Soft Local Completeness: Rethinking Completeness in XAI

Supplementary Material

A. Appendix Overview

The appendix provides additional materials and detailed analyses to support the findings and discussions presented in the main paper. A summary of its contents is as follows:

- **Evaluation Metrics** (Sec. B): Detailed descriptions of the evaluation metrics used in this work.
- **Explanation Methods** (Sec. C): Descriptions of explanation methods included in the evaluation.
- **SLOC Implementation Details** (Sec. D): Precise implementation details of the SLOC method.
- **Additional Quantitative and Qualitative Results** (Sec. E): Quantitative and qualitative comparisons of SLOC and other state-of-the-art explanation methods on additional model architectures across all benchmarks.
- **Ablation Study** (Sec. F): Comprehensive analyses of the effects of varying design choices and hyperparameters in the SLOC method.
- **Computational Complexity** (Sec. G): An analysis of the computational complexity of SLOC.
- **Runtime Comparison** (Sec. H): A comparison of the runtime of SLOC and other explanation methods.
- **Notable Explanation Examples** (Sec. I): Presentation and discussion of several notable attribution maps produced by SLOC.
- **Sanity Checks** (Sec. J): Experimental validation of the SLOC method using parameter and data randomization sanity tests, as proposed in [3].
- **Motivation Formalization** (Sec. K): Formalization of SLOC motivation introduced in Sec. 3.1.
- **Additional Axioms** (Sec. L): A discussion of additional axioms, with an analysis of those satisfied by the SLOC method.
- **Limitations and Future Work** (Sec. M): An exploration of the limitations of SLOC, along with potential avenues for future research.

B. Evaluation Metrics

There is no single measure or test set which is generally acceptable for evaluating explanation maps. In this section, we detail the evaluation metrics used in our experiments, including faithfulness evaluation metrics, segmentation evaluation metrics, and the FunnyBirds evaluation metrics.

B.1. Faithfulness evaluation metrics

To ensure comparability, evaluations in this research follow earlier works [26, 28, 49, 57] (but are not limited to them). In general, the various tests entail different types of masking of the original input according to the explanation maps and investigating the change in the model's prediction for the masked input compared to its original prediction based on the unmasked input. The difference in predictions refer to the model's original top-predicted class. A detailed description of the relevant experiments can be found in Sec. 4.1.

In what follows, we list and define the different evaluation measures used in this research:

1. Perturbation tests entail a stepwise process in which pixels in the original image are gradually masked out according to their relevance score obtained from the explanation map [28]. At each step, an additional 5% of the pixels are removed and the original image is gradually blacked out. The performance of the explanation model is assessed by measuring the area under the curve (AUC) with respect to the model's prediction on the masked image compared to its prediction with respect to the original (unmasked) image. We consider two types of masking:
   (a) Positive perturbation (**POS**), in which we mask the pixels in decreasing order, from the highest relevance to the lowest, and expect to see a steep decrease in performance, indicating that the masked pixels are important to the classification score. Hence, for the POS perturbation test, lower values indicate better performance.
   (b) Negative perturbation (**NEG**), in which we mask the pixels in increasing order, from lowest to highest. A good explanation would maintain the accuracy of the model while removing pixels that are not related to the class of interest. Hence, for the NEG perturbation test, lower values indicate better performance.
   (c) NEG-POS Difference (**NPD**) - $(NEG - POS)$ captures the contrast between the complementary NEG and POS metrics, with higher values indicating better performance.

   In both positive and negative perturbations, we measure the area-under-the-curve (AUC), for erasing between 5%-95% of the pixels. As explained above, results are reported with respect to the 'predicted' or the 'target' (ground-truth) class.
2. The deletion and insertion metrics [57] are described as follows:
   (a) The deletion (**DEL**) metric measures a decrease in the probability of the class of interest as more and more important pixels are removed, where the importance of each pixel is obtained from the generated explanation map. A sharp drop and thus a low area under the probability curve (as a function of the

fraction of removed pixels) means a good explanation.

(b) In contrast, the insertion (**INS**) metric measures the increase in probability as more and more important pixels are revealed, with higher AUC indicative of a better explanation.

(c) INS-DEL Difference (**IDD**) - $(INS - DEL)$ captures the contrast between the complementary INS and DEL metrics, with higher values indicating better performance.

Note that there are several ways in which pixels can be removed from an image [30]. In this work, we remove pixels by setting their value to zero. Gradual removal or introduction of pixels is performed in steps of 0.05 i.e., remove or introduce 5% of the pixels on each step).

3. The Accuracy Information Curve (**AIC**) and the Softmax Information Curve (**SIC**) [49] metrics are both similar in spirit to the receiver operating characteristics (ROC). These measures are inspired by the Bokeh effect in photography [52], which consists of focusing on objects of interest while keeping the rest of the image blurred. In a similar fashion, we start with a completely blurred image and gradually sharpen the image areas that are deemed important by a given explanation method. Gradually sharpening the image areas increases the information content of the image. We then compare the explanation methods by measuring the approximate image entropy (e.g., compressed image size) and the model's performance (e.g., model accuracy).

(a) The AIC metric measures the accuracy of a model as a function of the amount of information provided to the explanation method. AIC is defined as the AUC of the accuracy vs. information plot. The information provided to the method is quantified by the fraction of input features that are considered during the explanation process.

(b) The SIC metric measures the information content of the output of a softmax classifier as a function of the amount of information provided to the explanation method. SIC is defined as the AUC of the entropy vs. information plot. The entropy of the softmax output is a measure of the uncertainty or randomness of the classifier's predictions. The information provided to the method is quantified by the fraction of input features that are considered during the explanation process.

B.2. FunnyBirds Evaluation Metrics

The **FunnyBirds** synthetic data generation process enables intervention and inspection at the object part level rather than at the pixel level. Each FunnyBird consists of five distinct parts: *beak, wings, feet, eyes,* and *tail*. The FunnyBirds evaluation protocol assesses explainability across three aspects: Completeness, Correctness, and Contrastivity, and provides an overall score, which is the average of these three aspects. The relevant experiments are described in Sec. 4.1, [44].

We define the following notation:

- $PI(\cdot)$ – **Part Importance Score**: The total attribution summed within a given part.
- $P(\cdot)$ – **Set of Important Parts**: The parts considered important, where a part is deemed important if its **importance score** constitutes at least $t\%$ of the total attribution.
- D – The FunnyBirds dataset, containing N images x_n, each associated with a class label c_n.
- f – The model under evaluation, where $f(x_n)$ denotes the logit for the target class, and $\hat{f}(x)$ denotes the predicted class.
- $e_f(x_n)$ – The explanation generated for x_n with respect to its target class c_n.

Correctness (Cor.)

Measures the faithfulness of the explanation with respect to the model.

- **Single Deletion Protocol (SD)**:
  Quantifies correctness by evaluating the correlation between Part Importance Scores and the change in logits when individual parts are removed from the image.

$$SD = \frac{1}{2} + \frac{1}{2N}\sum_{n=1}^{N} \rho\left(PI(e_f(x_n)), f(x_n) - f(x_n'')\right)$$

  where x_n'' denotes the image obtained by removing a single bird part from x_n. ρ denotes the Spearman rank-order correlation coefficient.

Completeness (Com.)

Evaluates whether the explanation accounts for all relevant factors influencing the model's decision. The score is computed as the mean of the averaged completeness metrics (CSDC, PC, and DC), and the Distractability D.

- **Controlled Synthetic Data Check (CSDC)**
  Tests whether the explanation highlights all relevant parts required for classification:

$$CSDC = \frac{1}{N}\sum_{n=1}^{N} \max_i \frac{|P(e_f(x_n)) \cap \mathcal{P}'_{c_n,i}|}{|\mathcal{P}'_{c_n,i}|}$$

  where $\mathcal{P}'_{c_n,i}$ represents the minimal set of parts sufficient for correctly classifying an image as c_n.

- **Preservation Check (PC)**
  Quantifies whether preserving only the important parts identified by the explanation maintains the model's original prediction:

$$PC = \frac{1}{N}\sum_{n=1}^{N} \left[\hat{f}(x_n') = \hat{f}(x_n)\right]$$

where x'_n is the image obtained by removing all bird parts except $P(e_f(x_n))$.

- **Deletion Check (DC)**
  Quantifies whether removing explanation identified important parts leads to a change in the model's prediction:

$$DC = \frac{1}{N}\sum_{n=1}^{N}\left[\hat{f}(x''_n) \neq \hat{f}(x_n)\right]$$

  where x''_n is the image obtained by removing the identified important parts $P(e_f(x_n))$.

- **Distractability (D)**
  Ensures that explanations do not highlight irrelevant parts:

$$D = 1 - \frac{1}{N}\sum_{n=1}^{N}\frac{|P(e_f(x_n)) \cap \mathcal{P}''_{f(x_n)}|}{|\mathcal{P}''_{f(x_n)}|}$$

  where $\mathcal{P}''_{f(x_n)}$ denotes the set of non-important parts.

Contrastivity (Con.)

Measures how well explanations distinguish between different class outputs. Explanations for different classes should highlight class-specific parts.

- **Target Sensitivity Protocol (TS)**

$$TS = \frac{1}{2N}\sum_{n=1}^{N}\begin{array}{l}[PI'(e_f(x_n,\hat{c}_1)) > PI'(e_f(x_n,\hat{c}_2))] + \\ [PI''(e_f(x_n,\hat{c}_1)) < PI''(e_f(x_n,\hat{c}_2))]\end{array}$$

  For each input, two classes $\hat{c}_1$ and $\hat{c}_2$ are chosen such that they have exactly two non-overlapping common parts. PI', PI'' denote the summed part importances of the two parts belonging to classes $\hat{c}_1$, $\hat{c}_2$ respectively.

Accuracy and Background Independence

The FunnyBirds evaluation protocol reports, in addition to the metrics, the model's accuracy (Acc.) and background independence (B.I.) with respect to the dataset. B.I. measures the model's sensitivity to the entire image, computed as the ratio of background objects such that, when removed, the target logit decreases by less than 5%. Accuracy is relevant because an overly simplified model may be explainable but may not effectively solve the task at hand. For more details see [44].

B.3. Segmentation evaluation metrics

To quantitatively assess the alignment between the generated explanation maps and human-annotated GT segmentations, we employ the following standard segmentation metrics:

Mean Average Precision (mAP) The mean Average Precision at a given Intersection over Union (IoU) threshold τ is computed as:

$$\text{mAP}_\tau = \frac{1}{N}\sum_{i=1}^{N}\text{AP}^i_\tau, \tag{6}$$

where AP^i_τ is the average precision for the i-th sample, and N is the total number of samples. The final mAP score is obtained by averaging across multiple IoU thresholds.

Mean Intersection over Union (mIoU) The IoU for a given sample is defined as:

$$\text{IoU} = \frac{|S \cap G|}{|S \cup G|}, \tag{7}$$

where S is the predicted saliency region and G is the ground truth segmentation mask. The mean IoU is then computed as:

$$\text{mIoU} = \frac{1}{N}\sum_{i=1}^{N}\text{IoU}^i. \tag{8}$$

Pixel Accuracy (PA) Pixel accuracy measures the proportion of correctly classified pixels and is given by:

$$\text{PA} = \frac{\sum_{i=1}^{N}|S_i \cap G_i|}{\sum_{i=1}^{N}|G_i|}. \tag{9}$$

This metric quantifies the overall agreement between the predicted and ground truth segmentations.

A detailed description of the relevant experiments can be found in Sec. 4.1.

C. Explanation Methods

1. Grad-CAM (**GC**) [59] integrates the activation maps from the last convolutional layer in the CNN by employing global average pooling on the gradients and utilizing them as weights for the feature map channels.
2. Grad-CAM++ (**GC++**) [26] is an advanced variant of Grad-CAM that utilizes a weighted average of the pixel-wise gradients to generate the activation map weights.
3. Integrated Gradients (**IG**) [64] integrates over the interpolated image gradients.
4. Guided IG (**GIG**) [50] improves upon Integrated Gradients by introducing the idea of an adaptive path method. By calculating integration along a different path than Integrated Gradients, high gradient areas are avoided which often leads to an overall reduction in irrelevant attributions.
5. The FullGrad (**FG**) method [63] provides a complete modeling approach of the gradient by also taking the gradient with respect to the bias term, and not just with respect to the input.

6. LayerCAM (**LC**) [48] utilizes both gradients and activations, but instead of using the Grad-CAM approach and applying pooling on the gradients, it treats the gradients as weights for the activations by assigning each location in the activations with an appropriate gradient location. The explanation map is computed with a location-wise product of the positive gradients (after ReLU) with the activations, and the map is then summed w.r.t. the activation channel, with a ReLU applied to the result.
7. Ablation-CAM (**AC**) [32] is an approach that only uses the channels of the activations. It takes each activation channel, masks it from the final map by zeroing out all locations of this channel in the explanation map produced by all the channels, computes the score on the masked explanation map (the map without the specific channel), and this score is used to assign an importance weight for every channel. At last, a weighted sum of the channels produces the final explanation map.
8. The Transformer attribution (**T-Attr**) [28] method computes the importance of each input token by analyzing the attention weights assigned to it during self-attention. Specifically, it computes the relevance score of each token as the sum of its attention weights across all layers of the Transformer. The intuition behind this approach is that tokens that receive more attention across different layers are likely more important for the final prediction. To obtain a more interpretable and localized visualization of the importance scores, the authors also propose a variant of the method called Layer-wise Relevance Propagation (LRP), which recursively distributes the relevance scores back to the input tokens based on their contribution to the intermediate representations.
9. Generic Attention Explainability (**GAE**) [27] is a generalization of T-Attr for explaining Bi-Modal transformers.
10. Deep Integrated Explanations (**DIX**) [14] is an advanced version of Integrated Gradients which performs integration over intermediate network representations, instead of input image.
11. **RISE** [57] creates perturbations by masking areas in the image through upsampling of randomly drawn low-resolution binary grids. The class score corresponding to each masked version of the image serves as an importance score for that specific mask. Finally, a linear combination of all masks, weighted by their importance, forms the final attribution map.
12. Meaningful Perturbation (**MP**) [39] employs gradient descent to optimize a mask that, when applied to the original image, generates a perturbation by occluding a small yet critical region.
13. Extremal Perturbations (**EP**) [38] - under a given mask-size constraint, EP employs gradient descent to optimize a mask of the specified size that maximally enhances the model's output. By iterating over different mask sizes, EP determines the smallest mask size capable of pushing the model's output beyond a predefined threshold.
14. Learning to Explain (**LTX**) [12] introduces a surrogate 'explainer' model pretrained to mask as much of the input as possible while preserving the original prediction, thereby ensuring the retained features are those most relevant to the model's decision. Then, LTX finetunes the attribution per specific example, while monitoring the metric of interest, thereby allowing the selection of the best-performing attribution w.r.t. to the metric at hand.

D. SLOC Implementation Details

This section details the implementation and hyperparameter configuration used in our SLOC implementation. The full implementation is available in our GitHub repository. In Sec. F, we provide comprehensive ablation studies analyzing the impact of various SLOC hyperparameters.

Mask Generation Phase During this phase, a total of $|\mathcal{M}| = 1000$ masks were generated. Specifically, the masks were generated by sampling entire regions corresponding to square patches, rather than individual pixels. To this end, we define a grid of patches of size $L \times L$, which is sufficiently large to encompass the entire image, even when the grid is offset by up to L pixels in either direction along the axes. The grid is then overlaid onto the image, ensuring full coverage, and two offsets, denoted as b_x and b_y, are drawn independently from a discrete uniform distribution over the set $\{0, ..., L-1\}$. The grid is subsequently shifted according to the sampled offsets, with b_x and b_y controlling the shift along the x-axis and y-axis, respectively. For each patch defined by the offset grid placement, a Bernoulli random variable with a success probability of p is sampled, and all pixels within the patch are set to the outcome (either 0 or 1). In our experiments, $\mathcal{M}$ consists of 500 sampled masks with $L = 32$ and another 500 sampled masks with $L = 56$.

Tuning of p We consider two approaches for tuning the probability parameter p. The first approach (used in SLOC) tunes p per input and patch size by first sampling 50 masks for each value of p in the range $[0.2, 0.8]$ with increments of 0.05. For each value of p, we compute the variance of the model's prediction across the corresponding 50 masks and select the value that yields the highest variance.

The motivation for selecting the probability that maximize the prediction variance is that an effective set of masks should include a diverse spectrum of masks that encompasses (1) masks that preserve the original prediction, thereby exposing important features accounting for the prediction, and (2) masks that substantially reduce the prediction score, indicating that essential information has been occluded. This contrast helps SLOC to better differentiate between the most and least important regions in the input.

The second approach (used in SLOC$_{\text{xp}}$) tunes p globally per model and patch size. Specifically, we perform a linear search to identify the value of p that achieves the best performance according to the IDD metric, using a designated set of 1000 examples. The first approach (used in SLOC) tunes p on a per-input basis by selecting the value that maximizes the variance of the model's prediction across the sampled masks for that specific input. The second approach tunes p globally by selecting the value that maximizes performance on a chosen metric of interest over a set of examples.

These two approaches represent an inherent trade-off: per-input tuning introduces a modest runtime overhead during inference, but it avoids optimizing for the IDD metric, thereby being less biased toward improved faithfulness scores. In contrast, per-model tuning requires access to a representative dataset in order to optimize p with respect to a specific metric. We also note that the per-model approach could alternatively tune p to maximize prediction variance globally over the representative dataset; however, we found this strategy to be suboptimal in practice.

Optimization Phase The elements of the attribution map $\mathbf{a}_{\mathbf{x}}^{y}$ were initialized by sampling from $\mathcal{N}(1, 0.1)$[3]. Then, the optimization was performed using gradient descent on $\mathcal{L}$ with respect to $\mathbf{a}_{\mathbf{x}}^{y}$ (Eq. 4), setting $\lambda_1 = 0.1$ and $\lambda_2 = 0.01$ for $T = 500$ update steps. We employed the Adam optimizer with a learning rate of $\gamma = 0.1$, with learning rate decay of 0.9 every 45 steps, momentum parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, with no weight decay. It is worth noting that in this work, the optimization employs gradient descent rather than stochastic gradient descent. Specifically, we perform a batch gradient update, considering the information from all masks for a single update step, as appears in Eq. 3. In the future, we plan to investigate the benefit from stochastic updates, each time utilizing different subsets of $\mathcal{M}$. While this approach may lead to less accurate update steps, it could potentially improve overall convergence due to stochasticity.

E. Additional Quantitative and Qualitative Results

Tables 7 and 8 present quantitative comparisons of SLOC with other state-of-the-art explanation methods across multiple faithfulness metrics on the IN dataset, using the RN and ViT-B models, respectively. Results on the VOC dataset, using the DN model, are reported in Tab. 9. We observe that SLOC consistently emerges as the top-performing method on average, with GIG, IG, and LTX as the closest competitors, depending on the model, metric, and dataset. The performance of the different SLOC variants is consistent with the trends observed in Tabs. 1-3. Finally, qualitative comparisons of SLOC and other methods using the RN and ViT-B models are presented in Figs. 6 and 7.

Figures 8 and 9 illustrate the results of the FB evaluation protocol for the RN and ViT-B models, respectively, with Tables 10 and 11 presenting the corresponding numerical results. The quantitative results plots follow the visualization from [44] for consistency, depicting the three evaluation aspects: Completeness (Com), Correctness (Cor), and Contrastivity (Con). Additionally, they display Accuracy (Acc) and Background Independence (B.I), which depend on the model and dataset but are independent of the explanation method, serving as sanity-check metrics. The overall score is shown at the center of each plot. Notably, The FB Completeness (Com) metric differs from the definition of completeness used in this paper; it evaluates the extent to which an explanation accounts for all aspects of the model's decision, rather than whether the attributions sum to the model's response. A detailed description of the metrics is provided in Sec. B.2.

The results indicate that SLOC achieves the highest overall score across both RN and ViT-B models, primarily due to its strong performance in Correctness (a metric closely related to faithfulness), while maintaining competitive results in the Completeness and Contrastivity metrics compared to the best-performing methods in those categories (DIX, T-ATTR, and IG).

Tables 12 and 13 present additional results for segmentation tests on the IN-Seg dataset using the DN and ViT-S models, respectively. We observe that DIX and T-ATTR are the top-performing methods, while SLOC yields mediocre to competitive results depending on the metric. It is important to reiterate that segmentation performance may not always serve as a strong indicator of explanation quality [57], but it does offer insights into how well the generated explanations align with human-annotated ground-truth segmentation maps. Higher segmentation accuracy may suggest that a method excels at object detection rather than identifying the most informative features driving the model's prediction. The most relevant features for explanation often do not cover the entire object but instead represent a subset that is critical to the model's decision.

F. Ablation Study

We present extensive ablation studies investigating the significance of the different loss terms in Eq. 4, and the hyperparameters of SLOC. The ablation studies are conducted on

[3] This initialization was chosen for simplicity and has proven effective in practice, despite being arbitrary. An alternative strategy could involve setting the mean to the model's response divided by the number of input elements, along with a standard deviation that ensures predominantly positive values. However, such approaches remain unexplored and are left for future work.

Method	POS↓	NEG↑	DEL↓	INS↑	NPD↑	IDD↑	AIC↑	SIC↑
SLOC_{m}	10.96	**75.84**	8.68	**64.19**	**64.87**	**55.5**	**79.44**	**78.13**
SLOC	10.65	69.56	8.32	58.33	58.91	50.01	78.5	77.04
SLOC_{xp}	10.74	68.62	8.41	57.41	57.88	49.01	77.75	76.78
AC	16.7	66.96	12.76	55.71	50.26	42.95	77.17	74.59
DIX	10.21	58.33	7.83	48.16	48.11	40.33	71.15	68.81
EP	14.9	66.41	11.5	54.51	51.51	43.01	75.06	73.96
FG	16.79	65.9	12.94	54.9	49.11	41.96	74.16	71.54
GC	16.37	68.04	12.56	56.65	51.67	44.1	77.33	75.1
GC++	16.81	66.85	12.85	55.54	50.04	42.68	76.82	74.54
GIG	**9.4**	45.28	**7.68**	37.71	35.89	30.03	57.52	54.51
IG	9.9	44.22	7.76	37.14	34.32	29.38	56.56	54.23
LC	17.04	66.58	13.0	55.25	49.54	42.25	76.46	74.33
LTX	14.98	69.88	11.7	57.74	54.91	46.03	76.69	74.29
MP	17.16	50.81	13.6	41.34	33.65	27.74	64.71	62.52
RISE	15.8	62.3	12.04	51.93	46.5	39.89	77.31	74.77

Table 7. Faithfulness results for all combinations of method and metric, using the RN model on the IN dataset.

Method	POS↓	NEG↑	DEL↓	INS↑	NPD↑	IDD↑	AIC↑	SIC↑
SLOC_{m}	**21.54**	**89.01**	**14.93**	**63.05**	**67.48**	**48.12**	**85.42**	81.82
SLOC	22.39	85.72	15.7	59.85	62.73	43.71	84.82	**81.86**
SLOC_{xp}	22.36	85.10	15.56	59.27	62.98	43.88	84.74	81.58
DIX	32.09	77.01	21.14	51.17	44.92	30.03	79.88	74.92
EP	41.24	82.95	25.18	58.99	41.72	33.81	81.59	77.32
GAE	33.16	76.88	21.95	51.12	43.72	29.17	79.02	74.6
LTX	28.3	80.74	18.8	55.52	52.44	36.72	79.65	74.98
MP	36.63	78.49	23.83	52.82	41.87	28.99	80.58	76.11
RISE	49.7	77.42	32.7	50.09	27.72	17.4	76.65	72.16
TATTR	32.8	77.24	21.49	51.56	44.44	30.07	80.04	74.79

Table 8. Faithfulness results for all combinations of method and metric, using the ViT-B model on the IN dataset.

the ViT-S model.

Loss terms Table 14 presents the effect of ablating on different terms in $\mathcal{L}$ (Eq. 4) to faithfulness performance: SLOC_{xL1} and SLOC_{xTV} are versions that ablate on the L1 term and TV term by setting $\lambda_1 = 0$ and $\lambda_2 = 0$, respectively. $\text{SLOC}_{\text{xL1xTV}}$ is a version that simply set $\mathcal{L}$ to $\mathcal{L}_c$ (Eq. 2) which is equivalent to setting both $\lambda_1 = 0$ and $\lambda_2 = 0$. Finally, SLOC is our method. Tables 15 and 16 summarize the faithfulness results for various values of λ_1 and λ_2.

The results in Tables 14–16 indicate that both regularization terms, TV and L1, introduce trade-offs across the explanation metrics. For example, L1 regularization without TV (SLOC_{xTV}) outperforms SLOC across all metrics except for AIC and SIC. In contrast, applying TV without L1 (SLOC_{xL1}) leads to degradations in NEG and INS (and the corresponding summary metrics), while the remaining metrics remain largely unaffected. Finally, when no regularization is applied at all ($\text{SLOC}_{\text{xL1xTV}}$), we observe improvements in POS and DEL but significant degradations in NEG, INS, AIC, and SIC.

These results suggest that both regularization terms yield mixed effects on faithfulness metrics, highlighting no consistent trend. However, faithfulness is only one aspect of explanation quality. Therefore, in Tables 17 and 18, we further examine segmentation performance across varying values of λ_1 and λ_2, respectively. In this case, we observe that applying TV and L1 leads to marginal improvements in segmentation performance. This aligns with the observation that the regularization terms contribute to more focused and visually 'clean' explanations from a human perspective.

Patch size L Table 19 presents a comparative analysis of different settings of the patch size L. The results indicate that increasing the patch size consistently improves performance on the NEG, INS, AIC, and SIC metrics, while leading to a degradation in the POS and DEL metrics beyond a patch size of $L = 32$. This observation suggests that different metrics may benefit from different patch size settings. Therefore, in our experiments, we consider a combination of two sets of masks—one generated with a patch size of 32 and another with 56—as this configuration achieves balanced improvements across all metrics, yields the best values for the NPD and IDD summary metrics, and delivers state-of-the-art performance.

Number of sampled masks $|\mathcal{M}|$ Table 20 presents a comparison between different choices of $|\mathcal{M}|$. Notably, we observe that beginning from $|\mathcal{M}| = 1000$ SLOC exhibits

Method	POS↓	NEG↑	DEL↓	INS↑	NPD↑	IDD↑	AIC↑	SIC↑
SLOC$_m$	6.24	**61.32**	4.19	**45.15**	**55.09**	**40.96**	**68.6**	**71.89**
SLOC	**6.19**	53.4	**4.05**	39.01	47.2	34.97	65.63	70.92
SLOC$_{xp}$	6.40	52.61	4.08	38.11	45.74	34.04	66.65	70.58
AC	10.06	50.15	6.4	33.85	40.09	27.45	61.46	65.05
DIX	8.2	43.89	5.19	29.31	35.7	24.12	58.67	61.64
EP	9.08	49.12	5.91	33.1	40.04	27.19	63.7	65.53
FG	9.42	27.99	6.36	19.43	18.57	13.07	36.75	39.37
GC	9.85	51.83	6.24	34.89	41.99	28.64	63.16	65.68
GC++	10.11	49.43	6.5	33.03	39.31	26.54	61.42	64.15
GIG	6.47	30.37	4.3	21.46	23.9	17.16	41.59	44.56
IG	7.73	30.88	5.09	21.56	23.15	16.47	41.59	42.76
LC	10.14	49.29	6.52	32.94	39.15	26.42	61.25	64.39
LTX	8.98	54.84	5.92	36.93	45.86	31.01	61.37	65.11
RISE	9.38	43.75	6.02	30.78	34.38	24.76	61.99	64.67

Table 9. Faithfulness results for all combinations of method and metric, using the DN model on the VOC dataset.

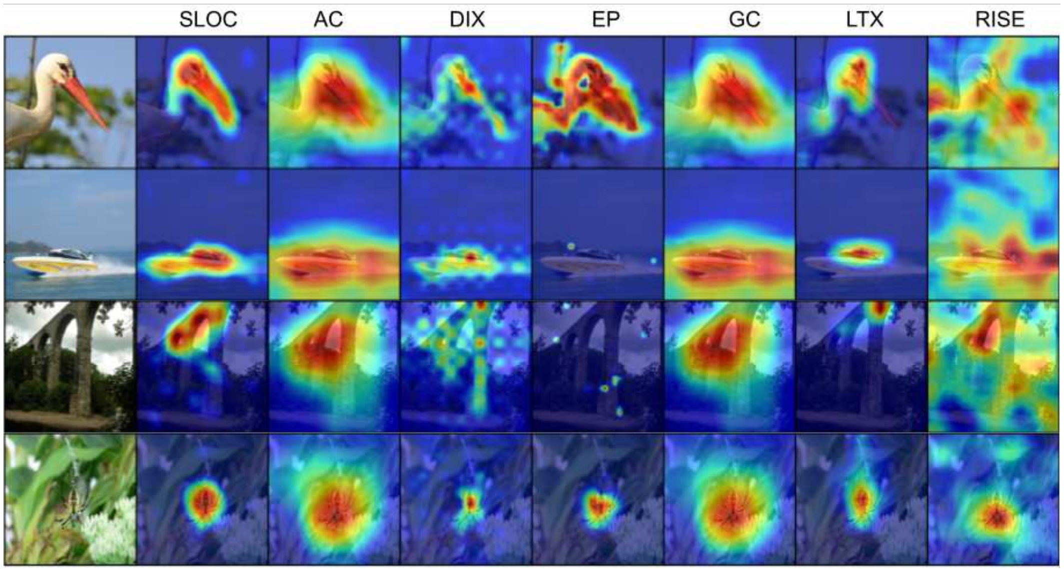

Figure 6. Qualitative comparison of attributions produced by different methods, using the RN model w.r.t. the classes (top to bottom): ’white stork’, ’speed boat’, ’viaduct’, ’black and gold garden spider’.

Method	Completeness↑	Correctness↑	Contrastivity↑	Overall↑
SLOC	0.85	0.60	**0.87**	**0.78**
SLOC$_{xp}$	0.86	**0.61**	0.85	0.77
AC	0.73	0.56	0.80	0.70
DIX	0.74	0.55	**0.87**	0.72
EP	0.82	0.57	0.8	0.73
FG	0.75	0.56	0.78	0.69
GC	0.74	0.55	0.86	0.72
GC++	0.74	0.55	0.87	0.72
GIG	0.65	0.54	0.49	0.56
IG	**0.86**	0.55	0.49	0.63
LC	0.74	0.55	0.86	0.72
RISE	0.70	0.56	0.61	0.62

Table 10. FunnyBirds evaluation results for the RN model.

Method	Completeness↑	Correctness↑	Contrastivity↑	Overall↑
SLOC	0.91	0.77	0.96	**0.88**
SLOC$_{xp}$	**0.92**	0.77	0.90	0.86
DIX	0.9	0.76	**0.97**	0.87
EP	0.89	0.76	0.71	0.79
RISE	0.78	**0.79**	0.75	0.77
T-Attr	0.9	0.74	0.95	0.87

Table 11. FunnyBirds evaluation results for the ViT-B model.

commendable performance. Although larger values of $|\mathcal{M}|$ may offer potential improvements, our findings suggest that $|\mathcal{M}| = 1000$ (comprising 500 masks with a patch size of 32 and another 500 with a patch size of 56) is sufficient to achieve state-of-the-art performance while maintaining relatively fast runtimes.

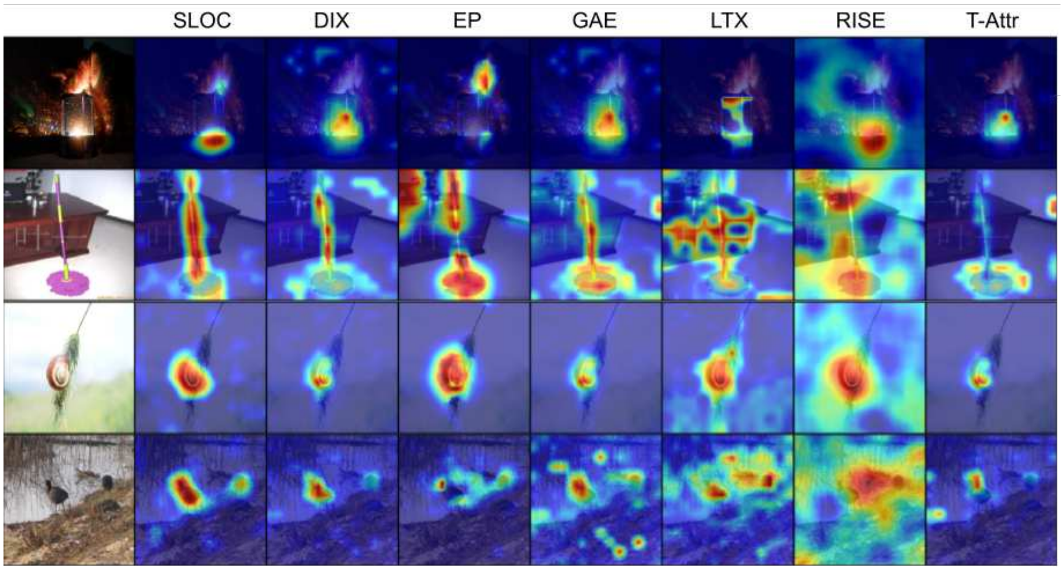

Figure 7. Qualitative comparison of attributions produced by different methods, using the ViT-B model w.r.t. the classes (top to bottom): 'spotlight', 'swab', 'snail', 'American coot'.

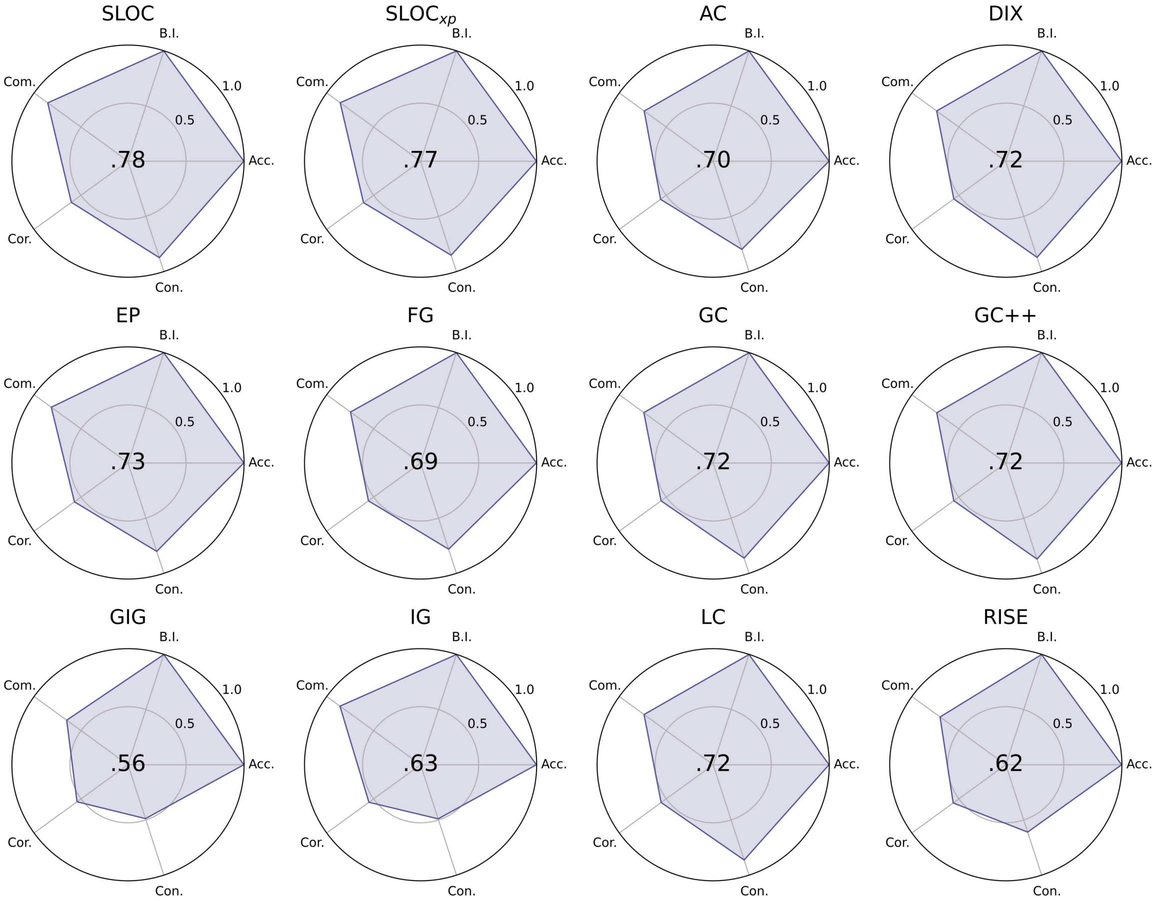

Figure 8. FunnyBirds evaluation results for model RN. See Sec. B.2 for metric descriptions.

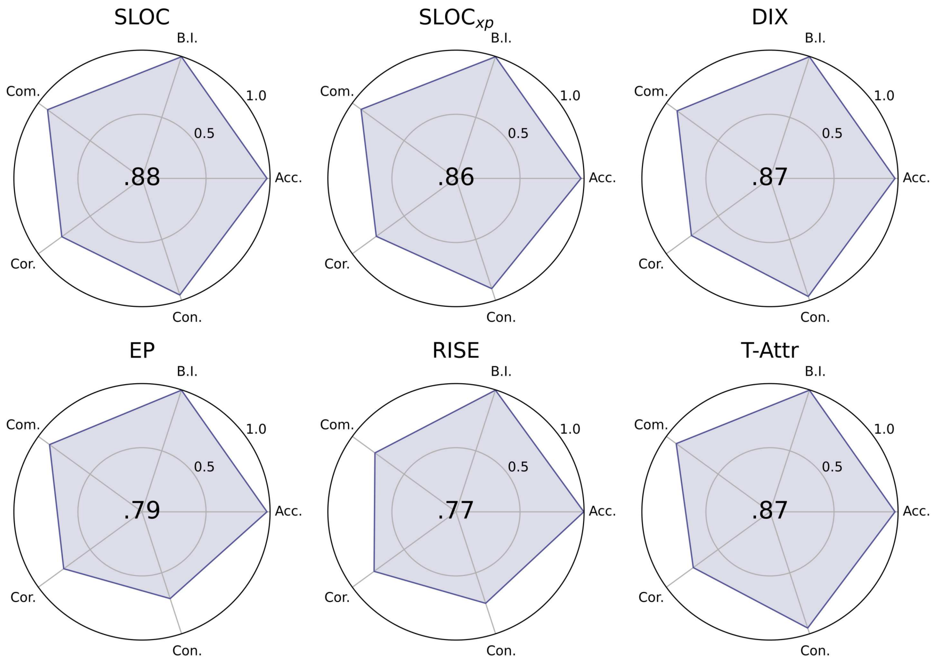
SLOC
SLOC$_{xp}$
DIX
EP
RISE
T-Attr

Figure 9. FunnyBirds evaluation results for model ViT-B. See Sec. B.2 for metric descriptions.

Method	SLOC	AC	DIX	EP	FG	GC	GC++	GIG	IG	LC	LTX	RISE
mIoU↑	0.55	0.56	0.64	0.51	0.45	0.55	0.56	0.49	0.47	0.56	0.56	0.50
mAP↑	0.79	0.86	0.87	0.77	0.76	0.86	0.87	0.77	0.77	0.87	0.85	0.79
PA↑	0.75	0.73	0.80	0.72	0.66	0.73	0.74	0.72	0.69	0.74	0.74	0.69

Table 12. Segmentation results for the DN model.

Method	SLOC	DIX	EP	GAE	LTX	MP	RISE	T-Attr
mIoU↑	0.52	0.63	0.50	0.61	0.56	0.55	0.50	0.68
mAP↑	0.76	0.81	0.76	0.79	0.81	0.74	0.75	0.83
PA↑	0.72	0.79	0.71	0.78	0.72	0.74	0.68	0.82

Table 13. Segmentation results for the ViT-S model.

	POS↓	NEG↑	DEL↓	INS↑	NPD↑	IDD↑	AIC↑	SIC↑
SLOC$_{\text{xTV}}$	13.33	78.16	11.0	67.44	64.83	56.44	80.91	79.53
SLOC$_{\text{xL1}}$	15.26	76.84	12.54	65.83	61.58	53.29	84.11	82.15
SLOC$_{\text{xL1xTV}}$	13.89	73.60	11.35	62.95	59.70	51.60	77.80	76.28
SLOC	15.35	77.87	12.59	66.76	62.52	54.17	83.87	82.2

Table 14. Faithfulness evaluation. Ablation study on the regularization terms in Eq. 4.

λ_1	POS↓	NEG↑	DEL↓	INS↑	NPD↑	IDD↑	AIC↑	SIC↑
0.0	15.26	76.84	12.54	65.83	61.58	53.29	84.11	82.15
0.01	15.35	77.87	12.59	66.76	62.52	54.17	83.87	82.2
0.05	15.45	77.56	12.65	67.18	62.11	54.53	83.5	81.74
0.1	15.65	77.26	12.7	66.71	61.61	54.01	83.04	81.5
0.25	15.71	76.34	12.73	66.09	60.63	53.36	82.33	81.29
0.5	15.98	74.98	13.04	64.85	59.0	51.81	80.9	80.88
1.0	17.51	69.7	14.08	60.19	52.19	46.11	77.81	78.21

Table 15. Faithfulness evaluation. Ablation study on λ_1 - the coefficient of the L1 regularization term in Eq. 4.

λ_2	POS↓	NEG↑	DEL↓	INS↑	NPD↑	IDD↑	AIC↑	SIC↑
0.0	13.33	78.16	11.0	67.44	64.83	56.44	80.91	79.53
0.01	14.33	79.27	11.85	68.63	64.94	56.78	82.56	81.17
0.1	15.34	77.78	12.59	66.78	62.44	54.18	83.6	81.92
0.2	15.84	78.0	12.97	66.57	62.16	53.6	83.51	81.84
0.5	16.74	78.38	13.69	66.69	61.64	52.99	83.67	82.44
1.0	17.7	78.93	14.37	66.95	61.24	52.58	84.0	82.36

Table 16. Faithfulness evaluation. Ablation study on λ_2 - the coefficient of the TV regularization term in Eq. 4.

λ_1	0	0.01	0.05	0.1	0.25	0.5	0.1
mIoU↑	0.52	0.55	0.55	0.54	0.52	0.49	0.43
mAP↑	0.79	0.81	0.81	0.81	0.79	0.76	0.71
PA↑	0.71	0.74	0.75	0.74	0.72	0.70	0.64

Table 17. Segmentation evaluation. Ablation study on λ_1 - the coefficient of the L1 regularization term in Eq. 4, using the RN model.

Probability p (Bernoulli parameter) for activating mask patches Table 21 presents a comparison of different choices for the parameter p, which determines the probability of setting a mask patch to 1. Notably, the last row (SLOC)—corresponding to tuning p per input—outperforms all fixed probability values across all metrics except for $p = 0.3$. We note that the optimal fixed p varies across different models (this ablation was conducted on the ViT-S model), highlighting an inherent trade-off in the choice between tuning p per input (SLOC) or per model

λ_2	0	0.01	0.1	0.2	0.5	0.1
mIoU↑	0.54	0.54	0.55	0.55	0.55	0.55
mAP↑	0.8	0.8	0.8	0.81	0.81	0.82
PA↑	0.73	0.73	0.74	0.74	0.74	0.74

Table 18. Segmentation evaluation. Ablation study on λ_2 - the coefficient of the TV regularization term in Eq. 4, using the RN model.

L	POS↓	NEG↑	DEL↓	INS↑	NPD↑	IDD↑	AIC↑	SIC↑
8	16.42	74.02	13.54	63.52	57.6	49.98	79.42	77.31
16	15.18	75.82	12.44	65.15	60.64	52.71	81.66	80.33
32	14.86	76.55	12.31	65.45	61.69	53.14	83.04	80.88
40	15.67	76.67	12.84	65.52	61.0	52.68	83.08	81.01
48	16.43	77.39	13.35	65.87	60.95	52.53	83.23	81.25
56	17.01	77.49	13.86	65.91	60.48	52.05	82.89	80.98
64	17.69	77.68	14.49	65.75	59.99	51.26	82.48	81.1
SLOC	15.7	77.38	12.83	66.1	61.68	53.26	83.19	81.36

Table 19. Faithfulness performance across different patch size settings.

$\lvert\mathcal{M}\rvert$	POS↓	NEG↑	DEL↓	INS↑	NPD↑	IDD↑	AIC↑	SIC↑
10	32.91	60.32	26.63	49.16	27.41	22.53	69.26	66.88
100	19.76	70.18	16.15	58.29	50.42	42.14	77.97	75.48
250	17.17	73.96	14.09	62.43	56.79	48.35	80.43	78.47
500	16.36	76.16	13.25	64.51	59.8	51.26	82.44	80.65
750	15.95	77.24	12.93	65.62	61.3	52.7	82.91	81.23
1000	15.7	77.38	12.83	66.1	61.68	53.26	83.19	81.36
1250	15.65	77.55	12.79	66.33	61.9	53.54	83.53	81.64
1500	15.54	78.06	12.77	66.7	62.52	53.93	83.25	81.66
2000	15.51	78.25	12.68	66.95	62.74	54.27	83.66	82.03

Table 20. Faithfulness performance for varying numbers of drawn masks.

p	POS↓	NEG↑	DEL↓	INS↑	NPD↑	IDD↑	AIC↑	SIC↑
0.1	18.35	76.28	14.67	64.86	57.93	50.2	81.53	79.85
0.2	16.54	77.76	13.35	66.33	61.21	52.98	82.75	81.17
0.3	15.73	78.21	12.80	66.82	62.49	54.02	83.36	81.56
0.4	16.1	77.75	13.33	66.52	61.65	53.19	82.5	80.96
0.5	17.39	76.11	14.77	65.15	58.71	50.38	81.83	79.62
0.6	19.56	74.38	16.81	63.16	54.82	46.35	80.1	77.79
0.7	23.14	71.45	19.74	60.05	48.31	40.31	78.02	75.1
0.8	26.07	67.94	22.35	56.68	41.87	34.33	75.73	72.63
0.9	31.24	65.15	26.57	53.88	33.91	27.31	72.46	69.19
SLOC	15.35	77.87	12.59	66.76	62.52	54.17	83.87	82.20

Table 21. Faithfulness performance for varying patch probability.

(SLOC$_{\text{xp}}$): in SLOC, p is being tuned during inference (for the specific input), while in SLOC$_{\text{xp}}$, access to a representative dataset is required to tune p in advance for each model.

T	POS↓	NEG↑	DEL↓	INS↑	NPD↑	IDD↑	AIC↑	SIC↑
25	20.12	75.85	16.41	63.68	55.73	47.27	80.1	77.64
50	17.03	79.86	13.75	68.47	62.83	54.72	83.26	81.77
75	17.19	80.26	13.99	68.9	63.07	54.9	83.9	82.7
100	17.24	80.2	14.12	68.8	62.95	54.68	84.21	82.54
200	16.52	79.1	13.54	67.86	62.57	54.32	83.41	82.44
250	16.19	78.76	13.28	67.59	62.57	54.31	83.71	82.26
500	15.34	77.86	12.59	66.8	62.52	54.21	83.43	82.11
750	14.88	77.46	12.26	66.36	62.58	54.11	83.82	81.97
1000	14.69	77.37	12.09	66.19	62.67	54.1	83.01	82.1

Table 22. Faithfulness performance for varying numbers of gradient update steps (iterations).

Number of gradient update steps T Table 22 reports the results for varying numbers of update steps (iterations) in the SLOC optimization process. We observe that the optimal number of iterations varies across metrics. While POS and DEL favor higher values of T, the best results for NEG and INS are obtained between 50–100 iterations, and the summary metrics favor 75 iterations. Since faithfulness metrics provide only a partial assessment of explanation quality, we found that setting $T = 500$ offers the best balance across all benchmarks (faithfulness, segmentation, FB) and produces satisfactory attributions from a human perspective.

Empirically, a higher number of iterations results in lower DEL and POS scores, which in turn lead to more focused and compact attributions. This can be explained by the fact that low AUC values in the DEL and POS metrics are encouraged by a sharp drop in the metric curves, caused by masking a relatively small percentage of elements in the image that correspond to the most influential features (assuming the explanation method indeed highlight the most influential ones). This indicates that the attribution concentrates on a small, compact region—often perceived as more meaningful and interpretable from a human perspective.

SLOC attributions aggregation vs. single-run approach Due to the inherent stochasticity in SLOC, arising from both mask sampling and the optimization process, different runs of SLOC on the same input may yield different attribution maps. This motivates two lines of investigation: (1) we examine the effect of aggregating N attribution maps produced by N independent runs of SLOC into a single, combined attribution map; and (2) since N runs of SLOC effectively involve sampling a total of $N|\mathcal{M}|$ masks, we study the behavior of SLOC when using $N|\mathcal{M}|$ masks in a single run, as the alternative of aggregating N independently generated attribution maps.

We consider four aggregation methods: **mean**, **median**, **minimum**, and **product**. The results for each method are summarized in Tabs. 23, 24, 25, and 26, respectively. Overall, we observe that median and minimum aggregations improve faithfulness results, followed by mean aggregation, which provides slight to negligible improvement. Product aggregation does not yield any gain and even results in slight degradations in some metrics. We note that for product aggregation, we apply ReLU to each attribution map before computing the product to avoid sign issues, as SLOC can produce negative attributions. Given this choice, we believe the degradation is due to the zeroing of all negative values, combined with the fact that product aggregation is equivalent to an intersection operation, which may be too aggressive.

A demonstration of the resulting attribution maps from the four different aggregation methods across varying values of N appears in the first four rows of Fig. 10. For $N = 1$, the attribution map is identical for all methods, as no aggregation is applied. As the number of attributions N increases, we observe a artifact reduction effect across all aggregation methods. Arguably, minimum aggregation produces the 'cleanest' attribution while still highlighting most of the influential features related to the prediction. Visually, mean and median aggregations produce similar attribution maps, with slightly less artifacts in the median aggregation. Product aggregation, while producing very clean attributions, fails to capture the significant regions and features of the object associated with the predicted classes.

Table 27 summarizes the results of the alternative approach: applying a single run of SLOC with $N|\mathcal{M}|$ sampled masks (for different values of N), allowing for a fair comparison to the aggregation methods. We observe improvements across all faithfulness metrics as N, and consequently the number of sampled masks, increases. However, this improvement plateaus for the majority of metrics starting from $N > 2$.

When comparing the single-run approach to the aggregation methods, we observe a slight improvement in POS and DEL compared to the leading aggregation methods (minimum and median). For AIC and SIC, the performance is comparable, while for the remaining metrics, the leading aggregation methods produce better results than the single-run approach.

The last row in Fig. 10 (the 'base' row) presents the resulting attributions for the single-run approach across varying values of N. Similar to the aggregation methods, we observe an artifact reduction effect as N increases. Arguably, the single-run method with $N = 10$ best highlights the important features in the image from a human perspective, while producing fewer artifacts than median aggregation. Minimum aggregation, while generating the least amount of artifacts, presents less coherent highlighting of the object compared to the single-run method.

Overall, we conclude that minimum and median aggregation methods show potential to improve both faithfulness results and visual quality. However, a comparable improvement can also be achieved by simply increasing the number of sampled masks in a single run of SLOC.

N	POS↓	NEG↑	DEL↓	INS↑	NPD↑	IDD↑	AIC↑	SIC↑
1	15.79	77.49	12.95	66.16	61.7	53.22	83.21	81.19
2	15.35	78.05	12.59	66.64	62.7	54.05	83.5	81.96
3	15.4	78.13	12.52	66.74	62.74	54.22	83.74	82.41
4	15.34	78.03	12.5	66.77	62.69	54.26	83.75	82.19
5	15.4	78.12	12.49	66.88	62.72	54.39	83.58	82.18
6	15.41	78.18	12.52	66.87	62.77	54.35	84.08	82.31
7	15.37	78.22	12.48	66.85	62.85	54.37	83.84	82.2
8	15.46	78.24	12.48	66.88	62.77	54.4	84.03	82.26
9	15.41	78.38	12.47	66.91	62.97	54.44	83.77	82.45
10	15.4	78.44	12.45	66.94	63.05	54.49	83.96	82.6

Table 23. Evaluating the effect of combining N generated attributions by **mean aggregation** to produce the final attribution map.

N	POS↓	NEG↑	DEL↓	INS↑	NPD↑	IDD↑	AIC↑	SIC↑
1	15.79	77.49	12.95	66.16	61.7	53.22	82.77	81.36
2	15.4	78.76	12.68	67.51	63.36	54.83	83.58	81.68
3	15.42	79.73	12.68	68.51	64.31	55.83	83.61	81.96
4	15.27	80.13	12.66	68.84	64.86	56.18	83.75	82.17
5	15.39	80.24	12.66	69.18	64.86	56.52	83.67	82.53
6	15.41	80.42	12.68	69.41	65.02	56.73	83.84	82.37
7	15.48	80.5	12.69	69.66	65.02	56.96	83.63	82.27
8	15.35	80.84	12.66	69.68	65.49	57.02	84.34	82.38
9	15.58	80.84	12.75	69.87	65.26	57.12	84.11	82.55
10	15.36	80.7	12.74	69.82	65.34	57.08	83.9	82.74

Table 24. Evaluating the effect of combining N generated attributions by **median aggregation** to produce the final attribution map.

N	POS↓	NEG↑	DEL↓	INS↑	NPD↑	IDD↑	AIC↑	SIC↑
1	15.79	77.49	12.95	66.16	61.7	53.22	82.62	81.57
2	15.4	78.76	12.68	67.51	63.36	54.83	83.09	81.6
3	15.48	79.06	12.57	67.9	63.57	55.34	83.19	81.61
4	15.38	79.07	12.58	68.04	63.69	55.46	83.34	81.81
5	15.37	79.08	12.53	68.14	63.72	55.6	83.41	81.74
6	15.3	79.19	12.47	68.26	63.89	55.8	83.5	81.93
7	15.27	79.2	12.49	68.19	63.93	55.7	83.1	81.88
8	15.43	79.21	12.51	68.24	63.78	55.73	83.28	81.89
9	15.29	79.28	12.5	68.4	64.0	55.9	83.63	81.99
10	15.29	79.65	12.49	68.51	64.36	56.02	82.98	82.11

Table 25. Evaluating the effect of combining N generated attributions by **minimum aggregation** to produce the final attribution map.

N	POS↓	NEG↑	DEL↓	INS↑	NPD↑	IDD↑	AIC↑	SIC↑
1	15.79	77.49	12.95	66.16	61.7	53.22	82.75	81.23
2	15.48	77.24	12.67	65.75	61.76	53.07	83.12	81.89
3	15.3	77.01	12.58	65.48	61.71	52.9	83.3	81.62
4	15.35	76.77	12.59	65.27	61.43	52.67	83.16	81.56
5	15.38	76.86	12.6	65.22	61.47	52.61	82.81	81.67
6	15.58	76.86	12.67	65.02	61.28	52.35	83.31	81.59
7	15.67	76.87	12.72	64.88	61.2	52.16	82.73	81.36
8	15.67	76.46	12.72	64.62	60.79	51.9	82.5	81.34
9	15.7	76.63	12.75	64.63	60.93	51.88	82.63	81.31
10	15.8	76.16	12.74	64.48	60.37	51.73	82.76	80.91

Table 26. Evaluating the effect of combining N generated attributions by **multiplying** them element-wise to produce the final attribution map.

G. Computational Complexity

The computational complexity of SLOC is determined by the number of gradient update steps T and the number of forwards passes through the model which corresponds to the number of sampled masks, $|\mathcal{M}|$. In our implementation, we use the same set of masks throughout the entire optimization process. This allows for precomputation of

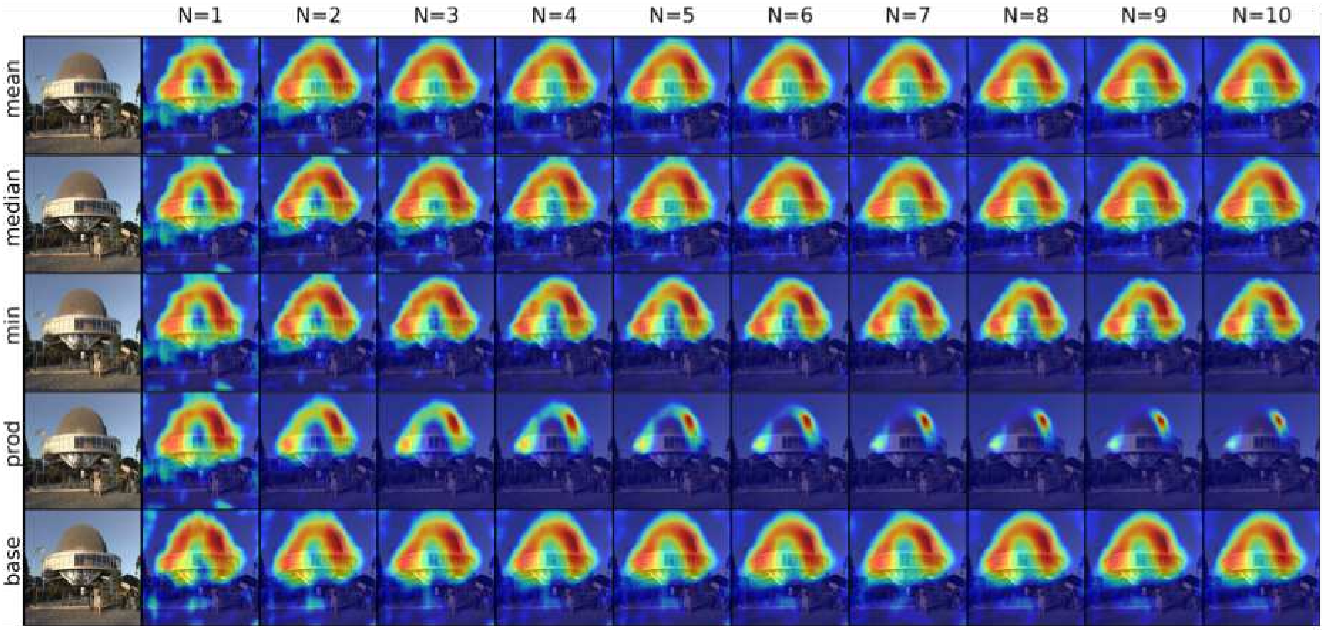

Figure 10. Aggregation of N attribution maps produced by independent runs of SLOC using different aggregation methods (mean, median, minimum, and product), compared to a single run of SLOC with $N|\mathcal{M}|$ masks.

N	POS↓	NEG↑	DEL↓	INS↑	NPD↑	IDD↑	AIC↑	SIC↑
1	15.38	77.54	12.74	66.16	62.16	53.42	83.48	81.18
2	15.16	77.73	12.51	66.71	62.57	54.2	83.58	81.99
3	15.1	78.3	12.53	67.11	63.21	54.57	84.06	82.0
4	15.15	78.25	12.42	67.12	63.1	54.7	83.54	82.22
5	15.13	78.0	12.37	67.23	62.87	54.86	83.94	82.36
6	15.17	78.18	12.43	67.3	63.01	54.87	83.83	82.34
7	15.17	78.11	12.42	67.15	62.94	54.73	84.13	82.29
8	15.15	78.22	12.38	67.32	63.07	54.94	83.92	82.49
9	15.29	78.41	12.38	67.33	63.13	54.95	84.03	82.46
10	15.11	78.24	12.33	67.31	63.13	54.98	83.89	82.6

Table 27. Single-run experiment. Each row reports faithfulness results obtained by SLOC using $N|\mathcal{M}|$ sampled masks for varying values of N (to match the total number of masks used across N attributions in the aggregation experiments).

the model's response $r(\mathbf{x^m})$ once, which is then reused in subsequent optimization steps. Specifically, before the optimization begins, each masked version of the input $\mathbf{x^m}$ is passed through the model once, and the corresponding set of model responses, $\{r(\mathbf{x^m})\}_{\mathbf{m}\in\mathcal{M}}$, is stored. Additionally, for each mask, we precompute and store $|\mathbf{m}|$ which is required for normalizing the completeness gap, as shown in Eq. 2.

Subsequently, the optimization proceeds through a series of gradient updates, each relying on a sequence of lightweight tensor arithmetic operations as outlined in Eq. 3. Importantly, during the optimization process, both $r(\mathbf{x^m})$ and $|\mathbf{m}|$ are fixed, precomputed values that remain constant across all gradient updates. The only term that changes with each update is the attribution map $\mathbf{a}^y_\mathbf{x}$, which is refined iteratively in each step.

As a result, at least in theory, SLOC optimization is more efficient than other optimization-based attribution methods [12, 39] that require both forward and backward passes (due to gradient backpropagation through the model) at each update step, leading to higher computational costs per update. Additionally, SLOC can become computationally lighter than path integration methods, if the number of interpolations in the integral approximation is equivalent to the number of masks drawn in SLOC. This is because each interpolation step in the path integration requires a forward-backward pass for gradient computation, whereas each mask in SLOC only requires a forward pass. In Sec. H, we present runtime comparisons demonstrating that SLOC achieves faster runtimes relative to other explanation methods.

It is important to note that the computation for each mask is independent, hence embarrassingly parallel. As long as the computational resources support accommodating the batch of masks in GPU memory, including the propagation of the perturbed (masked) inputs through the model, the gradients for all sub-maps can be computed in parallel via a single forward pass. Accordingly, the precomputation of the model responses for all perturbed inputs is also embarrassingly parallel and can be efficiently achieved in a single pass through the model using GPU parallelization.

	SLOC	SLOC$_{\text{xp}}$	DIX	EP	GAE	LTX	MP	RISE	T-Attr
Runtime (seconds)	6.79	3.43	0.4	13.25	0.03	7.1	4.05	9.11	1.02

Table 28. Runtime comparison between SLOC and other attribution methods using the ViT-S model.

H. Runtime Comparison

We evaluated the runtime efficiency of SLOC in comparison to other explanation methods by running each method on the same random subset of 100 examples from the IN dataset using the ViT-S model. The resulting runtimes are presented in Tab. 28. We observe that SLOC variants demonstrate competitive to superior runtime performance relative to other perturbation- and optimization-based methods. However, alternative approaches such as DIX, T-Attr, and GAE run faster than SLOC. Notably, SLOC$_{\text{xp}}$ offers improved runtime efficiency over SLOC, at the cost of slightly reduced performance on the FB benchmark. As explained in Sec. G, SLOC benefits from independent mask generation, enabling parallel processing on a GPU, which reduces computation time. Furthermore, SLOC requires fewer masks than other perturbation-based methods. For instance, RISE [57] reports using between 4,000 and 8,000 masks.

While SLOC's optimization phase is sequential, it avoids model evaluation or backpropagation - a step required by other optimization-based methods such as MP, EP, and LTX. This is since the model's response $r(\mathbf{x}^{\mathbf{m}})$ in Eq. 4 is precomputed and remains fixed throughout the optimization process. This design ensures that the runtime of SLOC's optimization phase is unaffected by the size or complexity of the model.

Finally, it is worth noting that explanations, unlike predictions, are often used for debugging and auditing purposes, where the added computation time for a more accurate and informative explanation is generally regarded as a worthwhile trade-off.

I. Notable Explanation Examples

In this section, we present several case studies of attribution maps generated by SLOC, highlighting its effectiveness across diverse input settings. Additionally, we demonstrate the limitations of optimizing for global vs. local completeness.

Multi-instance settings Figure 11 presents a synthetic image featuring two identical instances of the *’indigo-bunting’* bird, on the left and right halves of the image. This image yields a response of 0.9999 for the *’indigo-bunting’* class for the complete image, by the RN model. When masking out the right half, preserving only the left bird, or masking out the left half preserving only the right half, the model response remains very high: 0.9992 or 0.9955, respectively. Thus, across the three regions: the complete image, the right half, and the left half — no attribution map can fully satisfy local completeness for all corresponding sub-maps. Yet, the SLOC-generated attribution highlights both copies of the bird. This example demonstrates that the soft nature of SLOC enables the generation of high-quality attributions even in cases where local completeness is inherently infeasible.

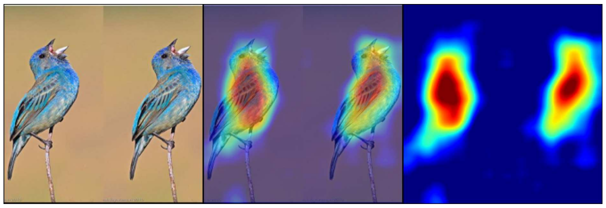

Figure 11. Two copies of the same object are shown (left) along with the corresponding attribution map generated by the SLOC method. The image yields a very high (above 0.99) model response for the *‘indigo-bunting’* class by the ViT-S model. However, when the image is masked to preserve only one of the copies, the response remains very high. Therefore, when considering the three corresponding masks—one preserving the entire image, one preserving the right half, and one preserving the left half—no attribution map can satisfy local completeness for all three corresponding sub-maps. Despite these potential ‘collisions’, the SLOC method produces an attribution map that effectively captures both copies.

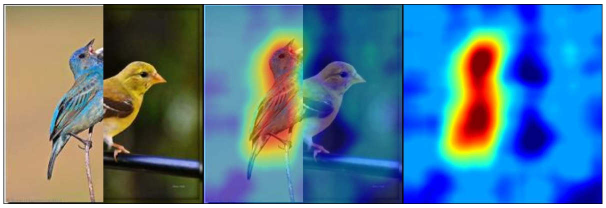

Figure 12. A synthetic multi-class image featuring the classes indigo-bunting and goldfinch. The SLOC attribution map for ViT-S prediction of the goldfinch class shows negative attributions (dark blue) on the indigo-bunting's body, as its presence decreases the predicted probability for the goldfinch class.

Multi-class settings Figure 12 presents a synthetic image created by combining two images (from IN): one of class *‘indigo-bunting’* (left) and one of class *‘goldfinch’* (right). This results in an image composed of two distinct

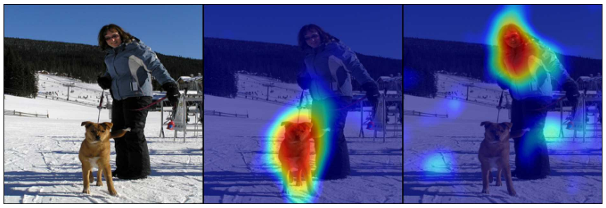

Figure 13. A multi-class image from the VOC dataset with RN SLOC attribution maps for the classes *dog* (center) and *person* (right).

regions, each corresponding to a different class. The figure presents the SLOC attribution for the ViT-S prediction of the *'goldfinch'* class. Notably, the attributions in the dark-blue area containing the indigo-bunting body are negative, indicating that the presence of the indigo-bunting reduces the predicted probability for the *'goldfinch'* class.

Figure 13 shows an image from the VOC dataset, where the top predicted classes by the RN model are *dog* and *person*. The sub-figures present the corresponding SLOC attribution maps for *dog* (center) and *person* (right), providing further empirical evidence of SLOC's ability to highlight the relevant features that contribute to the prediction of each individual class.

Global vs. Local Completeness Optimization Figure 14 compares attribution maps generated for the RN model and the *'jellyfish'* class using SLOC, which minimizes completeness gaps locally over a set of sub-maps (Eq. 4), against those obtained by minimizing the completeness gap **globally** over the entire attribution map. The latter approach, referred to as the global completeness loss, is defined as $(r(\mathbf{x})[y] - \sum_{i=1}^{n} \mathbf{a}_x^y[i])^2$. Subfigures (a) show the attribution map produced by SLOC (top) and its overlay on the input image (bottom). Subfigures (b) present the attribution map obtained by minimizing the global completeness loss (top) and its overlay on the input (bottom). Subfigures (c) display the attribution map generated by minimizing global completeness with an additional total variation (TV) regularization term (top) and its overlay (bottom). Although the attribution maps in (b) and (c) satisfy global completeness, they fail to provide faithful explanations. The attribution map in (b) is dominated by noise, while the one in (c) is influenced by artifacts induced by TV regularization applied to random initialization. This example highlights the limitations of global completeness as a sole optimization objective. In contrast, SLOC, by promoting local completeness, produces faithful attribution maps.

J. Sanity Checks

In order to further evaluate the soundness and validity of SLOC, we conducted both the *parameter randomization* and *data randomization* sanity tests as proposed by [3].

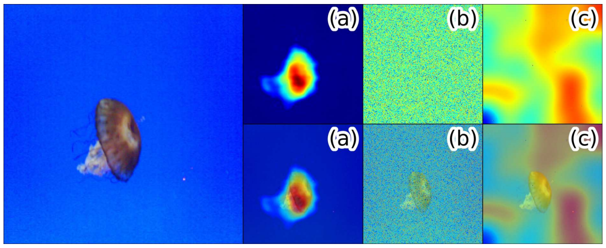

Figure 14. Attribution maps produced for the RN model for class *'jellyfish'*. (a) attribution map produced by SLOC. (b) attribution map obtained by minimizing the **global**-completeness-loss aiming to satisfy global-completeness. (c) attribution map obtained by minimizing global-completeness with an additional TV regularization term. The top row displays the attribution maps, while the bottom row displays the attribution maps overlaid on the image. Although global-completeness is satisfied by the attribution maps in (b) and (c), it is insufficient for producing a faithful explanation.

The experiments utilize the ImageNet ILSVRC 2012 validation set [31] with the VGG-16 [61] model and SLOC.

J.1. Parameter Randomization Test

The parameter randomization test compares the explanation maps produced by the explanation method based on two setups of the same model architecture: (1) trained - the model is trained on the dataset (e.g., a pretrained VGG-16 model that was trained on ImageNet), and (2) random - the same model architecture, with random weights (e.g., a randomly initialized VGG-16 model). For an explanation method that relies on the actual model to be explained, we anticipate significant differences in the explanation maps produced for the trained model and those produced for the random model. Conversely, if the explanation maps are similar, we conclude that the explanation method is insensitive to the model's parameters, and thus may not be useful for explaining the model's prediction. It is worth noting that parameter randomization sanity checks were found inadequate as a criterion for ranking attribution methods, due to the observed performance gap between faithfulness metrics and randomization tests [23]. Nevertheless, we report these results for the sake of completeness.

Given a trained model, we consider two types of parameter randomization tests: The first test randomly re-initializes all weights of the model in a cascading manner (layer after layer). The second test independently randomizes one layer at a time, while keeping all other layers fixed. In both cases, we compare the resulting explanations obtained by using the model with random weights to those derived from the original weights of the model.

J.1.1. Cascading Randomization

The cascading randomization method involves the randomization of a model's weights, starting from the top layer and successively moving down to the bottom layer. This process leads to the randomization of the weights from the top to the bottom layers. Figure 15 presents the Spearman correlation between the original explanation map obtained by SLOC and the original (pretrained) model and the explanation map obtained by SLOC and each of the cascade randomization versions of the original model. The markers on the x-axis are between '0' and '16', where $x = k$ means that the weights of the last k layers of the model are randomized. At $x = 0$ there is no randomization, hence the correlation with the original model is perfect. Starting from $x = 1$ (marked by the horizontal dashed line) and up to $x = 16$, the graph depicts a progressive cascade randomization of the original model. We observe that randomizing the weights starting from the top layer reduces the correlation with the explanation map of the original model to nearly zero. This behavior showcases the sensitivity of SLOC to the model's parameters - an expected and desired property for any explanation method [3].

Figure 16 displays a representative example of explanation maps (bottom) and their overlay to the original image (top), illustrating the cascading randomization process. The first column presents explanation maps produced by SLOC and the original model, while the rest of the columns present explanation maps produced by SLOC and cascading randomized models, where the number i above each column indicates that the explanation map is produced by a model in which the weights of the last i layers were randomized. It is evident that the quality of produced explanation maps significantly degrades as more and more layers are set with random weights.

J.1.2. Independent Randomization

We further consider another version of the model's parameters randomization test, in which a layer-by-layer randomization is employed, one layer at a time. In this test, we aim to isolate the influence of the randomization of each layer, hence randomization is applied to one layer's weights at a time, while all other layers' weights are kept identical to their values in the original model. This randomization methodology enables comprehensive evaluation of the sensitivity of the explanation maps w.r.t. each of the model's layers.

Figure 15 presents results for the independent randomization tests. At $x = 0$ no randomization was applied and the correlation to the original model is perfect. For $x = i$ ($i > 0$) the graph indicates the correlation of the original model with a model in which only the weights of the i-th penultimate layer were randomized while the weights of all other layers were kept untouched. We observe that the correlation values are effectively zero across all layers which indicates SLOC's sensitivity to weight randomization in each layer separately. This property is a desired property for an explanation method, as it indicates the method's sensitivity to each of the model's layers, independently. Finally, Fig. 17 presents a qualitative example in the same fashion as Fig. 16, this time for the independent randomization test. We observe that the quality of all explanation maps produced by a randomized version of the model differs significantly from the original explanation map. We conclude that SLOC successfully passes both types of parameter randomization tests.

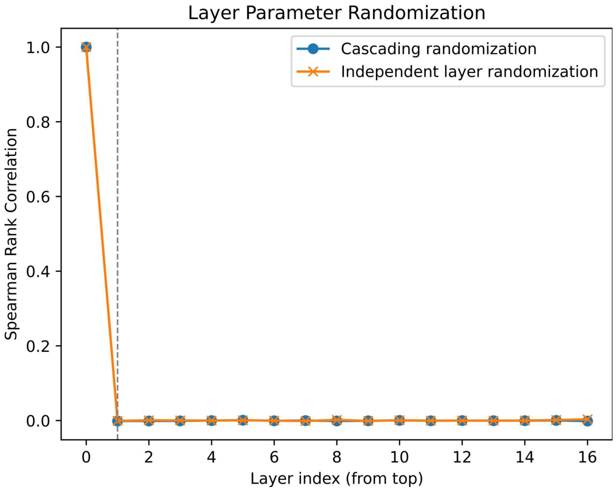

Figure 15. **VGG-16 Layer Parameter Randomization:** This figure illustrates two types of layer randomization types: **Orange (Independent Layer Randomization)** shows the randomization process applied independently to each layer of the model, while the remaining weights are kept fixed at their original values. **Blue (Cascading Randomization)** depicts the sequential randomization of layer weights, starting from the last layer and progressing towards a selected layer. The x-axis represents the layer index, which, for cascading randomization, also corresponds to the number of layers being randomized. The y-axis shows the averaged Spearman rank correlation between the explanation maps produced by SLOC using the original model and the model with randomized weights. The first data point at $x = 0$ corresponds to no randomization (the original model), where the correlation between the explanation maps is 1.0. The dashed line indicates the point where randomization begins. We observe that randomizing even a single layer in either approach reduces the average correlation to nearly zero. This is a desired outcome, confirming that SLOC passes the sanity check. For further details see Secs. J.1.1 and J.1.2.

J.2. Data Randomization Test

The data randomization sanity test is a method used to assess whether an explanation method is sensitive to the labeling of the data used for training the model. This is done by

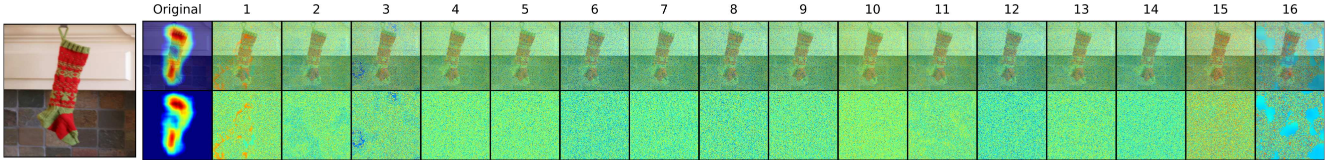

Figure 16. **Cascading Randomization on VGG-16 (ImageNet):** The figure presents the original explanations (first column) for the 'Christmas stocking' class. The progression from left to right illustrates the gradual randomization of network weights up to the layer number indicated at the top of each column, starting from the last layer. The second row displays the resulting saliency maps, while the first row shows the saliency maps overlaid on the original image. We observe that randomizing even just the top layer significantly disrupts the explanation. This behavior is desired, as it demonstrates that SLOC passes the sanity check. For further details, see Sec. J.1.1.

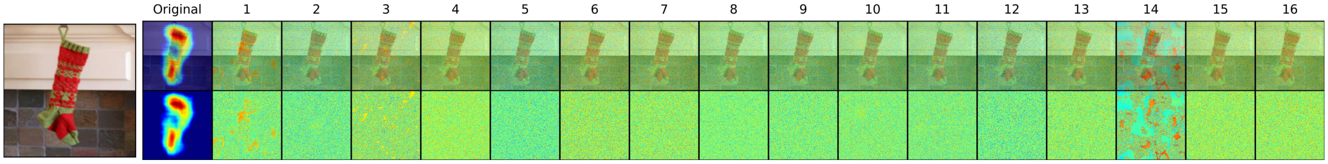

Figure 17. **Independent Randomization on VGG-16 (ImageNet):** Similar to Figure 16, this example randomizes each specific layer independently, while the remaining weights are retained at their original values. We observe that randomizing any single layer significantly disrupts the produced explanation, confirming that SLOC passes the sanity check.

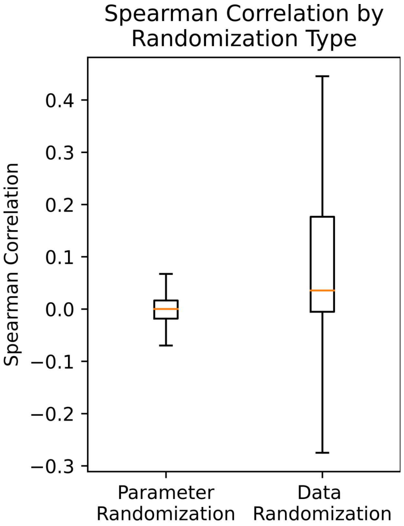

Figure 18. **Parameter and Data Randomization Tests**: Spearman rank correlation box plots for SLOC with the VGG-16 model.

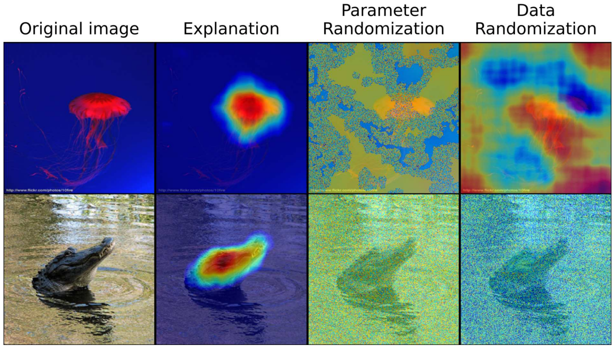

Figure 19. **Sanity Checks:** Rows 1 and 2 present the SLOC results for the parameter randomization and data randomization tests on images of the classes "jellyfish" and "American alligator", respectively. From left to right: the original image, the explanation map produced by SLOC with the trained model, the explanation map produced by SLOC with the untrained model (where the model's weights are randomly initialized without further training), and the explanation map produced by SLOC with a model trained on random labels.

comparing the explanation maps produced by the explanation method for two models with identical architecture that were trained on two different datasets: one with the original labels and another with randomly permuted labels. If the explanation method is sensitive to the labeling of the dataset, we would expect the produced explanation maps to differ significantly between the two cases. However, if the method is insensitive to the permuted labels, it indicates that it does not depend on the relationship between instances and labels that exists in the original data. To conduct the data randomization test, we permute the training labels in the dataset and train the model to achieve a training set accuracy greater than 95%. Note that the resulting model's test accuracy is never better than randomly guessing a label. We then compute explanations on the same test inputs for both the model trained on true labels and the model trained on randomly permuted labels. Figure 18 presents two box plots, one computed for the Spearman correlation values obtained for the parameter randomization test (cascading version), and another for the data randomization test. We can see that the correlation values are very low indicating SLOC's sensitivity to both parameter randomization and data randomization. Specifically, we conclude that SLOC successfully passes the data randomization test.

Finally, Fig. 19 presents additional qualitative examples for both tests. The first row shows two explanation maps produced by SLOC w.r.t. the "jellyfish" class. We see that when SLOC utilizes an ImageNet pretrained VGG-16 model, it produces a focused explanation map (around the cat), but when applying SLOC to the same model with random weights, or to the model trained with random-labels, it fails to detect the jellyfish in the image. The second row shows a similar example for class "American alligator".

K. Motivation Formalization

In this section, we formalize the motivation for SLOC as outlined in 3.1. Notably, the following setting and its assumptions are simplified and do **not** reflect real-world conditions; they serve **solely** to illustrate the underlying motivation. Importantly, SLOC does **not** necessitate these assumptions in practice and its effectiveness is rigorously validated through extensive experiments.

Overview

Our setting involves an image $\mathbf{x}$ with an object of interest that drives the model's prediction of class y. We demonstrate that by enforcing local-completeness across a set of sub-maps, we can derive an attribution map that highlights a compact region containing the object of interest, thereby ensuring faithfulness.

The set of sub-maps is represented by corresponding binary masks $\mathcal{M}$. For a given attribution map $\mathbf{a}_x^y$ and binary mask $\mathbf{m}_i$, their dot product $\mathbf{a}_x^y \cdot \mathbf{m}_i$ yields the sum of attributions in the corresponding sub-map. We define $r(\mathbf{x} \circ \mathbf{m}_i)[y]$ as the model response for class y on the masked image. Local-completeness is satisfied for the sub-map corresponding to $\mathbf{m}_i$ when these two terms are equal.

Our framework operates under the following simplifying assumptions:

- The mask set $\mathcal{M}$ comprises only masks that either completely preserve or completely remove the object of interest.
- When the object is preserved, the model response equals R; otherwise, the response is 0.
- The set of masks includes a mask that preserves the entire image.

An ideal attribution would precisely highlight the object of interest. Consider the intersection region defined by the element-wise product of all masks that preserve the object, along with the complement of all masks that delete the object. This intersection region necessarily contains the object of interest. Furthermore, as the number of sub-maps increases, the salient regions become more precisely defined, potentially converging toward the ideal attribution in our setting.

Now consider an attribution map that assigns a total attribution of R within this intersection area and zero values outside it. This attribution satisfies local-completeness for all sub-masks corresponding to masks in $\mathcal{M}$. For masks preserving the object, the corresponding sub-map contains the intersection, yielding an attribution sum of R. For masks removing the object, the mask and intersection are mutually exclusive, yielding an attribution sum of 0.

Moreover, we prove that any attribution obtained by enforcing local completeness for all corresponding sub-maps and ensuring all attribution values are non-negative can have positive values only within the intersection region. We prove this by contradiction. Assume there exists a pixel outside the intersection region for which the attribution is positive. For such a pixel outside the intersection, one of two cases must hold:

1. There exists a preserving mask that does not include this pixel, or
2. There exists a deleting mask that does include this pixel.

(Otherwise that pixel is in the intersection region).

In the first case, consider the mask that preserves the entire image. The attribution sum for this mask equals R by local-completeness. However, the attribution map contains both the contributions from the sub-map corresponding to the preserving mask (which sums to R) and the positive attribution of the pixel outside the intersection. This would make the total attribution sum strictly greater than R, violating local-completeness for the full-image mask — a contradiction.

In the second case, for a deleting mask that includes the pixel with positive attribution, the sum of attributions within this mask would be greater than zero. However, since this is a deleting mask, local-completeness requires the attribution sum to be zero—another contradiction.

Therefore, a non-negative attribution map satisfying local-completeness must assign zero attribution to all pixels outside the intersection region, with a total sum of R inside this region.

It's worth noting that our formulation imposes no constraints on the distribution of attribution within the intersection region. In the absence of additional information, distributing attribution uniformly is a reasonable choice. This insight motivates the incorporation of Total Variation Loss in the SLOC optimization framework, which naturally promotes spatial coherence in the attribution map.

Lemma:

Let $x \in \mathbb{R}^n$ be an image, and let p be a set of pixels that constitute an object of interest in the image, belonging to class y. Let $r(\cdot)[y]$ denote the model response for class y, and let $\mathbf{m}_p \in \{0,1\}^n$ denote a binary mask that defines the set of pixels p. Let $\mathcal{M} \subset \{0,1\}^n$ be a set of N binary masks such that:

1. $\mathcal{M} = \mathcal{M}_s \cup \mathcal{M}_d$
2. $\mathbf{m}_i \circ \mathbf{m}_p = \mathbf{m}_p \wedge r(\mathbf{x} \circ \mathbf{m}_i)[y] = R, \quad \forall \mathbf{m}_i \in \mathcal{M}_s; \quad \mathcal{M}_s = \{\mathbf{m}_1, ..., \mathbf{m}_k\}$ - masks that completely preserve p.
3. $\mathbf{m}_j \circ \mathbf{m}_p = \mathbf{0}_n \wedge r(\mathbf{x} \circ \mathbf{m}_j)[y] = 0, \quad \forall \mathbf{m}_j \in \mathcal{M}_d; \quad \mathcal{M}_d = \{\mathbf{m}_{k+1}, ..., \mathbf{m}_N\}$ - masks that completely delete p.
4. $\mathbf{m}_1 = \mathbf{1}_n \in \mathcal{M}_s$ - We denote $\mathbf{m}_1$ as the mask that preserves the entire image.

where $\circ$ denotes the element-wise product, and $\mathbf{0}_n$ and $\mathbf{1}_n$ represent vectors of length n with all elements equal to 0 and 1, respectively.

Let $\mathbf{a}_x^y$ be an attribution map such that:

$$\mathbf{a}_x^y \cdot \mathbf{m}_I = R, \quad \mathbf{a}_x^y \circ (1 - \mathbf{m}_I) = \mathbf{0}_n \tag{10}$$

where:

$$\mathbf{m}_I = \mathbf{m}_1 \circ \cdots \circ \mathbf{m}_k \circ (1 - \mathbf{m}_{k+1}) \circ \cdots \circ (1 - \mathbf{m}_N) \tag{11}$$

$\mathbf{m}_I$ represents the intersection of the masks that preserve p with the inversion of the masks that delete p. In other words, the total attribution within $\mathbf{m}_I$ sums to R, with all attributions outside $\mathbf{m}_I$ being zero.

Then:

1. **Completeness is satisfied for all sub-maps induced by $\mathcal{M}$:**

$$\mathbf{a}_x^y \cdot \mathbf{m}_j = r(\mathbf{x} \circ \mathbf{m}_j)[y], \quad \forall \mathbf{m}_j \in \mathcal{M}$$

2. **Uniqueness of $\mathbf{m}_I$ as the maximal subregion with nonzero attribution**: Consider any non-negative attribution $\mathbf{b}_x^y$ that satisfies completeness for all sub-maps (i.e., $\mathbf{b}_x^y \cdot \mathbf{m}_j = r(\mathbf{x} \circ \mathbf{m}_j)[y], \ \forall \mathbf{m}_j \in \mathcal{M}$ and $\mathbf{b}_x^y \geq \mathbf{0}_n$). Then we have:

$$(1 - \mathbf{m}_I) \circ \mathbf{b}_x^y = \mathbf{0}_n$$

This implies that $\mathbf{b}_x^y$ has non-zero attribution **only within the region defined by $\mathbf{m}_I$.**

Proof:

Part 1: Completeness for all sub-maps

For masks $\mathbf{m}_i \in \mathcal{M}_s$:

$$\begin{aligned}
\mathbf{a}_x^y \cdot \mathbf{m}_i &= \sum_{u=1}^{n} \mathbf{a}_x^y[u] \cdot \mathbf{m}_i[u] && (12)\\
&= \sum_{\mathbf{m}_I[u]=1} \mathbf{a}_x^y[u] \cdot \mathbf{m}_i[u] + \sum_{\mathbf{m}_I[u]=0} \mathbf{a}_x^y[u] \cdot \mathbf{m}_i[u] && (13)\\
&= \sum_{\mathbf{m}_I[u]=1} \mathbf{a}_x^y[u] \cdot \mathbf{m}_i[u] + \sum_{\mathbf{m}_I[u]=0} 0 \cdot \mathbf{m}_i[u] && (14)\\
&= \sum_{\mathbf{m}_I[u]=1} \mathbf{a}_x^y[u] \cdot \mathbf{m}_i[u] && (15)\\
&= \sum_{\mathbf{m}_I[u]=1} \mathbf{a}_x^y[u] \cdot \mathbf{m}_I[u] && (16)\\
&= \mathbf{a}_x^y \cdot \mathbf{m}_I = && (17)\\
&= R = r(\mathbf{x} \circ \mathbf{m}_i)[y] && (18)
\end{aligned}$$

Transition (14) follows from (10). Since $\mathbf{a}_x^y[u] \cdot (1 - \mathbf{m}_I[u]) = 0$ for all $u \in 1, \ldots, n$, it follows that if $\mathbf{m}_I[u] = 0$, then $\mathbf{a}_x^y[u] = 0$. Transition (16) follows from the definition of $\mathbf{m}_I$, if $\mathbf{m}_I[u] = 1$, then for any $\mathbf{m}_i \in \mathcal{M}_s$, we have $\mathbf{m}_i[u] = 1$ as well, hence $\mathbf{m}_I[u] = \mathbf{m}_i[u]$. Transition (18) follows directly from (10).

For masks $\mathbf{m}_j \in \mathcal{M}_d$:

$$\begin{aligned}
\mathbf{a}_x^y \cdot \mathbf{m}_j &= \sum_{u=1}^{n} \mathbf{a}_x^y[u] \cdot \mathbf{m}_j[u] && (19)\\
&= \sum_{\mathbf{m}_I[u]=1} \mathbf{a}_x^y[u] \cdot \mathbf{m}_j[u] + \sum_{\mathbf{m}_I[u]=0} \mathbf{a}_x^y[u] \cdot \mathbf{m}_j[u] && (20)\\
&= \sum_{\mathbf{m}_I[u]=1} \mathbf{a}_x^y[u] \cdot 0 + \sum_{\mathbf{m}_I[u]=0} \mathbf{a}_x^y[u] \cdot \mathbf{m}_j[u] && (21)\\
&= \sum_{\mathbf{m}_I[u]=1} \mathbf{a}_x^y[u] \cdot 0 + \sum_{\mathbf{m}_I[u]=0} 0 \cdot \mathbf{m}_j[u] && (22)\\
&= 0 = r(\mathbf{x} \circ \mathbf{m}_j)[y] && (23)
\end{aligned}$$

Transition (21) follows from the definition of $\mathbf{m}_I$ in (11), which implies that if $\mathbf{m}_I[u] = 1$, then $\mathbf{m}_j[u] = 0$. Similarly, transition (22) follows from (10): since $\mathbf{a}_x^y[u] \cdot (1 - \mathbf{m}_I[u]) = 0$, it follows that if $\mathbf{m}_I[u] = 0$, then $\mathbf{a}_x^y[u] = 0$.

Part 2: Nonzero attribution only within the intersection

Given that:

$$\mathbf{b}_x^y \geq \mathbf{0}_n \tag{24}$$

and

$$\mathbf{b}_x^y \cdot \mathbf{m}_j = r(\mathbf{x} \circ \mathbf{m}_j)[y], \ \forall \mathbf{m}_j \in \mathcal{M} \tag{25}$$

we proceed by contradiction.

Suppose there exists $v \in \{1, \ldots, n\}$ such that:

$$\mathbf{m}_I[v] = 0 \wedge \mathbf{b}_x^y[v] > 0 \tag{26}$$

By the definition of $\mathbf{m}_I$ (11), there are two possible cases:
1. There exists a mask $\mathbf{m}_q \in \mathcal{M}_s$ such that $\mathbf{m}_q[v] = 0$, or
2. There exists a mask $\mathbf{m}_q \in \mathcal{M}_d$ such that $\mathbf{m}_q[v] = 1$.

Case 1: If there exists a mask $\mathbf{m}_q \in \mathcal{M}_s$ such that $\mathbf{m}_q[v] = 0$, then:

$$\begin{aligned}
\mathbf{b}_x^y \cdot \mathbf{m}_1 &= \sum_{u=1}^{n} \mathbf{b}_x^y[u] && (27)\\
&= \sum_{\mathbf{m}_q[u]=1} \mathbf{b}_x^y[u] + \sum_{u=v} \mathbf{b}_x^y[u] + \sum_{u \neq v \wedge \mathbf{m}_q[u]=0} \mathbf{b}_x^y[u] && (28)\\
&\geq \sum_{\mathbf{m}_q[u]=1} \mathbf{b}_x^y[u] + \sum_{u=v} \mathbf{b}_x^y[u] && (29)\\
&= \mathbf{b}_x^y \cdot \mathbf{m}_q + \mathbf{b}_x^y[v] && (30)\\
&= r(\mathbf{x} \circ \mathbf{m}_q)[y] + \mathbf{b}_x^y[v] + && (31)\\
&= R + \mathbf{b}_x^y[v] && (32)\\
&> R && (33)\\
&= r(\mathbf{x} \circ \mathbf{m}_1)[y] = \mathbf{b}_x^y \cdot \mathbf{m}_1 && (34)
\end{aligned}$$

which is a **contradiction**. Transition (29) follows directly from the nonnegativity of $\mathbf{b}_x^y$ (24). Transition (31) follows from $\mathbf{b}_x^y$ satisfying local completeness (25). Transition (32) follows since $\mathbf{m}_q \in \mathcal{M}_s$. Transition (33) follows from the positivity of $\mathbf{b}_x^y[v]$ (26)

Case 2: If there exists a mask $\mathbf{m}_q \in \mathcal{M}_d$ such that $\mathbf{m}_q[v] = 1$, then:

$$\begin{aligned}
\mathbf{b}_x^y \cdot \mathbf{m}_q &= \sum_{u=1}^{n} \mathbf{b}_x^y[u] \cdot \mathbf{m}_q[u] && (35)\\
&\geq \mathbf{b}_x^y[v] \cdot \mathbf{m}_q[v] && (36)\\
&= \mathbf{b}_x^y[v] \cdot 1 && (37)\\
&> 0 && (38)\\
&= r(\mathbf{x} \circ \mathbf{m}_q)[y] = \mathbf{b}_x^y \cdot \mathbf{m}_q && (39)
\end{aligned}$$

which is a **contradiction** as well. Transition (36) follows from the nonnegativity of $\mathbf{b}_x^y$ (24). Transition (38) follows from the positivity of $\mathbf{b}_x^y[v]$ (26). □

L. Additional Axioms

We have demonstrated how SLOC harnesses the concept of completeness locally as a guiding principle to produce high-quality explanations. We now extend this discussion to include additional axioms relevant to attributions and attribution methods. By axioms, we refer to the desired properties of either the method or the explanation.

Sundararajan et al. [65] discuss the following additional axioms: **Sensitivity**, **Implementation Invariance**, and **Linearity**. Erion et al. [36] extend this discussion by introducing the **Smoothness** and **Sparsity** as additional axiomatic attribution priors.

Two definitions are provided for Sensitivity: (a) For every input and baseline that differ in one feature but have different predictions, the differing feature must receive a non-zero attribution. (b) If the model output does not depend on an input, the attribution for that input must always be zero.

The Implementation Invariance axiom is satisfied when attributions remain identical for functionally equivalent networks. Two networks are considered functionally equivalent if their outputs are identical for all inputs.

The Linearity axiom is satisfied if, when two models are composed linearly to form a third model,

$$f_3(\mathbf{x}) := a \cdot f_1(\mathbf{x}) + b \cdot f_2(\mathbf{x}),$$

the attribution of the composed model is the weighted sum of the attributions for f_1 and f_2, with weights a and b, respectively.

The Smoothness and Sparsity priors are not binary traits that are either satisfied or violated but rather quantitative attributes for which loss terms can be defined. This approach aligns with our definition of the *completeness-gap* (Eq. 2) as a soft measure of completeness.

As a black-box method, SLOC naturally satisfies the Implementation Invariance axiom. It relies solely on the model's responses to perturbed images and is entirely agnostic to the model's internal implementation. While the attributions depend on the randomly selected masks, these masks can remain fixed and reused across runs.

Regarding the Smoothness and Sparsity priors, SLOC incorporates L1 and TV regularization terms into its loss function (Eq. 4), alongside the completeness-gap term which results in smoother and sparser explanations accordingly. When choosing between explanations with similarly low completeness gaps, we prioritize those that are smooth and sparse.

SLOC does not inherently guarantee satisfaction of the Sensitivity or Linearity axioms. Addressing these limitations to further enhance the quality of explanations is left for future research.

M. Limitations and Future Work

SLOC relies on the generation of random masks and perturbations to produce high-quality explanations for image classification models. While effective, this approach has potential for improvement through more sophisticated mask-generation techniques that could further enhance the accuracy and robustness of the explanations. We outline several promising directions for future research to explore and build upon the current strengths of SLOC.

One area for improvement is the development of more advanced methods for mask selection. By incorporating information from previous perturbations and their corresponding model responses, it may be possible to guide the generation of subsequent masks in a more informed and targeted manner. However, this introduces a trade-off: leveraging such information could make the mask generation process more sequential, thereby reducing its parallelizability. Future research could explore this trade-off and investigate strategies to balance informed mask generation with computational efficiency.

Another promising direction involves the use of progressive multi-resolution perturbations—starting with coarse masks composed of large patches (low resolution) and gradually refining the analysis by decreasing patch size in regions identified as influential. This hierarchical approach may lead to more accurate and interpretable explanations.

In parallel, incorporating image-specific features such as segmentation-based masks represents an alternative approach. Instead of relying on randomly positioned patches, masks could be constructed using more meaningful segments derived from a segmentation algorithm, potentially aligning the perturbations more closely with the structure of the input image and improving the interpretability of the resulting attributions.

Another area that worth further investigation pertains to the SLOC optimization phase. In this phase (Sec. 3.2), SLOC evaluates the completeness-gap across all masks at every optimization step using gradient descent. Future work could explore the use of stochastic gradient descent, where a subset of masks is considered at each step. Introducing stochasticity in the optimization phase might improve convergence, while still allow for the generation of masks and responses in advance, enabling parallelization to be leveraged.

By exploring these avenues, SLOC can be further refined to produce even more accurate, reliable, and robust explanations.

Last but not least, as part of our future research we plan to investigate the applicability of SLOC to additional domains, such as natural language processing, audio models, and recommender systems. Yet, applying the concept behind SLOC to new domains would require necessary adaptations to support different types of input representations.