G & H* together satisfies Definition 4 with $\delta = 0.2$ and hence the verifier rejects the certificate $S$ for Definition 5.

### A.2.3 Additional examples on VGG-16, ResNet-50 and Inception-v3

Illustrations of $\delta$-disjoint and $\delta$-overlapping attributions and their corresponding redactions performed on various images are shown in Figures 6, 7, and 8. Note that the label pairs chosen in these figures are picked from top-5 rather than top-2.

We mention an example in Figure 6 (VGG-16, example 1), wherein the method flags it as a $\delta$-disjoint attribution, even though a human inspection shows that no object corresponding to the class `moving_van` is present. This is a limitation, as previously mentioned, even though the method shows that differing segments cause the $\delta$ attributions for the two classes.

Figure 7 and 8 demonstrate our method on a challenging example, which contains both a spider and a bee. In a pair of classes that correspond to a spider and a bee, it is classified as a $\delta$-disjoint attribution; however with two spider class labels, it is classified as a $\delta$-overlapping attribution. Specifically, in Figure 7, it is observed that for Inception-v3, the image is categorized into $\delta$-disjoint attribution for pair of labels (`garden_spider`, `bee`) and (`bee`, `black_and_gold_garden_spider`), whereas the same image for labels (`garden_spider`, `barn_spider`) is categorized into $\delta$-overlapping attribution as shown in Figure 8.

## A.3 Running time estimates for our methods

All our experiments were run on an Apple Macbook Pro with M1 Pro chip, 16GB RAM, running macOS 12.1, and the running time estimates below correspond to this hardware.

SAM segmentation algorithm: 127 seconds per image.
Pixel-wise attribution and Segment-wise attribution: 64 seconds for a pair of chosen labels.
Algorithm mentioned in Section 5: 15 seconds for a pair of chosen labels.
Algorithm mentioned in Section 6: 16 seconds for a pair of chosen labels.
Algorithm mentioned in Section A.2.1: 16 seconds for a pair of chosen labels.
Algorithm mentioned in Section A.2.2: 900 seconds for a pair of chosen labels.

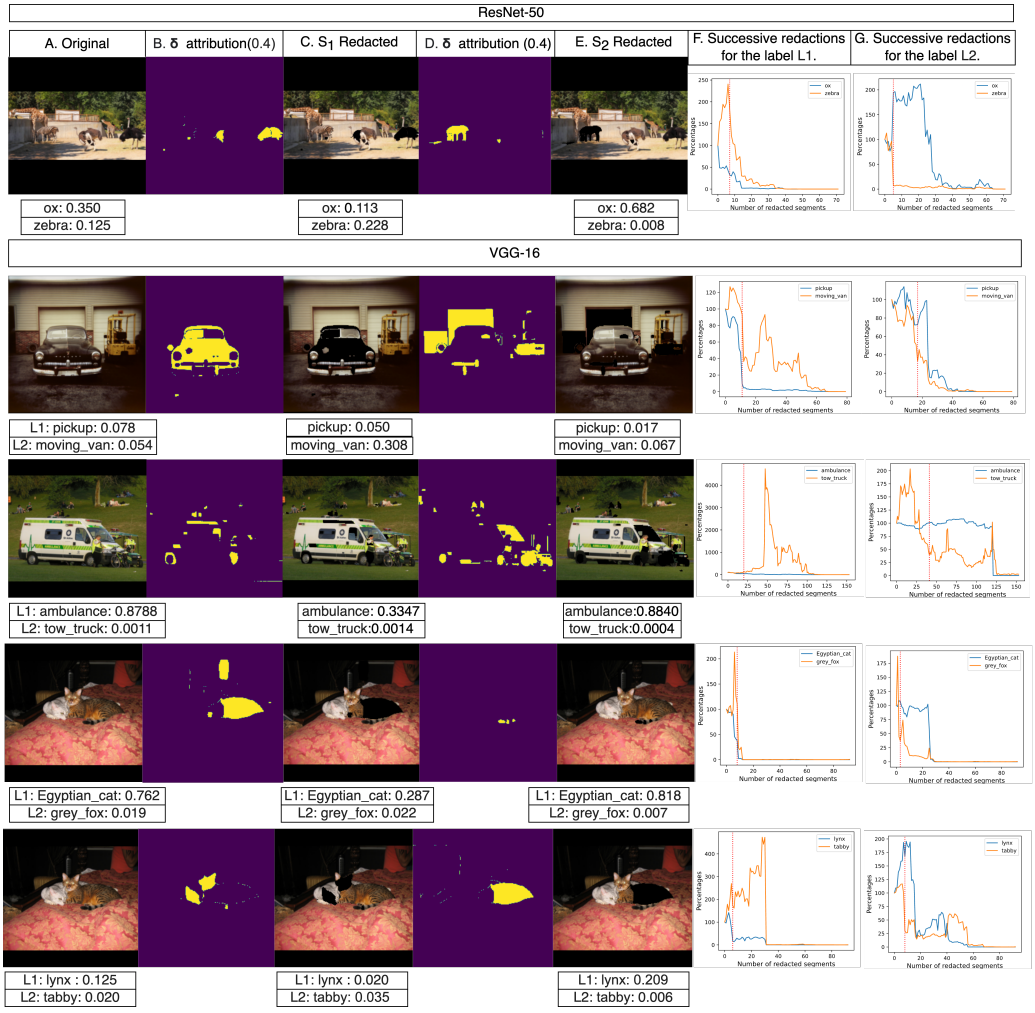



Figure 6: Illustrations of $\delta$-disjoint attributions with $\delta$=0.4 on ResNet-50 and VGG-16 using various images following a similar pipeline as Figure 2. Image used in the first row is not from the ImageNet validation dataset.

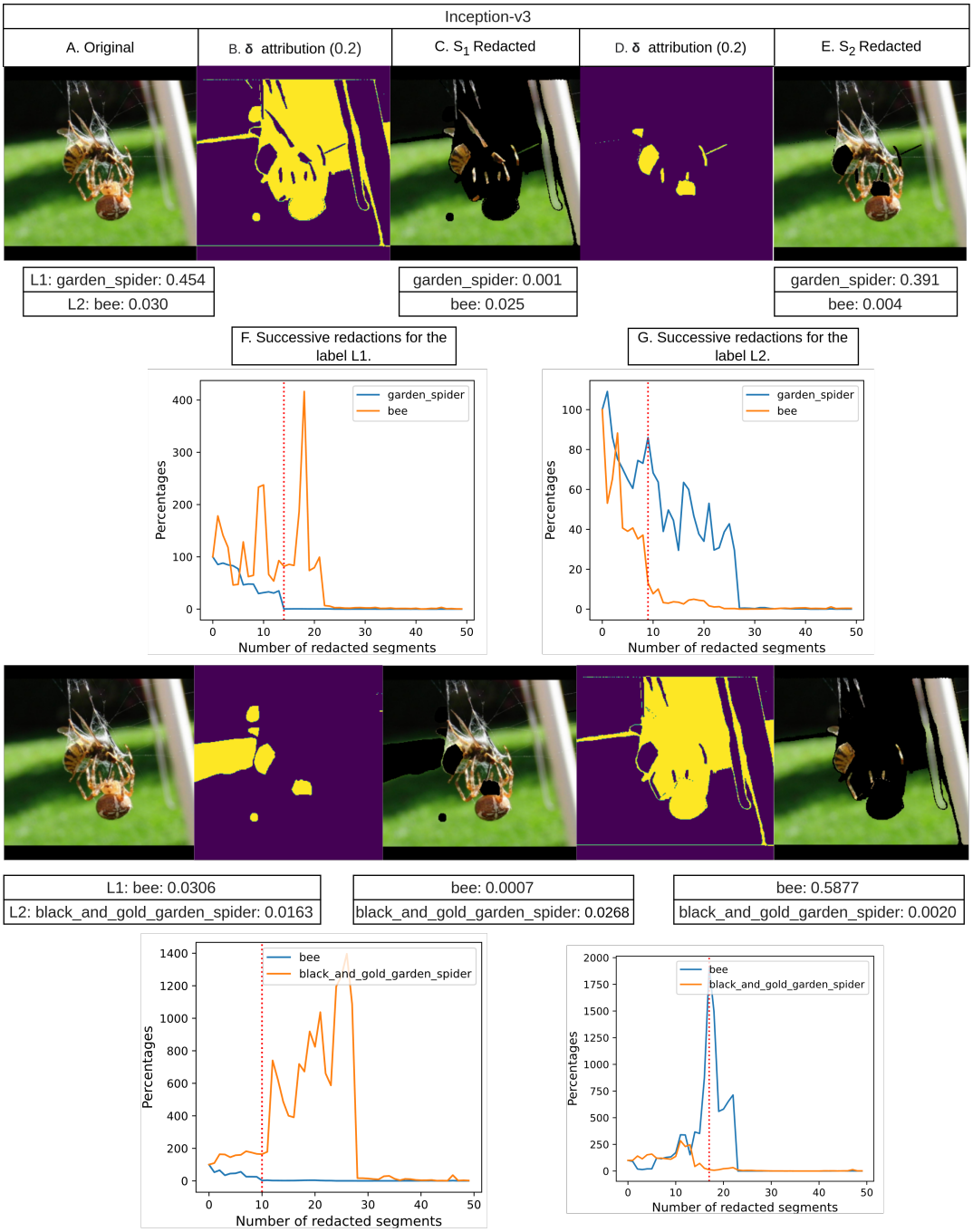



Figure 7: Illustrations of $\delta$-disjoint attributions with $\delta$=0.4 on Inception-v3 for two pairs of labels using a single image following a similar pipeline as Figure 2. Observe that the image indeed has both a bee and a spider. The image is from the `garden_spider` class from the ImageNet validation dataset.

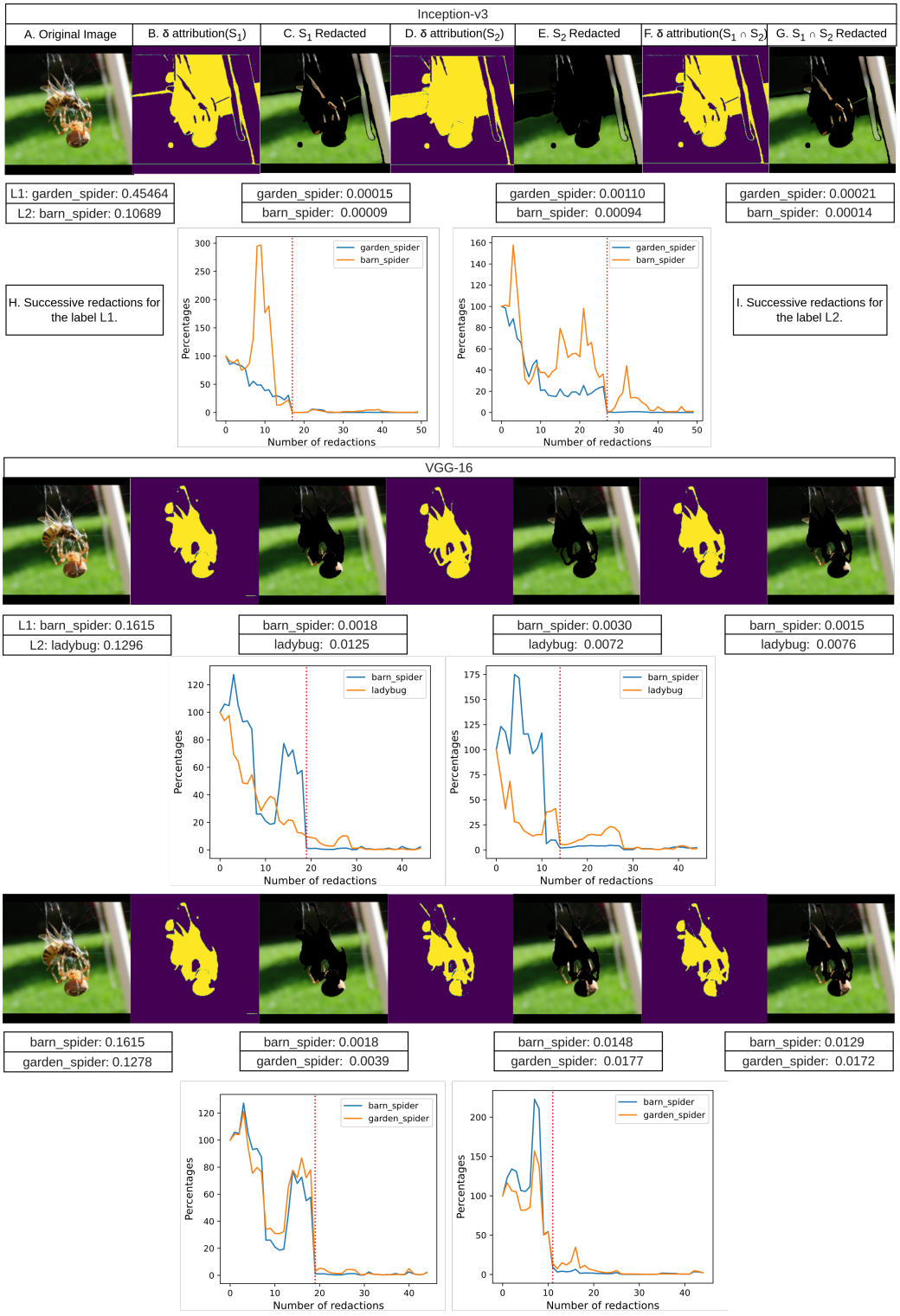



Figure 8: Illustrations of $\delta$-overlapping attributions with $\delta$=0.2 on Inception-v3 and VGG-16 for various labels using the same image as Figure 7 and following a similar pipeline as Figure 3.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please refer Sections 3, 5 and 6.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer section 7.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please refer sections 4, 5 and 6.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Provided the GitHub repository of the code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer sections 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experiments demonstrate performance of the proposed methods on a number of examples. We lack ground truth data for these examples, and therefore do not report statistical significance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer section A.3

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA] .

    Justification: There is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Please refer sections 4.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA] .

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA] .

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.