

---

# Scaling Symbolic Methods using Gradients for Neural Model Explanation

---

**Subham S. Sahoo   Subhashini Venugopalan   Li Li   Rishabh Singh   Patrick Riley**  
 Google Research  
`{subhamsahoo,vsubhashini,leeley,rising,pfr}@google.com`

## Abstract

Symbolic techniques based on Satisfiability Modulo Theory (SMT) solvers have been proposed for analyzing and verifying neural network properties, but their usage has been fairly limited owing to their poor scalability with larger networks. In this work, we propose a technique for combining gradient-based methods with symbolic techniques to scale such analyses and demonstrate its application for model explanation. In particular, we apply this technique to identify minimal regions in an input that are most relevant for a neural network’s prediction. Our approach uses gradient information (based on Integrated Gradients [23]) to focus on a subset of neurons in the first layer, which allows our technique to scale to large networks. The corresponding SMT constraints encode the minimal input mask discovery problem such that after masking the input, the activations of the selected neurons are still above a threshold. After solving for the minimal masks, our approach scores the mask regions to generate a relative ordering of the features within the mask. This produces a saliency map which explains “where a model is looking” when making a prediction. We evaluate our technique on three datasets - MNIST, ImageNet, and Beer Reviews, and demonstrate both quantitatively and qualitatively that the regions generated by our approach are sparser and achieve higher saliency scores compared to the gradient-based methods alone.

## 1 Introduction

Satisfiability Modulo Theory (SMT) solvers [3] are routinely used for symbolic modeling and verifying correctness of software programs [21], and more recently they have also been used for verifying properties of deep neural networks [14]. In this work, we present a new approach to use SMT solvers for explaining neural network decisions.

We consider the problem of model explanation as one of identifying a minimal set of features in a given input that is critical to a model’s prediction [5, 16]. With this definition, SMT solvers could present an interesting approach to model explanation as the search space for identifying such a minimal feature set is exponential in the number of input features. We can encode a neural network using real arithmetic [14] and use an SMT solver to optimize over the constraints to identify a minimal set of inputs that can explain the prediction. However, this is challenging for SMT solvers as the decision procedures for solving these constraints have exponential complexity, and is further exacerbated by the large number (millions) of parameters in typical neural network models. Thus, previous approaches for SMT-based analysis of neural networks have been quite limited, and have only been able to scale to networks with few thousands of parameters.

To address this issue of scalability, instead of doing minimization by encoding the entire network, our approach takes advantage of the gradient information, specifically, Integrated Gradients (IG) [23] to identify and encode a much simpler set of linear constraints pertaining to the first hidden layer. We encode the mathematical equations of a neural network as SMT constraints using the theory of Linear

Real Arithmetic (LRA), and use z3 solver [4] as it additionally supports optimization constraints such as minimization. The SMT solver then finds a minimal subset by performing minimization on these equations. Thus, our approach, which we refer to as SMUG, is able to scale **Symbolic Methods Using Gradient** information while still providing a faithful explanation of the neural network’s decision.

SMUG relies on two key assumptions. First, that the activations of a relatively small subset of neurons in the first layer is sufficient for approximately representing the final response of the network. Second, that an input which produces activations of those neurons which are at least as large as a fraction of their current activations is a good proxy for explaining what is important to the model. The quantitative and qualitative evaluations in this work support these assumptions (notably for feedforward and convolutional networks with ReLU activations), and its applicability to other model architectures and activations is left for future work.

We evaluate SMUG on three datasets: MNIST [15], ImageNet [7], and Beer Reviews [17]. We show that we can fully encode the minimal feature identification problem for a small feedforward network (without gradient-based neuron selection) for MNIST, but we observe that the SMT solver scales poorly for even intermediate sized networks. On ImageNet, we observe that our method performs better than Integrated Gradients [23] and several strong baselines. Additionally, we observe that our approach finds significantly sparser masks (on average 17% of the original image size). Finally, we also show that our technique is applicable to text models where it performs competitively with other methods including SIS [5] and Integrated Gradients [23].

This paper makes the following key contributions:

- We present a technique (SMUG) to encode the minimal input feature discovery problem for neural model explanation using SMT solvers.
- Our approach, uses gradient information to scale SMT-based analysis of neural networks to larger models and input features. Further, this approach overcomes the issue of choosing a “baseline” parameter for Integrated Gradients [13, 22].
- We empirically evaluate SMUG on image and text datasets, and show that the minimal features identified by it are both quantitatively and qualitatively better than several baselines.

## 2 Related Work

SMT based symbolic techniques have been used for verifying neural network properties [12, 14]. Reluplex [14] extends the simplex method to handle ReLU functions by leveraging its piecewise linear property and presents an iterative procedure for gradual satisfaction of the constraints. [12] proposes a layer-wise analysis using a refinement-based approach with SMT solvers for verifying the absence of adversarial input perturbations. [26] present a linear programming (LP) formulation again using the piecewise linear property of ReLU to find minimal changes to an input to change a network’s classification decisions. [9] uses Reluplex to learn input properties in the form of convex predicates over neuron activations, which in turn capture different behaviors of a neural network. While these techniques have shown promising results, scaling these approaches for larger neural networks and performing richer analysis based on global input features still remains a challenge.

While the above SMT based techniques focus on verifying properties of deep networks, our work focuses on applying symbolic techniques to the related task of model explanation, i.e. to say where a model is “looking”, by solving for the input features responsible for a model’s prediction. Some explanation techniques are model agnostic (i.e., black-box) while others are back-propagation based. Model agnostic (black-box) explanation techniques such as SIS, LIME [1, 5] have a similar formulation of the problem as ours in the sense that they perturb the input pixels by masking them and optimize to identify minimal regions affecting the performance of the model. This formulation can lead to evaluating the model on out of distribution samples [11] with potential for adversarial attacks [19]. In contrast, back-propagation based methods [2, 23, 18] examine the gradients of the model with respect to an input instance to determine pixel attribution. Our work builds on the IG method [23]. IG integrates gradients along the “intensity” path where the input (image or text embedding) is scaled from an information-less baseline (all zeros input, e.g., all black or random noise image) to a specific instance. This helps the model determine attribution at the pixel level. In our work, we use IG to determine important nodes in the first layer (closest to the input). The key improvement of our technique over IG is that, by using IG only on the first layer and then using SMT

solver based solution to determine saliency on the input, we not only preserve faithfulness, but also overcome the issue of choosing an appropriate baseline image for IG [13, 22, 25].

### 3 Method: Scaling Symbolic encoding Using Gradient (SMUG) information

We now describe our approach SMUG, which combines attribution based on gradient information with an SMT-based encoding of the minimal input feature identification problem. For text applications, the mask can be directly applied to individual words; for images, we show how to generate saliency maps from the predicted mask in order to determine pixelwise attribution.

#### 3.1 Symbolic Encoding of Neural Networks

SMT solvers can be used to encode the semantics of a neural network [14]. In particular, given a fully connected neural network with  $n$  hidden layers, weights  $W = \{W_1, W_2, \dots, W_n\}$ , biases  $B = \{b_1, b_2, \dots, b_n\}$ , activation function  $\phi$ , and final layer softmax  $\sigma$ , we can use the SMT theory of nonlinear real arithmetic to obtain a symbolic encoding of the network. Let  $X \in \mathbb{R}^{m \times n}$  denote an input image with  $m \times n$  pixels,  $M \in \{0, 1\}^{m \times n}$  an unknown binary mask,  $L_i$  the output (i.e., activations) of the  $i^{\text{th}}$  layer ( $L_0 = X$  is the input) and  $\alpha(p_j)$  the output of  $j^{\text{th}}$  logit in the final layer:

$$L_i \equiv \phi(W_i L_{i-1} + b_i) \quad \alpha(p_j, W, B, X) \equiv \sigma_j(W_n L_{n-1} + b_n)$$

Given this symbolic encoding, we can encode the minimal input mask generation problem as:

$$\exists M : \text{minimize}(\sum_{ij} M_{ij}) \wedge \alpha(p_{\text{label}}, W, B, M \odot X) > \alpha(p_l, W, B, M \odot X) \quad \forall l \neq \text{label} \quad (1)$$

where  $p_{\text{label}}$  and  $p_l$  refer to the logits corresponding to the true label and the other labels respectively. The number of constraints grow with increasing network size and SMT decision procedure for Non-linear Real Arithmetic is doubly exponential. Even for piecewise ReLU networks, SMT decision procedures for Linear Real Arithmetic combine simplex-based methods (exponential complexity) with other decision procedures such as Boolean logic (NP-complete complexity), which causes the solving times to grow dramatically with network size. When we apply this encoding even for a small feed-forward network on MNIST dataset, the SMT solver does not scale well (Section 4). This motivates our proposed approach for using gradient information to simplify the SMT constraints.

#### 3.2 Scaling Symbolic Encoding for Model Explanation

We present two key ideas to scale up the symbolic encoding to analyze large neural networks, especially in the context of the minimal input mask discovery problem.

First, we score neurons in the first layer of the network that are important to the prediction. The primary aim of this step is to encode just the first layer of the network to bypass modeling of complex non-linear activation functions in the subsequent layers. To this end, we use Integrated Gradients (IG) [23] by treating the first layer activations ( $L_1$ ) as an input to the subsequent network. This assigns an attribution score to each hidden node where a node with a positive score is relevant to the prediction and a node with a negative / zero score is considered irrelevant. More specifically, suppose  $F : \mathbb{R}^{a \times b \times c} \rightarrow [0, 1]^d$  represents a deep network and  $x$  represents the input. Integrated gradients are obtained by accumulating the gradients at all points along the straightline path from an “information-less” baseline  $x'$  to the input  $x$ . The information-less baseline in this case is an all zeros tensor. The path can be parameterized as  $g(\alpha) = x' + \alpha \cdot (x - x')$ . IG is then given by Eq. 2.

$$IG(x) = (x - x') \int_{\alpha=0}^1 \frac{\partial F(g(\alpha))}{\partial g(\alpha)} d\alpha \quad (2)$$

The second key idea of our approach is that we only consider activations with the highest positive attributions. Empirically, we still observe scaling issues with SMT when considering all first layer neurons ( $L_1$ ), so a method for picking a subset of neurons is important for practical application.

With these two ideas, we formalize the problem as follows. Let  $X \in \mathbb{R}^{m \times n}$  denote an input image with  $m \times n$  pixels, and  $M \in \{0, 1\}^{m \times n}$  denote an unknown binary mask. Let  $N_\theta : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^k$  be a neural network that maps input images to  $k$  hidden neuron (also referred to as *important neurons*),

where  $\theta$  denotes the learnt weight parameters.  $D^k$  represents the set of  $k$  important nodes with the highest attributions in  $IG(N_\theta(X))$ . The goal now is to learn a mask  $M$  such that:

$$\min\left(\sum_{ij} M_{ij}\right) : N_\theta(M \odot X)_t > \gamma \cdot N_\theta(X)_t \quad \forall t \in D^k \quad (3)$$

where  $\gamma$  is a parameter which regulates how “active” the neurons remain after masking.

### 3.3 SMT formulation of Minimal Input Mask Discovery Problem

Given  $D^k$ , a set of  $k$  neurons with highest positive attributions in  $IG(N_\theta(x))$ , our goal is to find a minimal mask such that the activations of these neurons are above some threshold ( $\gamma$ ) times their original activation values. Eq. 4 shows the constraints for a minimal mask. The first set of constraint specifies that the unknown mask variable  $M$  can only have 0 and 1 as possible entries in the matrix. The second and third set of constraints encode the activation values of the first layer of network with corresponding masked and original inputs respectively. The fourth set of constraint states that the activations of these  $k$  neurons should be at least  $\gamma$  times the original activation values, and the final constraint adds the optimization constraint to minimize the sum of all the mask bits. Note that here we show the formulation for a feedforward network and a input  $X$  with 2 channels, but it can be extended to convolutional networks and an input with 3<sup>rd</sup> channel as well in a straightforward manner where the mask variables across the same channel share the same mask variable.

$$\begin{aligned} \exists M : \bigwedge_{1 \leq i \leq m, 1 \leq j \leq n} (M_{ij} == 0) \vee (M_{ij} == 1) \bigwedge_{\forall i \in D^k} o_i^m = (W_1(X \odot M) + b_1)_i \\ \bigwedge_{\forall i \in D^k} o_i = (W_1 X + b_1)_i \bigwedge_{\forall i \in D^k} o_i^m > \gamma \cdot o_i \bigwedge \text{minimize}(\Sigma_{ij} M_{ij}) \end{aligned} \quad (4)$$

### 3.4 Constructing Saliency Map from Binary Mask

The SMT solver generates a minimal binary input mask by solving the constraints shown in Eq. 4. We further use the attribution scores obtained from IG to assign importance scores to each mask pixel. A mask variable  $M_{ij}$  that is assigned a value of 1 by SMT is assigned a score  $s_{ij}$  computed as:

$$s_{ij} = \sum_{1 \leq p \leq k} \alpha(o_p) \mathbb{1}_{\text{receptive}(o_p)}(x_{ij}) \quad \forall i, j : M_{ij} = 1 \quad (5)$$

where  $\alpha(o_p)$  denotes the attribution score assigned by IG for neuron  $o_p$  and the indicator function denotes that pixel  $x_{ij}$  is present in the receptive field of  $o_p$ , i.e. it is present in the linear SMT equation used to compute  $o_n$ . These scores are then used to compute a continuous saliency map for an input. Finally, to amplify the pixel differences for visualization purposes in gray scale, we scale the non-zero score values between 0.5 and 1.

## 4 Experimental Setup

**Datasets.** We empirically evaluate SMUG on two image datasets, MNIST [15], and ImageNet [7], as well as a text dataset of Beer Reviews from [17].

**MNIST:** We use the MNIST dataset to show the scalability of the full network encoding in SMT (presented in Section 3.1). We use a feedforward model consisting of one hidden layer with 32 nodes (ReLU activation) and 10 output nodes with sigmoid, one each for 10 digits (0 - 9). For 100 images chosen randomly from the validation set, the SMT solver could solve the constraint shown in Eq. 1 (returns SAT) for only 41 of the images. For the remaining 59 images, the solver returns UNKNOWN, which means the given set of constraints was too difficult for the solver to solve.

**ImageNet:** We use 3304 images ( $224 \times 224$ ) with ground truth bounding boxes from the validation set of ImageNet. The images were selected by filtering for those where the model classification was correct and the image had a ground truth bounding box annotation for the object class. We use the Inception-v1 model from [24] which classifies images into one of the 1000 ImageNet classes.

**Beer Reviews:** To evaluate SMUG on a textual task we consider the review rating prediction task on the Beer Reviews dataset consisting of 70k training examples, 3k validation and 7k test examples. Additionally, the dataset comes with ground truth annotations where humans provide the rationale (select words) that correspond to the rating and review. We train a 1D CNN model to predict the rating for the *aroma* of the beer on a scale from 0 to 1. This model is identical to the model used in [5] and consists of a convolution layer with 128 kernels followed by a ReLU, a fully connected layer, and a sigmoid. It achieves a validation mean square error of 0.032.

**Metrics.** Assessing the quality of the saliency maps, especially binary masks, is challenging. The change in confidence of the classifier (between the original and masked) image alone may not be a reliable measure since the masked input could fall out of the training distribution [11]. Instead, we use the metric proposed in [6] shown in Eq. 6. This metric, which we term Log Sparsity Confidence difference (**LSC**) score, first finds the tightest bounding box that captures the entire mask, then computes confidence on the cropped box resized to the original image size (we use bilinear interpolation). This not only helps keep images closer to the training distribution, but also helps evaluate explanations without the need for groundtruth annotations. The LSC score is computed as:

$$LSC(a, c) = \log(\tilde{a}) - \log(c), \quad \tilde{a} = \max(0.05, a) \quad (6)$$

where  $a$  is the fractional area of the rectangular cropped image and  $c$  is the confidence of the classifier for the true label on the cropped image. A saliency map that is compact and allows the model to still recognize the object would result in a lower LSC score. LSC captures model confidence as well as compactness of the identified salient regions, both of which are desirable when evaluating an explanation. The compactness in particular also makes it suitable for evaluating SMT based methods for the effect of minimization. We adapt LSC to also assess continuous valued saliency maps by, 1) setting a threshold on the continuous valued saliency map to convert them to a binary mask and 2) iterating over multiple thresholds (in steps) to identify the one that results in the best LSC score. We also report the fraction of images for which the mask generated by a given method is better (i.e. produces an equal or lower LSC score) than other methods, which we refer to as **Win%**.

**Comparison methods.** The final saliency mask for SMUG comes from Eq. 5 (Sec. 3.4). We compare this to the saliency maps, and bounding boxes from several baselines described below.

**SMUG<sub>base</sub>** is a variant of SMUG that does not perform SMT-based minimization. Here, in Eq. 3, we simply set  $M_{ij} = 1$  for each pixel  $x_{ij}$  that is in the receptive fields of the top- $k$  neurons in the first layer ( $L_1$ ) selected by IG. We note here that in case of both SMUG and SMUG<sub>base</sub> in the formulations in Eq. 3 and 4, we set  $k = 3000$ ,  $\gamma = 0$  for ImageNet, and  $k = 100$ ,  $\gamma = 0$  for text experiments (this choice is explored more in the supplementary material).

**IG** corresponds to Integrated Gradients [23] with the black image as a baseline.

**SIS** refers to Sufficient Input Subset [5], which finds multiple disjoint subsets of input features (in decreasing order of relevance) which individually allow a confident classification. We compare against SIS only on the text dataset as it did not scale for ImageNet because of memory requirements.

**GROUNDTRUTH** corresponds to the baseline that uses human annotated bounding box, which capture the object corresponding to the image label.

**MAXBOX** denotes maximal mask spanning the entire image.

**CENTERBOX** uses a bounding box placed at the center of the image covering half of the image area.

**OPTBOX** refers to a bounding box that approximately optimizes for LSC. The saliency metric in Eq. 6 relies on finding a single bounding box for an image. To find a box that directly maximizes the metric, we first discretize the image into subgrids of  $10 \times 10$  pixels; and then perform a brute force search by selecting 2 points on the grid (to represent opposite corners of a rectangle) and identify a subgrid with the best score.

## 5 Results

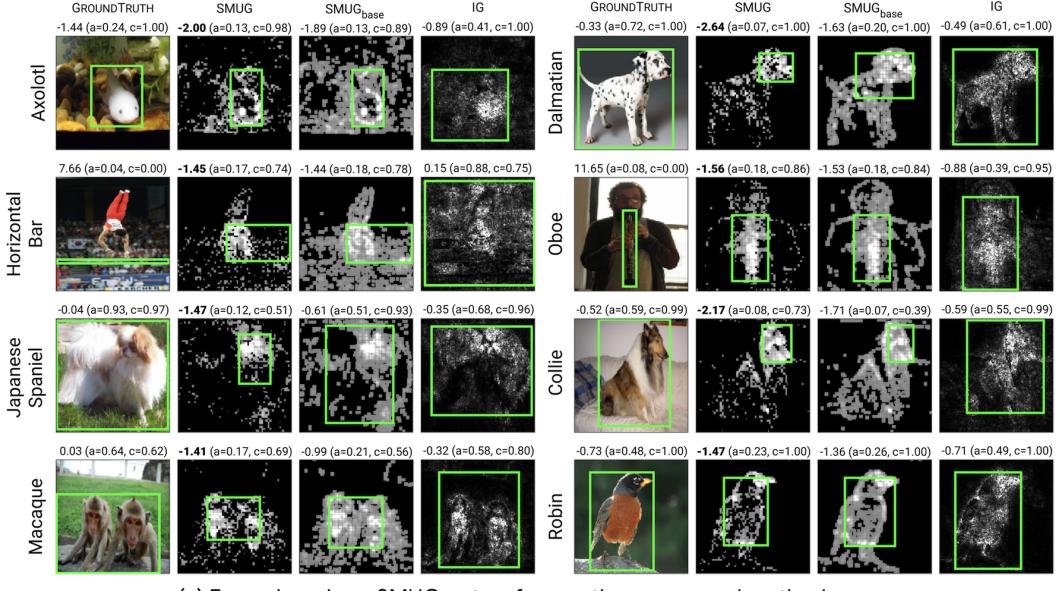
### 5.1 ImageNet

As mentioned previously, when computing masks using SMUG, for ImageNet we set  $k = 3000$  in Eq. 3 and 4. Further, each masking variable  $M_{ij}$  is used to represent a  $4 \times 4$  grid of pixels instead of a single pixel (to reduce running time). Table 1 presents quantitative results reporting the median LSC score and Win% values. Fig. 1 present qualitative examples

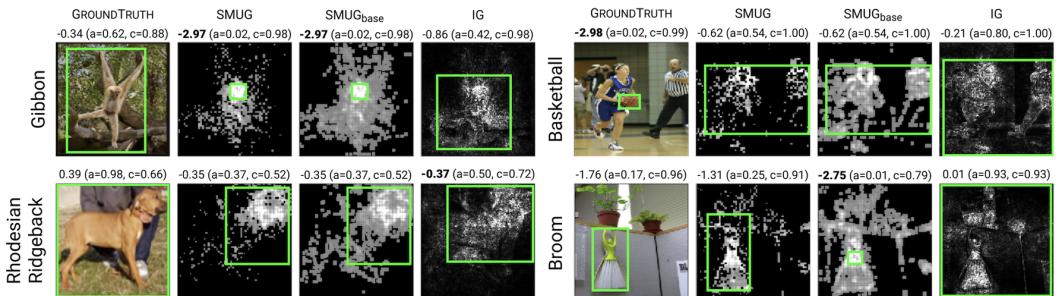
**IG vs SMUG and SMUG<sub>base</sub>.** From Table 1, we observe that SMUG and SMUG<sub>base</sub> achieve a significantly better score ( $-1.26$  and  $-1.23$  resp.) compared to IG ( $-0.34$ ). As observed in some

Table 1: **ImageNet**. We report the median LSC score ( $\downarrow$  lower is better) along with the 75th (top) and 25th percentile (bottom) values, and the mean Win% score in percentage ( $\uparrow$  higher is better) with binomial proportion confidence interval (normal approximation) on 3304 images in the validation set. The Win% values don't sum to 100 due to overlap when methods achieve identical scores.

Method	SMUG	SMUG <sub>base</sub>	GROUNDTRUTH	IG	CENTERBOX	MAXBOX	OPTBOX
<b>LSC <math>\downarrow</math></b>	<b>-1.26<sup>0.75</sup><sub>-1.80</sub></b>	-1.23 <sup>0.71</sup> <sub>-1.76</sub>	-0.34 <sup>0.04</sup> <sub>-0.81</sub>	-0.29 <sup>0.05</sup> <sub>-0.62</sub>	-0.64 <sup>0.29</sup> <sub>-0.69</sub>	0.04 <sup>0.23</sup> <sub>0.00</sub>	-2.27 <sup>1.79</sup> <sub>-2.71</sub>
<b>Win% <math>\uparrow</math></b>	<b>63.9 <math>\pm</math> 1.6</b>	50.2 $\pm$ 1.7	5.7 $\pm$ 0.8	3.6 $\pm$ 0.6	5.9 $\pm$ 0.8	0.3 $\pm$ 0.2	-



(a) Examples where SMUG outperforms other compared methods.



(b) Examples where SMUG performs less favorably.

Figure 1: Examples where SMUG outperforms other compared methods (top 4 rows), and where SMUG performs less favorably (last 2 rows) based on LSC. The green box on the original image highlights the groundtruth box; for the saliency methods it represents the bounding box with the best LSC score. Numbers on top denote the LSC score, the fractional area of the bounding box ( $a$ ), and the confidence of the classifier ( $c$ ) on the cropped region.

qualitative examples (Fig. 1), SMUG tends to assign high scores to a much more localized set of pixels whereas IG distributes high scores more widely (spatially). As LSC metric favors compactness, which is desirable for human interpretability, it results in better scores for SMUG and SMUG<sub>base</sub>.

**Choice of baseline for IG.** Another reason why SMUG<sub>base</sub> and SMUG outperform IG is that, they apply IG to the first layer of the network (as opposed to the input/image layer). IG attribution is known to be noisy [20], further attributions produced by IG depend on the choice of the baseline [13, 25, 22]. The reason for this can be observed from Eq. 2. In Eq. 2,  $x'$  represents the baseline “information-less” image. Based on this, the input dimensions close to the baseline receive very low attributions even though they might be important. i.e., if  $i, j$  denote pixel locations, when

$x_{i,j} - x'_{i,j} \approx 0$ , the attribution  $IG_{i,j}(x) \approx 0$  irrespective of how important the pixels are. For instance, black pixels (RGB value of  $(0, 0, 0)$ ) will receive an exact 0 attribution for a black baseline. In fact, for any baseline, IG will be insensitive to the dimensions close to the baseline value. We believe that 0 activations in the first layer is a more natural baseline for IG for ReLU networks, which is quantitatively observable in better LSC scores.

**SMUG vs SMUG<sub>base</sub>.** Based on the LSC scores in Table 1, SMUG narrowly outperforms SMUG<sub>base</sub>. Recall however, that SMUG is a sparser version of SMUG<sub>base</sub> obtained from the minimization constraints of the SMT solver (Eqns. 3 and 4). To compare the 2 methods, we also measure **sparsity** of the masks. This is defined as the fraction of the pixels with non-zero attributions to the total number of pixels in the image. We find that the average SMUG<sub>base</sub> mask has a sparsity of **43%** while the average SMUG mask has a sparsity of just **17%**. This is also evident from the examples in Fig. 1. The symbolic encoding effectively masks pixels less relevant to the prediction while retaining the model’s confidence, and hence maintains a high LSC score.

**GROUNDTRUTH, CENTERBOX.** Based on qualitative examples Figs. 1, we can observe that in almost all cases the object is at the center of the image. Hence, CENTERBOX is likely to capture some part of the image. Further, a fair number of objects are large covering much of the image e.g., Fig. 1 *Macaque, Collie, Robin, Dalmation*. In these cases, the groundtruth bounding boxes are also large to fully cover all pixels corresponding to the object. In contrast, SMUG saliency maps are more compact for both large and small objects, and hence achieve a better LSC score.

## 5.2 Discussion: Analyzing the LSC metric and OPTBOX

The LSC metric makes a trade-off when optimizing for both compactness and confidence. To analyze this we look at several qualitative examples (Fig. 2) of the bounding boxes identified by the OPTBOX brute-force approach to optimize the LSC metric. We observe that OPTBOX often finds bounding boxes that are very small, typically a sufficiently discriminative region or pattern in the image (e.g., *typewriter keyboard, paddlewheel, vine snake* in Fig. 2), or the full object if the object is itself small (e.g., *basketball, plunger*). In all these cases we find that SMUG highlights several other aspects of the object as well (typewriter’s tape; the dog’s eyes, nose and ears, etc.)

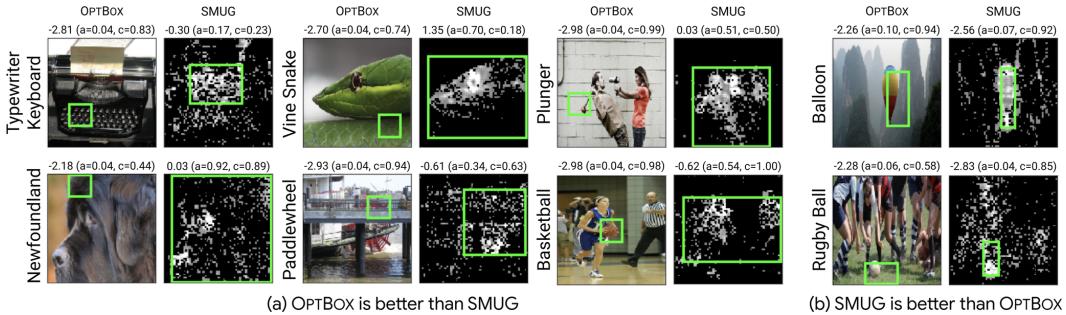


Figure 2: OPTBOX vs SMUG. 6 columns on the left correspond to the images for which OPTBOX gets a better score than SMUG. 2 columns on the right correspond to the images for which SMUG got a better LSC score than OPTBOX. Numbers at the top denote the LSC score, fractional area of the bounding box  $a$  and the confidence of the classifier  $c$  on the cropped region.

In particular, OPTBOX seems to exploit the fact that the LSC metric is somewhat invariant to the size/scale of the object (because of the resizing). This can be seen from the examples in Fig. 3. LSC favors compact regions where the model continues to have high confidence, and the OPTBOX score (Fig. 3(f)) is very good for the box around the small cat alone. However, as evidenced from the confidences of the model for Fig. 3 (a), (b) and (c) a good saliency map should assign greater attribution to  $\text{cat}_{\text{big}}$ . And SMUG and IG do exactly that, highlighting that both SMUG and IG capture the model’s behaviour correctly.

## 5.3 Text Dataset: Beer Reviews

We also evaluate our method on the model trained to rate Beer Reviews. Table 2 presents the results comparing the methods using the LSC score. Our approach performs competitively with the other

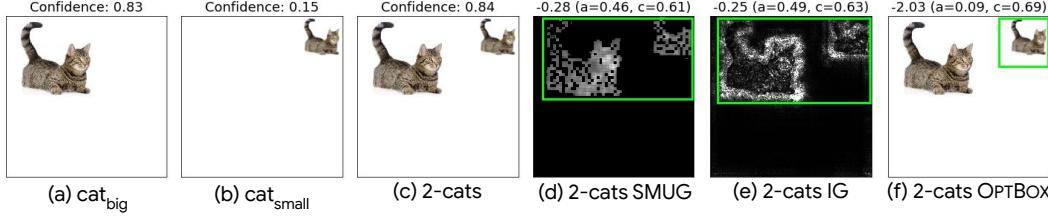


Figure 3: (a) shows an image of a cat ( $\text{cat}_{\text{big}}$ ) placed on a white background that is classified with a confidence of 0.83. (b) shows an image of the same cat ( $\text{cat}_{\text{small}}$ ), scaled to a quarter of its original size, that is classified with a confidence of 0.15. (c) By placing  $\text{cat}_{\text{big}}$  next to  $\text{cat}_{\text{small}}$  we observe a significant jump in the classifier’s confidence from 0.15 with  $\text{cat}_{\text{small}}$  alone to 0.84 on 2-cats. While (d) SMUG and (e) IG correctly attribute the model confidence to  $\text{cat}_{\text{big}}$ , (f) OPTBOX exploits the fact that the LSC metric rescales the object in the salient bounding box.

methods including SIS, IG, and GROUNDTUTH (human annotated words). The solution of SIS consists of multiple disjoint set of words of varying relevance. A saliency map is constructed by scoring the words in the sets between  $(0,1]$  on the basis of relevance of the set. One thing to note is that, unlike images where the masked image can be cropped and resized as input to compute the LSC metric, the same strategy cannot be followed on the text model. Specifically, ImageNet models are trained with extensive data-augmentation including random crops and resizing, and the modified image is less likely to be out-of-distribution. Whereas in the case of text, the model doesn’t employ any form of meaningful augmentation, and the masked text is much more likely to come from a distribution that has not been seen during training. Hence, we present qualitative examples in Fig. 4, and several more randomly selected examples from the test set in the supplementary material.

Table 2: **Beer Reviews.** We report the median LSC score ( $\downarrow$  lower is better) with 75th (top) and 25th (bottom) percentile, and mean Win% score ( $\uparrow$  higher is better) with binomial proportion confidence interval (normal approximation) for 787 text examples in the annotation set of Beer Reviews.

Method	SMUG	SMUG <sub>base</sub>	SIS	IG	GROUNDTRUTH
<b>LSC<math>\downarrow</math></b>	$-2.72_{-2.81}^{+2.62}$	$-2.75_{-2.83}^{+2.65}$	$-2.67_{-2.75}^{+2.60}$	$-2.68_{-2.79}^{+2.60}$	$-1.66_{-1.97}^{+1.34}$
<b>Win%<math>\uparrow</math></b>	$34.19 \pm 3.42$	$53.78 \pm 3.59$	$18.92 \pm 2.82$	$15.00 \pm 2.57$	$0.27 \pm 0.37$



Figure 4: Example comparing our method (SMUG, SMUG<sub>base</sub>) with SIS and IG on a test sample from the Beer Reviews dataset. Green color signifies a positive relevance, red color signifies negative relevance. The underlined words are human annotations.

## 6 Conclusion

We present an approach that uses SMT solvers for computing minimal input features that are relevant for a neural network prediction. In particular, it uses attribution scores from Integrated Gradients to find a subset of important neurons in the first layer of the network, which allows the SMT encoding of constraints to scale to larger networks for finding minimal input masks. We evaluate our technique

to analyze models trained on image and text datasets and show that the saliency maps generated by our approach are competitive or better than existing approaches and produce sparser masks.

## Broader Impact

Our work is a step towards improving the explainability of neural network model decisions. With deep neural networks increasingly being applied to critical domains such as criminal justice and healthcare [10, 8] there is a growing necessity to make these models more interpretable. As with other model explanation techniques, including the one presented in this work, our goal is to help decision makers understand and trust the functionality of their models. As discussed in Sec. 5.2, our method and proposed evaluation can be used to understand the model as well as study potential biases (which we include in the supplement).

## References

- [1] D. Alvarez-Melis and T. S. Jaakkola. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*, 2018.
- [2] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7), 2015.
- [3] C. Barrett and C. Tinelli. Satisfiability modulo theories. In *Handbook of Model Checking*, pages 305–343. Springer, 2018.
- [4] N. Bjørner, A.-D. Phan, and L. Fleckenstein. *vz*-an optimizing smt solver. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 194–199. Springer, 2015.
- [5] B. Carter, J. Mueller, S. Jain, and D. Gifford. What made you do this? understanding black-box decisions with sufficient input subsets. *arXiv preprint arXiv:1810.03805*, 2018.
- [6] P. Dabkowski and Y. Gal. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*, pages 6967–6976, 2017.
- [7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [8] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115–118, 2017.
- [9] D. Gopinath, H. Converse, C. Pasareanu, and A. Taly. Property inference for deep neural networks. In *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 797–809, 2019.
- [10] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22):2402–2410, 2016.
- [11] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 9734–9745, 2019.
- [12] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu. Safety verification of deep neural networks. In *International Conference on Computer Aided Verification*, pages 3–29. Springer, 2017.
- [13] A. Kapishnikov, T. Bolukbasi, F. Viégas, and M. Terry. Xrai: Better attributions through regions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4948–4957, 2019.

- [14] G. Katz, C. Barrett, D. Dill, K. Julian, and M. Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks, 2017.
- [15] Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. 2010.
- [16] J. Macdonald, S. Wäldchen, S. Hauch, and G. Kutyniok. A rate-distortion framework for explaining neural network decisions. *arXiv preprint arXiv:1905.11092*, 2019.
- [17] J. McAuley, J. Leskovec, and D. Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In *2012 IEEE 12th International Conference on Data Mining*, pages 1020–1025. IEEE, 2012.
- [18] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.
- [19] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju. Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 180–186, 2020.
- [20] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.
- [21] S. Srivastava, S. Gulwani, and J. S. Foster. Vs 3: Smt solvers for program verification. In *International Conference on Computer Aided Verification*, pages 702–708. Springer, 2009.
- [22] P. Sturmels, S. Lundberg, and S.-I. Lee. Visualizing the impact of feature attribution baselines. *Distill*, 2020. <https://distill.pub/2020/attribution-baselines>.
- [23] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 2017.
- [24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.
- [25] S. Xu, S. Venugopalan, and M. Sundararajan. Attribution in scale and space. *arXiv preprint arXiv:2004.03383*, 2020.
- [26] X. Zhang, A. Solar-Lezama, and R. Singh. Interpreting neural network judgments via minimal, stable, and symbolic corrections. In *Advances in Neural Information Processing Systems*, pages 4874–4885, 2018.

## Supplementary Materials

Here, we present additional quantitative and qualitative examples supporting the results in Sec. 5. In particular Sec. A shows qualitative examples on MNIST, Sec. B presents quantitative and qualitative comparisons with regard to the choice of key parameters of our proposed SMUG explanation technique. Sec. C presents additional qualitative examples comparing the output of SMUG with other methods on text samples from the Beer Reviews dataset. Sec. D presents several more qualitative examples from ImageNet comparing the saliency masks produced by SMUG with those produced by other methods. In Sec. D.1 we also discuss an example where an explanation technique can be used to identify potential biases of the trained model. Finally, Sec. E presents some examples of concrete constraints that the solver optimizes.

### A MNIST

In this section, we present some more details about our experiments with MNIST using the full SMT encoding from Eq. 1 in Sec 3.1. Table 3 shows the SMT solver runtimes and whether the constraints were solved (SAT). We observe that with a timeout of 60 minutes, the SMT solver could solve the full constraints for only 34 of the 100 images. Another interesting point to observe is that the solver could not solve any of the instances for digits 0 and 3. We also show some of the minimal masks generated by the SMT solver for few MNIST images in Figure 5.

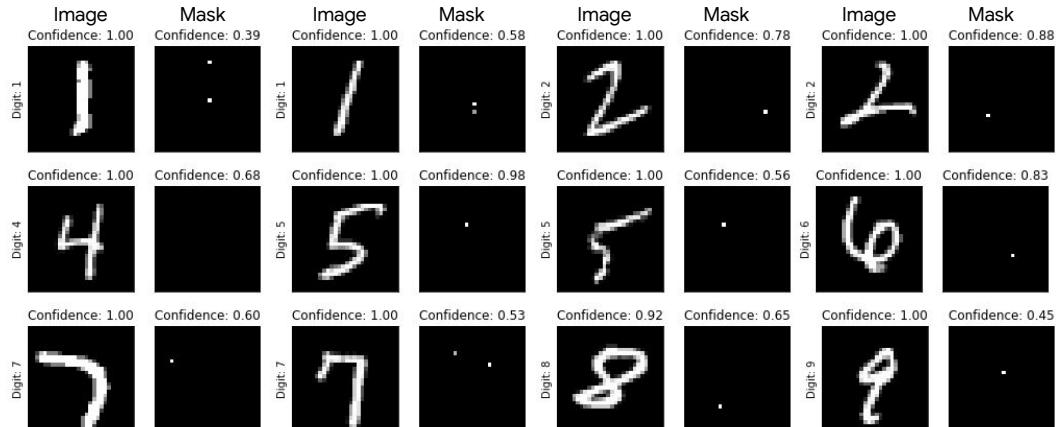


Figure 5: MNIST images and the corresponding masks corresponding to Sec. 3.1.

Table 3: **Solver Runtime and SAT instances.** We report the average solver runtime and instances solved per digit with a timeout set at 60 mins.

Digit	0	1	2	3	4	5	6	7	8	9	ALL
<b>Runtime (mins)</b>	N.A.	31.19	45.26	N.A.	33.09	35.68	42.80	53.11	36.05	19.62	35.59
<b>SAT Instances</b>	0/8	8/14	4/8	0/11	8/14	4/7	4/10	2/15	1/2	3/11	34/100

## B Hyperparameter Choices

In our proposed approach, the choice of top- $k$ , and  $\gamma$  (Eqns. 3, 4, 5) have an effect on the final quality of the explanations, and the time it takes for the solver to identify the mask. This section presents quantitative and qualitative comparisons for different choices of top- $k$  and  $\gamma$ .

### B.1 Quantitative comparison for different choices of top- $k$ and $\gamma$

Fig. 6 presents quantitative comparisons for different choices of top- $k$  and  $\gamma$ .

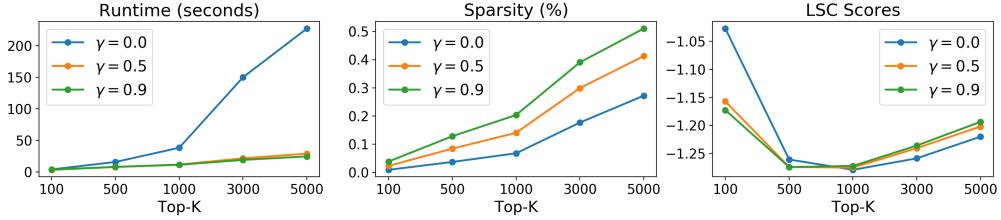


Figure 6: Hyperparameter Comparision

**top- $k$**  We analyze images with top- $k \in \{500, 1000, 3000, 5000\}$ . Increasing the  $k$  value increases the receptive field and the discovered input masks also grow in size with increasing values of  $k$ . Figure 6 shows how the solver runtime and the mask size vary with  $k$ . As expected, larger  $k$  values results in larger number of constraints and therefore larger solving times as well as larger mask sizes.

**Gamma** We analyze the effect of  $\gamma \in \{0.0, 0.5, 0.9\}$ , also shown in Fig. 6. We observe that by decreasing gamma values, the masks become sparser. The key reason behind this is that with smaller gamma values, the SMTsolver is less constrained to maintain the original neural activations for the selected neurons, and hence can ignore additional input pixels that do not have a large effect. It is also noteworthy to notice that the solver run-time increases with decreasing value of  $\gamma$ .

## B.2 Top-k vs $\gamma$ on ImageNet - Qualitative examples

Fig. 7 presents qualitative examples of the saliency maps on Imagenet examples for different choices of top- $k$  and  $\gamma$ .

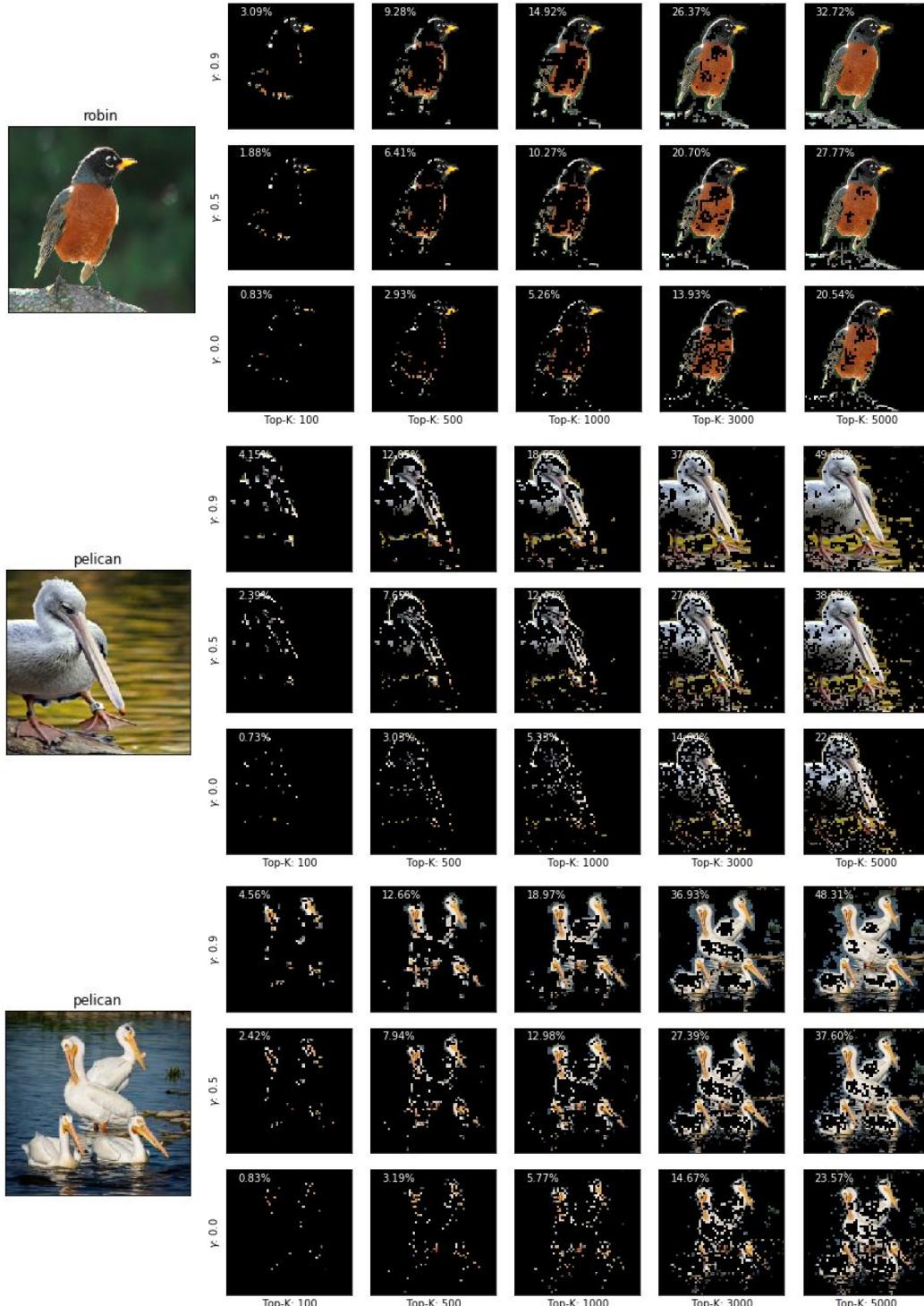


Figure 7: Qualitative examples of the masks generated by SMUG on examples from Imagenet for different choices of top- $k$  (columns) and  $\gamma$  (rows).  $\gamma = 0$  is the most minimal mask. Even at low values of top- $k$  and  $\gamma$  SMUG highlights pixels relevant for the object class.

## C Qualitative text examples

Fig. 8 presents additional examples comparing the output of SMUG with SIS, SMUG<sub>base</sub>, and IG on the Beer Reviews dataset.



Figure 8: Examples comparing our method (SMUG, SMUG<sub>base</sub>) with SIS and IG on test samples from the Beer Reviews dataset. Green color signifies a positive relevance, red color signifies negative relevance. The underlined words are human annotations.

## D Additional qualitative image examples: SMUG

Fig. 9 presents boolean masks and the saliency maps produced by SMUG on several ImageNet examples. Fig. 10 and 11 present additional examples comparing the saliency masks and bounding box (for LSC) produced by SMUG, SMUG<sub>base</sub>, and IG.

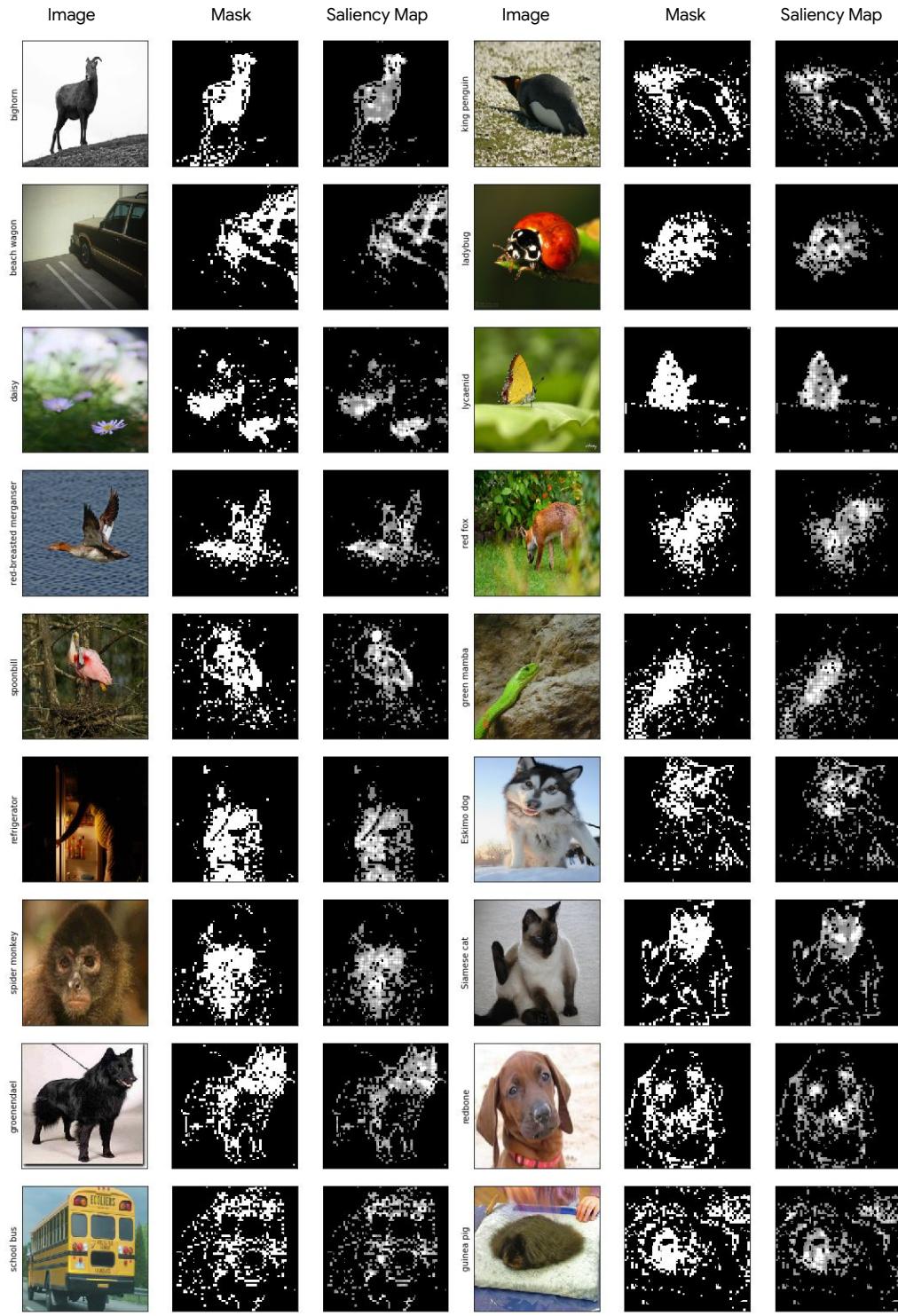


Figure 9: Examples showing the boolean masks and the saliency maps produced by SMUG on several ImageNet examples.

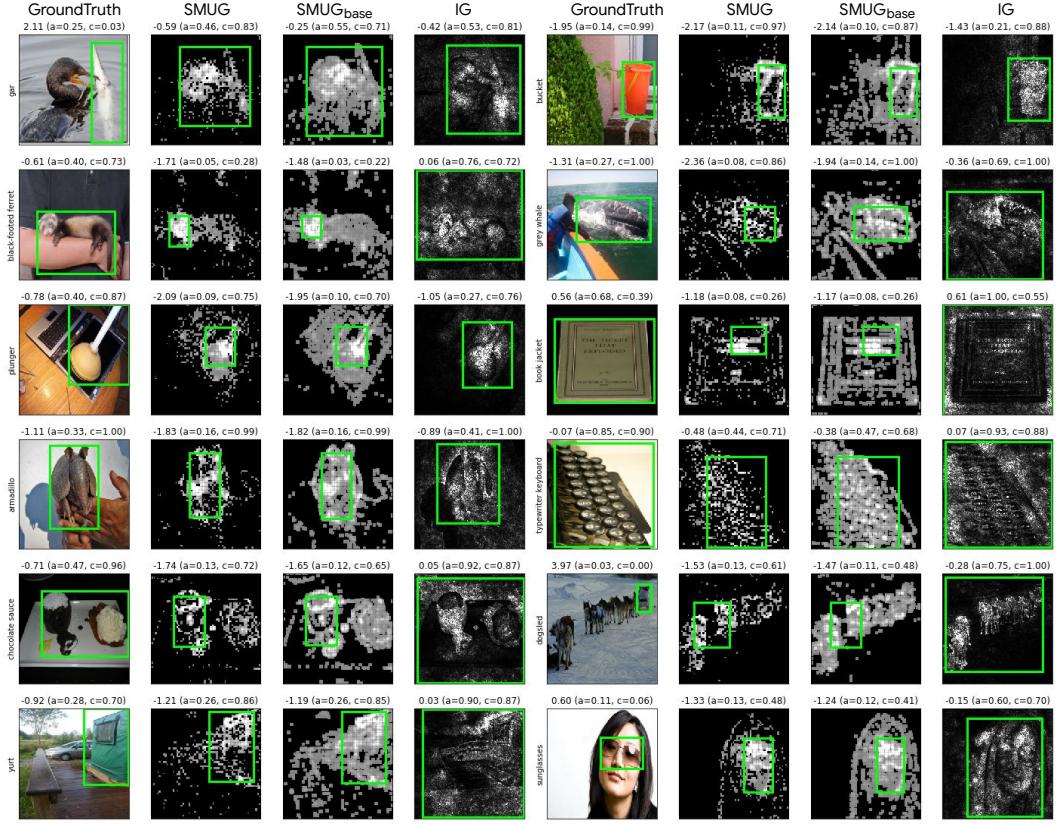


Figure 10: Examples comparing saliency maps where SMUG outperforms SMUG<sub>base</sub>, and IG. The green box on the original image highlights the groundtruth box; for the saliency methods it represents the bounding box with the best LSC score. Numbers on top denote the LSC score, the fractional area of the bounding box ( $a$ ), and the confidence of the classifier ( $c$ ) on the cropped region.

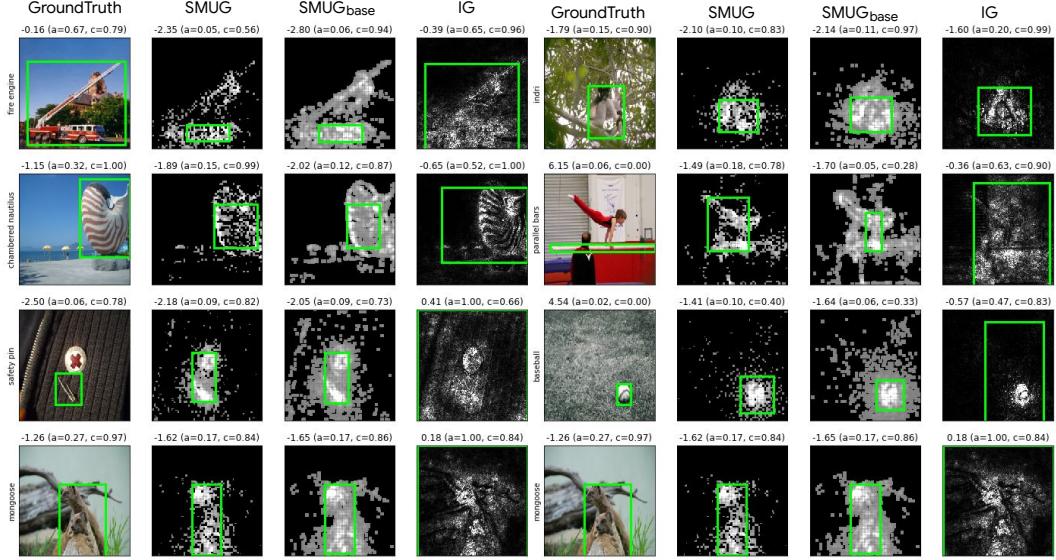


Figure 11: Examples comparing saliency maps where SMUG<sub>base</sub>, or IG outperforms SMUG. The green box on the original image highlights the groundtruth box; for the saliency methods it represents the bounding box with the best LSC score. Numbers on top denote the LSC score, the fractional area of the bounding box ( $a$ ), and the confidence of the classifier ( $c$ ) on the cropped region.

## D.1 Biases

Model explanation techniques can also be particularly useful in studying model biases. Fig. 12 shows some examples where the model correctly predicts the class as “parallel bars” but it appears to actually focus more on the person leaping over the bar to make the prediction as opposed to looking at the bar itself. These can help us understand and identify cases where the model has developed a bias (in this case, based on training data).

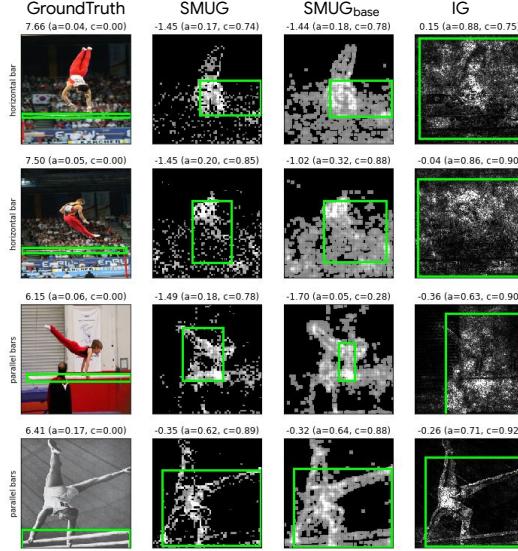


Figure 12: [Model bias] Examples where the model correctly predicts the class as “parallel bars” and “horizontal bars” for the corresponding inputs but the model’s focus is on the person leaping over the bar (i.e., the predicted object class) itself.

## E SMT constraints

In this section, we present an example set of SMT constraints obtained by our technique for an example image from ImageNet. For brevity, we show the set of constraints for  $k = 5$  and  $\gamma = 0$ . As mentioned in Sec 5.1, each masking variable  $M_{ij}$  corresponds to a  $4 \times 4$  grid of pixels, where the grid is denoted by  $X_{i:i+3,j:j+3}$ . For example, the mask variable  $M_{132,135}$  corresponds to the pixel grid  $X_{132:135,132:135}$ . Following Eq. 4 in Sec 3.3, the SMT constraints corresponding for top  $k = 5$  IG positive attributions in the first layer are given by:

$$99.53X_{132,132} - 58.37X_{132,136} + 4.88X_{132,140} - 141.25X_{136,132} + 639.97X_{136,136} + 10.29X_{136,140} - 9.66X_{140,132} + 20.30X_{140,136} - 25.19X_{140,140} - 0.58 > 0$$

$$-270.67M_{120,150} + 101.23M_{142,144} + 10.38M_{113,124} + 207.98M_{122,121} + 640.64M_{121,121} - 100.72M_{121,126} + 25.06M_{121,165} - 75.49M_{121,156} + 75.47M_{112,154} - 0.36 > 0$$

$$2925.38X_{144,132} - 395.09X_{144,136} + 81.61X_{148,132} - 999.88X_{148,136} - 82.70X_{152,132} + 17.08X_{152,136} + 0.21 > 0$$

$$-20.87X_{76,80} + 8.40X_{76,84} - 122.72X_{80,80} + 929.71X_{80,84} + 85.52X_{84,80} + 138.99X_{84,84} - 0.01 > 0$$

$$231.34X_{168,148} + 722.71X_{168,152} + 80.18X_{172,148} + 663.96X_{172,152} + 5.37X_{176,148} + 4.63X_{176,152} + 0.12 > 0$$