

# Scaling Symbolic Methods using Gradients for Neural Model Explanation

subhamsahoo@, vsubhashini@, leeley@, rising@, pfr@  
Google Research

# Goal

Generate explanations for a neural network's prediction by identifying a minimal subset of input features critical to the model's prediction.

# Goal

Generate explanations for a neural network's prediction by identifying a minimal subset of input features critical to the model's prediction. The purpose of our work is address these 2 major issues -

1. Scalability of SMT Solvers to Deep Neural Networks.

# Goal

Generate explanations for a neural network's prediction by identifying a minimal subset of input features critical to the model's prediction. The purpose of our work is address these 2 major issues -

1. Scalability of SMT Solvers to Deep Neural Networks.
2. Overcome out-of-distribution inputs issue which is seen in perturbation based explainability methods.

# Satisfiability Modulo Theory (SMT) Solvers

Highly engineered tools for solving NP-Hard Problems - Program Verification, theorem proving etc.



```
i = 0;  
x = j;  
while (i < 50) {  
    i++;  
    x++;  
}  
if (j == 0)  
    assert (x >= 50);
```

Image credit: <https://leodemoura.github.io/slides.html>

# Satisfiability Modulo Theory (SMT) Solvers

Highly engineered tools for solving NP-Hard Problems - Program Verification, theorem proving etc.

$$\begin{aligned} & \sin^3(x) = \cos(x \log(y)) \\ & (x > 3.0 \vee y < 2.0) \wedge \\ & (x = y \vee x \neq y - 1.0) \wedge \\ & \quad y < 1.0 \end{aligned}$$



```
i = 0;
x = j;
while (i < 50) {
    i++;
    x++;
}
if (j == 0)
    assert (x >= 50);
```

Image credit: <https://leodemoura.github.io/slides.html>

# Satisfiability Modulo Theory (SMT) Solvers

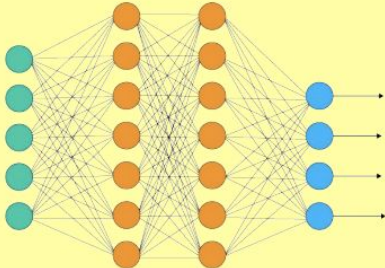
Highly engineered tools for solving NP-Hard Problems - Program Verification, theorem proving etc.


$$\sin^3(x) = \cos(x \log(y))$$

$$(x > 3.0 \vee y < 2.0) \wedge$$

$$(x = y \vee x \neq y - 1.0) \wedge$$

$$y < 1.0$$





```

i = 0;
x = j;
while (i < 50) {
  i++;
  x++;
}
if (j == 0)
  assert (x >= 50);

```

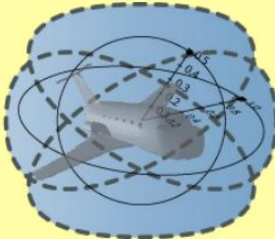


Image credit: <https://leodemoura.github.io/slides.html>

# The Issue of scalability

SMT solvers in their current form are difficult to scale to large networks and are limited to small NNs (typically feat. ~5K Neurons) \*.

\* Gopinath et al., 2019; Ignatiev et al., 2019



# The Issue of scalability

SMT solvers in their current form are difficult to scale to large networks and are limited to small NNs (typically feat. ~5K Neurons) \*.

Why?

- The number of constraints grow linearly with increasing network size.
- SMT decision procedure for Nonlinear Real Arithmetic is doubly exponential.
- Designing improved solvers (ex. reluplex) has led to incremental improvements.

In this work we wish to improve scalability by leveraging gradient informations.

\* Gopinath et al., 2019; Ignatiev et al., 2019

# Model Explanation

- an application where symbolic methods can be scaled by leveraging gradients.

Identify and highlight input features relevant for neural network's prediction.

Image



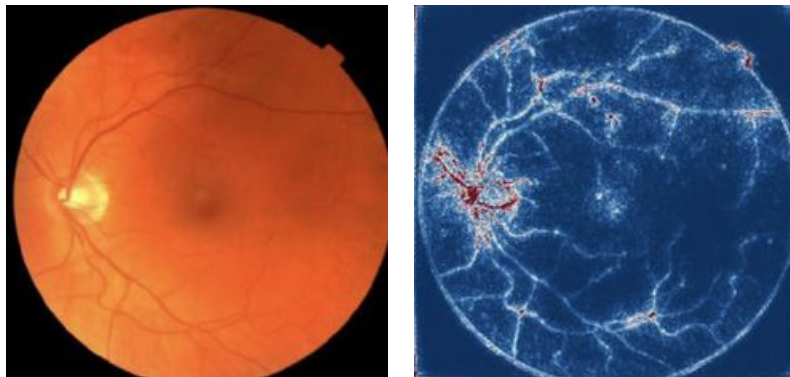
Text

11.2 oz bottle split and poured into a new belgium **globe** . 4.3 % abv , 4°c - 6°c , 40199 on label . a - faintly **cloudy** pink in color with a dense white head . frilly lace drapes across the glass **delicately** . **s** - **fresh** bushels of **raspberries just rinsed** gives the aroma an **inviting nose** , **soft** wheat and lemonade pull through and 'fake-up . the aroma . it almost begins **smelling** like a berry weiss-sunset wheat combo . initially good though ... **t** - **fruity** raspberry is far too syrupy with a lingering corn syrup finish . light wheat in the background balanced just a bit but it 's overly fake in it 's flavor profile . it almost seems as if lemonade is blended in . **m** - sugary sweet and slight with a highly carbonated finish and light body . **o** - overly sweet and fake , it 's got a good aroma initially **but** definitely lacking elsewhere . too syrupy but not awful .

Predict ratings for  
aroma of a beer from  
customer reviews

# Model Explanation

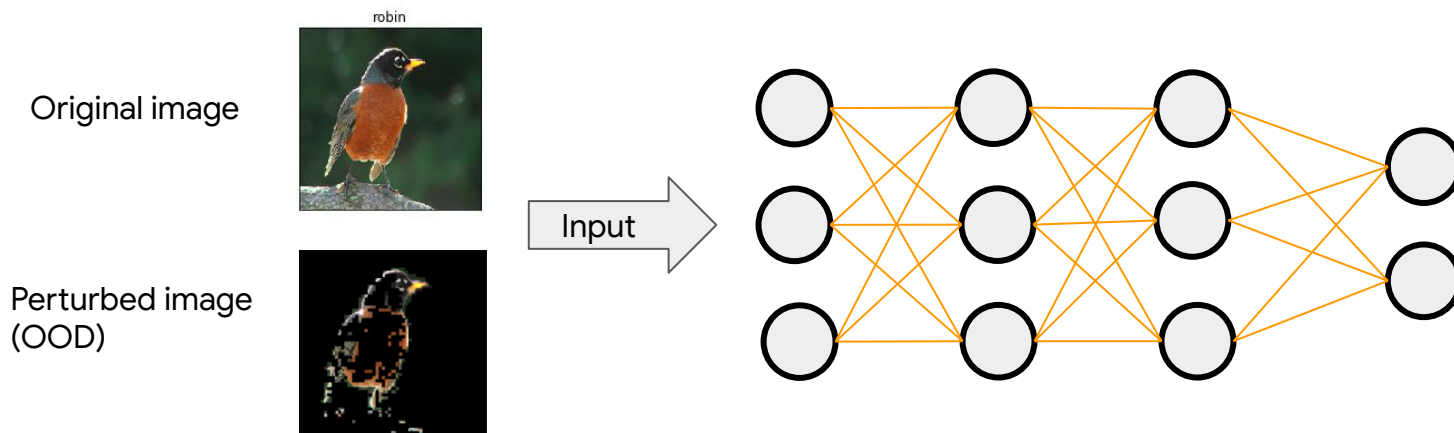
- an application where symbolic methods can be scaled by leveraging gradients.
- Better understanding of what the model has learnt.
- Models can perform tasks sometimes better than humans and can be used to discover new things. Ex. Predicting anemia from retinal fundus.



Source: <https://www.blog.google/technology/health/anemia-detection-retina/>

# The Issue of Out Of Distribution input

- A lot of the saliency methods \*, esp. masking methods, perturb input and analyze the change in activations to identify important features.
- The perturbed image is out of distribution.



\* Carter et al., 2018; Macdonald et al., 2019; Fong et al., 2017

# The Issue of Out Of Distribution input

Dataset: MNIST

Neural Network: Fully Connected [784 (input), 32 (relu), 10 (softmax)]

Objective \*: Find a minimal mask s.t. the masked image is classified correctly.

$$\min\left(\sum_{ij} M_{ij}\right) : \operatorname{argmax} L_{|L|-1}(N_{\theta}(X)) = \operatorname{argmax} L_{|L|-1}(N_{\theta}(M \odot X))$$

$X \in \mathbb{R}^{m \times n}$       Input image

$M \in \{0, 1\}^{m \times n}$       Unknown binary mask

$L_{|L|-1}(N_{\theta}(X))$       Logits of the final layer

$N_{\theta}$                       Neural Network

\* Carter et al., 2018; Macdonald et al., 2019

# The Issue of Out Of Distribution input

Dataset: MNIST

Neural Network: Fully Connected [784 (input), 32 (relu), 10 (softmax)]

Objective \*: Find a minimal mask s.t. the masked image is classified correctly.

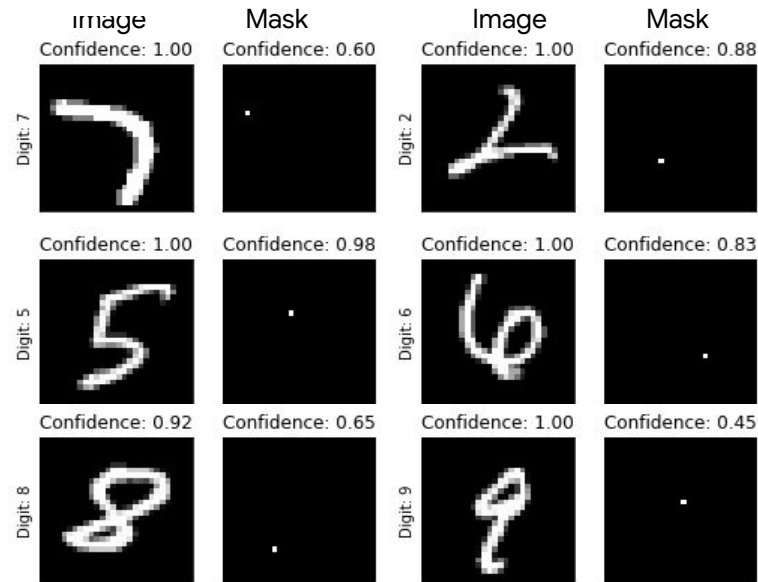
$$\min(\sum_{ij} M_{ij}) : \operatorname{argmax} L_{|L|-1}(N_{\theta}(X)) = \operatorname{argmax} L_{|L|-1}(N_{\theta}(M \odot X))$$

$X \in \mathbb{R}^{m \times n}$  Input image

$M \in \{0, 1\}^{m \times n}$  Unknown binary mask

$L_{|L|-1}(N_{\theta}(X))$  Logits of the final layer

$N_{\theta}$  Neural Network



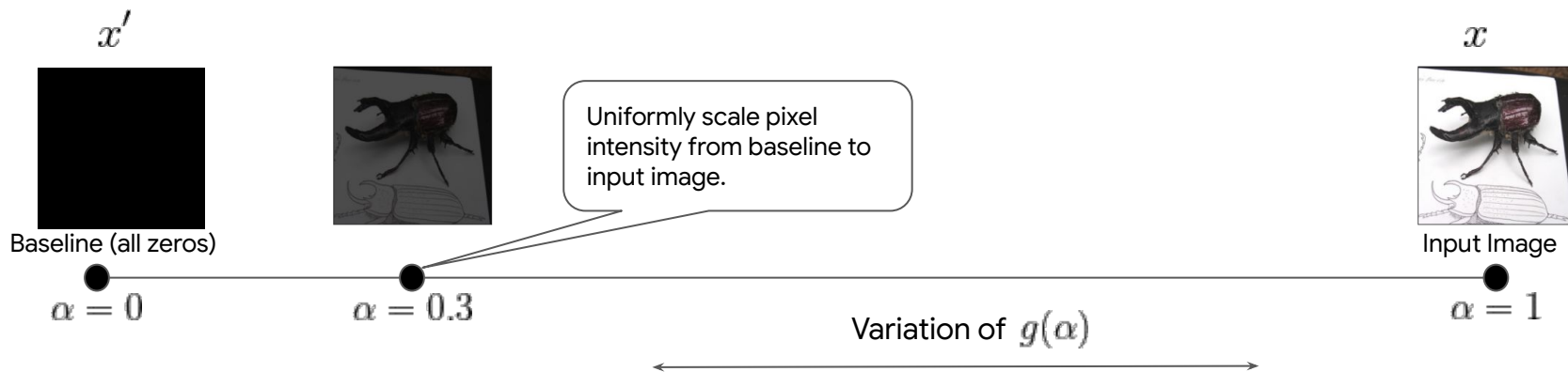
\* Carter et al., 2018; Macdonald et al., 2019

# Integrated Gradients

$$IG(x) = (x - x') \int_{\alpha=0}^1 \underbrace{\frac{\partial F(g(\alpha))}{\partial g(\alpha)}}_{\text{Gradient}} d\alpha$$

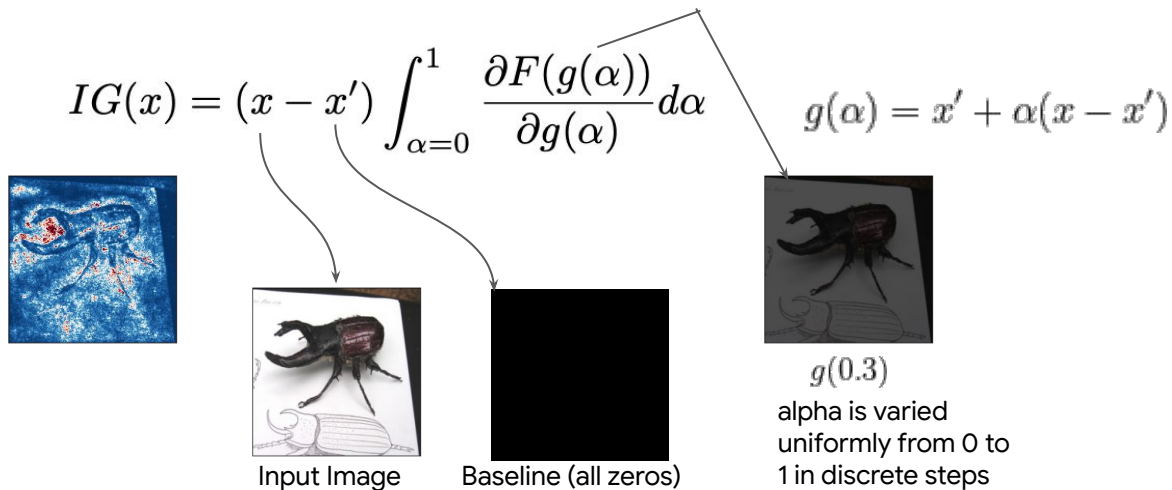
Model

$$g(\alpha) = x' + \alpha(x - x')$$



\* Sundarajan et al., 2017

# Integrated Gradients



\* Sundarajan et al., 2017

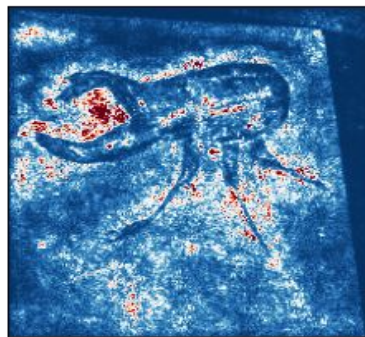


# Integrated Gradients - baseline issue

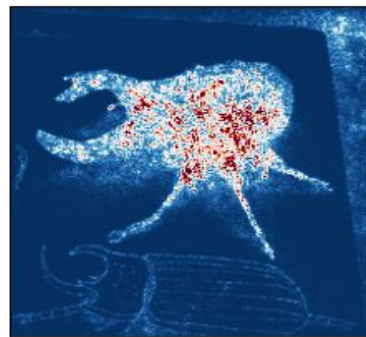
$$IG(x) = (x - x') \int_{\alpha=0}^1 \frac{\partial F(g(\alpha))}{\partial g(\alpha)} d\alpha$$



Image



IG Black Baseline



IG White Baseline

\* Kapishnikov et al., 2019

# SMT + Gradients a potential solution

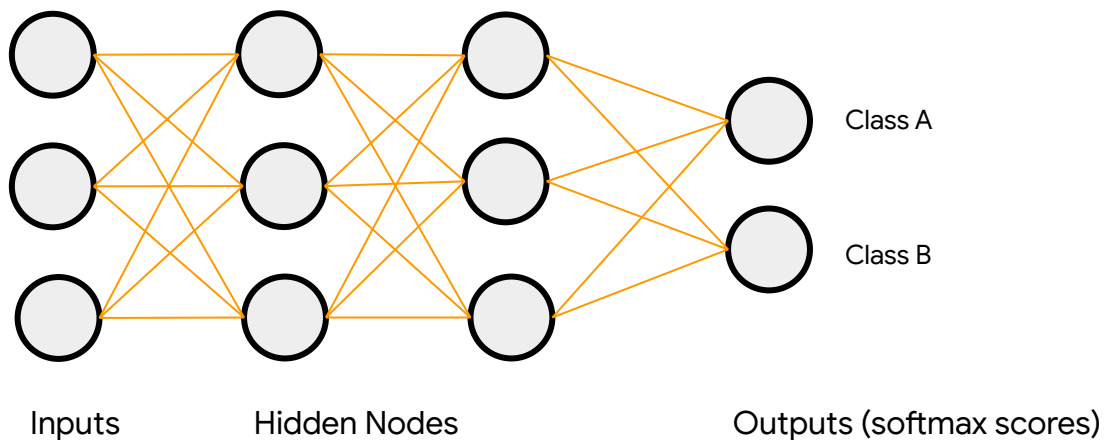
- Identify nodes (**top-k**) in the first layer that capture information relevant for the prediction using gradient based methods.

# SMT + Gradients a potential solution

- Identify nodes (**top-k**) in the first layer that capture information relevant for the prediction using gradient based methods.
- Find a minimal mask which retains a fraction (**gamma**) of the activations of these nodes instead of the final confidence (prevents ood issue).

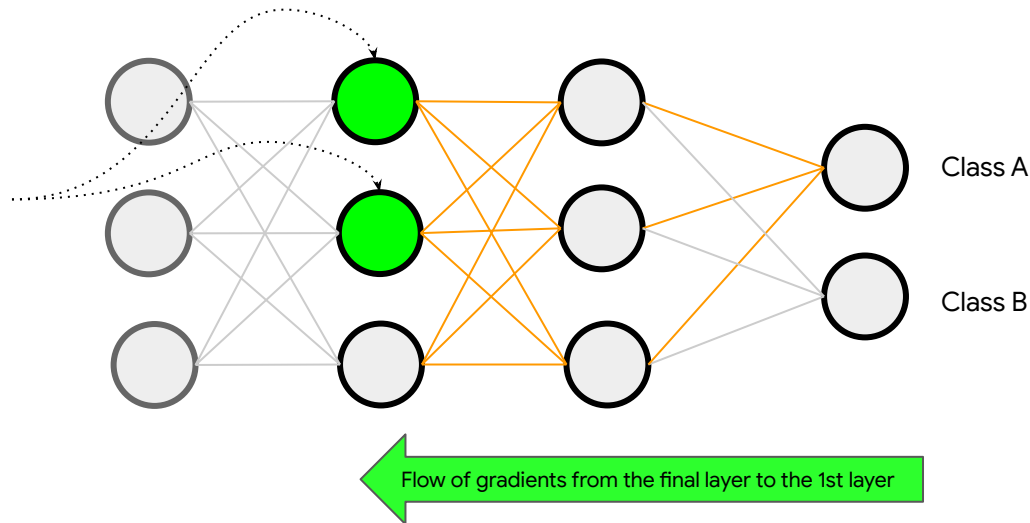
This eliminates the need to encode the full neural network.

# SMUG: Scaling Symbolic Methods Using Gradients



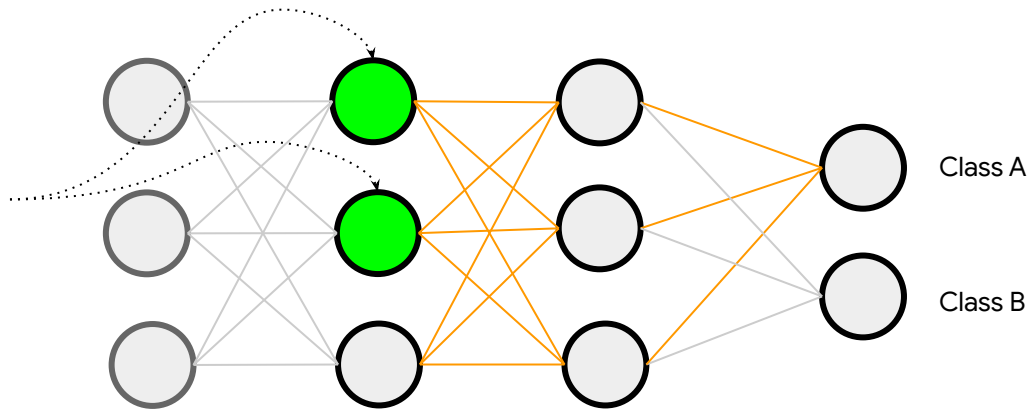
# Gradients from the deeper layers

Top-k nodes chosen based on Integrated Gradient scores. In this example  $k = 2$



# Gradients from the deeper layers

Top-k nodes chosen based on Integrated Gradient scores. In this example k = 2



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

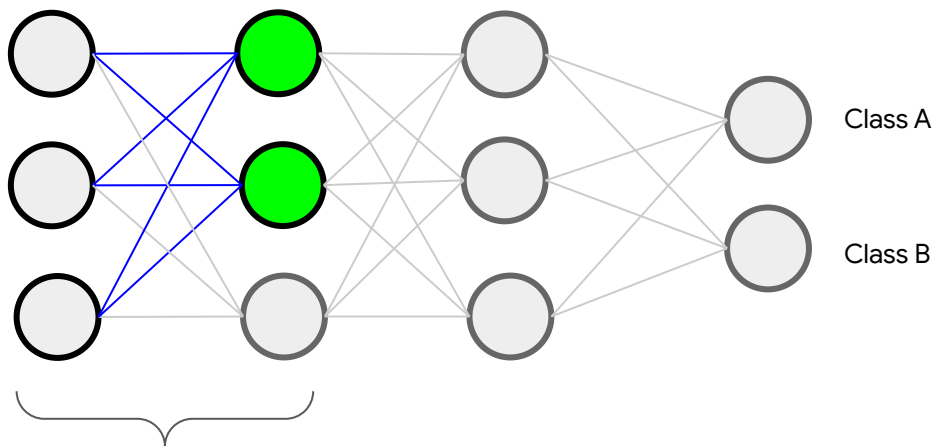
Baseline:  $\mathbf{0}$

First layer activations corresponding to the input

Flow of gradients from the final layer to the 1st layer

Integrated Gradients identifies nodes in the first layer that capture information relevant to the prediction

# Symbolic encoding of the layers near the input



Symbolic Encoding

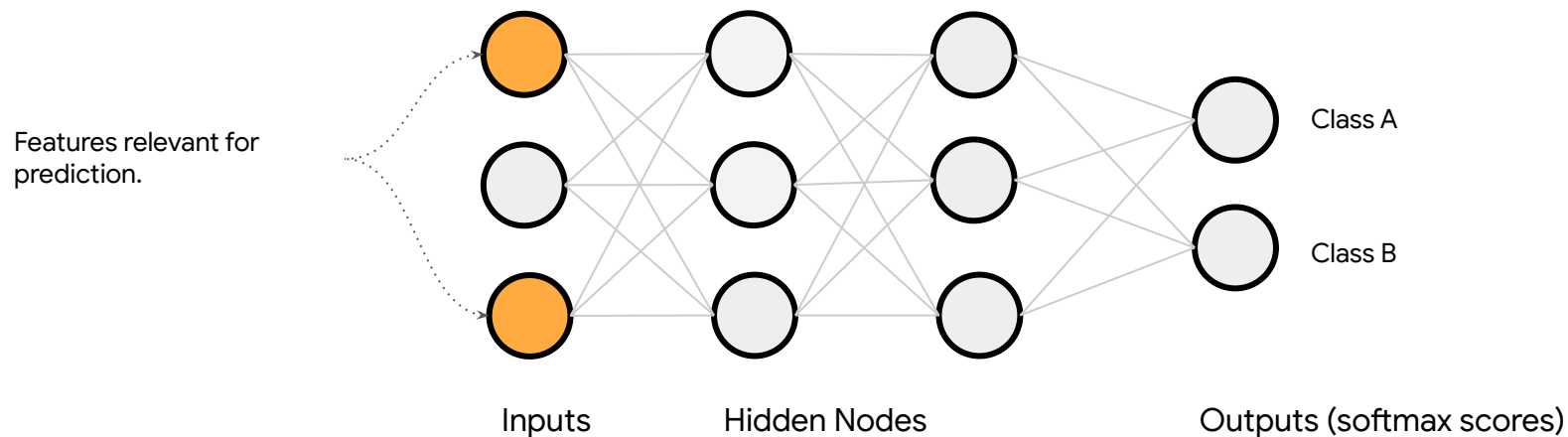
$$\exists M : \bigwedge_{1 \leq i \leq m, 1 \leq j \leq n} (M_{ij} == 0) \vee (M_{ij} == 1) \bigwedge_{\forall i \in D^k} o_i^m = (W_1(X \odot M) + b_1)_i$$

$$\bigwedge_{\forall i \in D^k} o_i = (W_1 X + b_1)_i \bigwedge_{\forall i \in D^k} o_i^m > \gamma \cdot o_i \bigwedge \text{minimize}(\sum_{ij} M_{ij})$$

M = Mask  
X = Image

$D^k$ , a set of  $k$  neurons with highest positive attributions in  $IG(N_\theta(x))$

# Identify relevant features

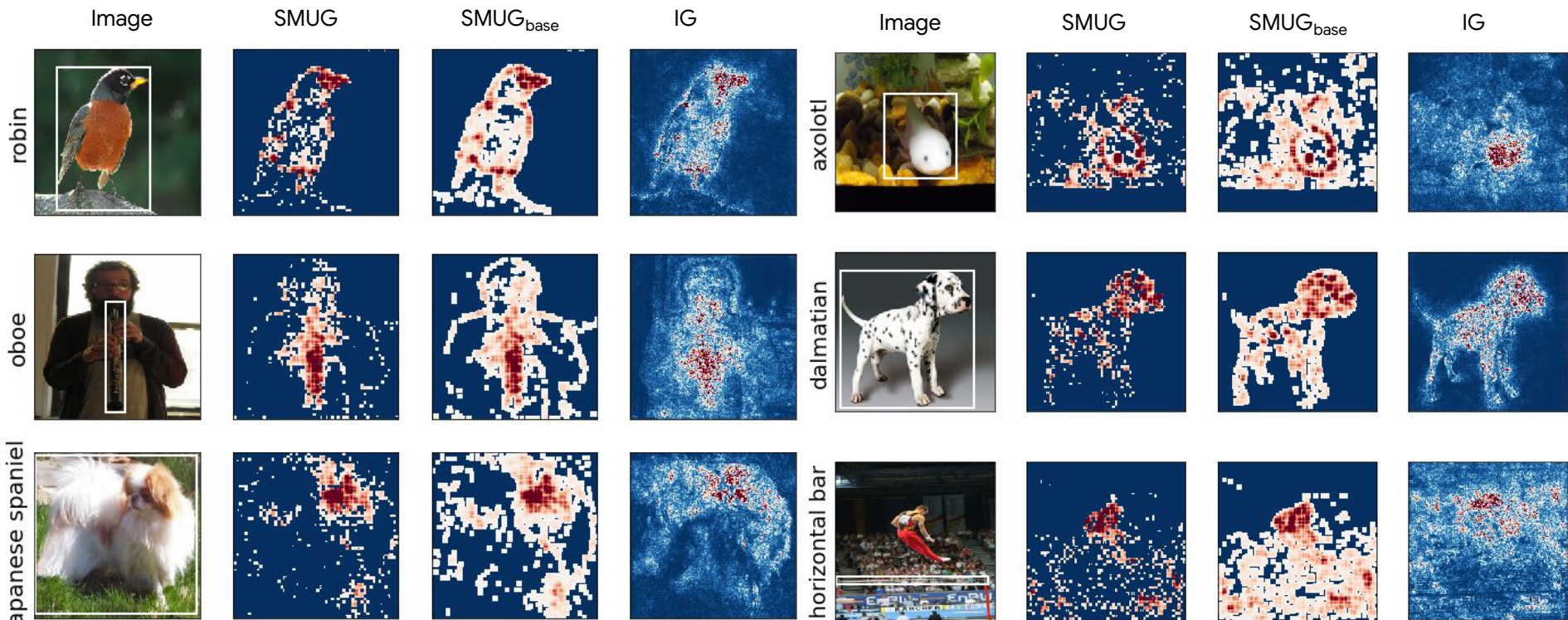


Use the IG attribution scores for the hidden nodes in the first layer to assign importance scores to mask pixels associated with that hidden node.

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



# Image Saliency



# Quantitative Analysis: LSC Score

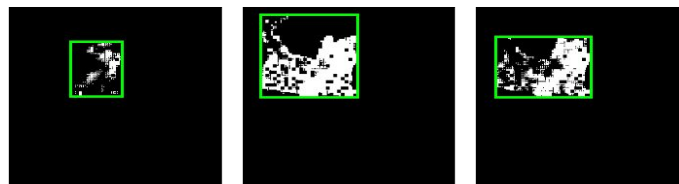
1. Convert a continuous valued saliency map to a boolean mask.



\* Dabkowski et al., 2017

# Quantitative Analysis: LSC Score

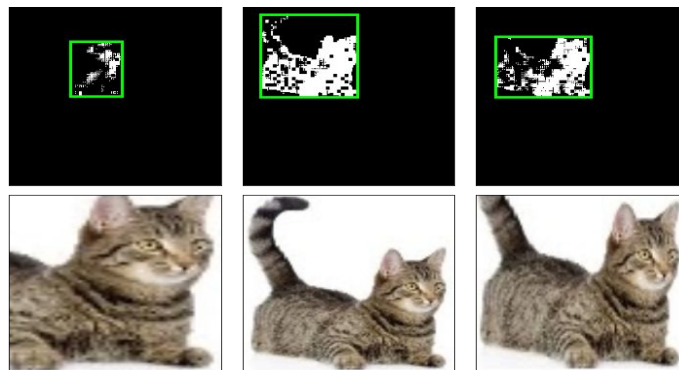
1. Convert a continuous valued saliency map to a boolean mask.
2. Find the tightest rectangular crop for the boolean mask.



\* Dabkowski et al., 2017

# Quantitative Analysis: LSC Score

1. Convert a continuous valued saliency map to a boolean mask.
2. Find the tightest rectangular crop for the boolean mask.
3. Compute the confidence of the classifier on the cropped image.



\* Dabkowski et al., 2017

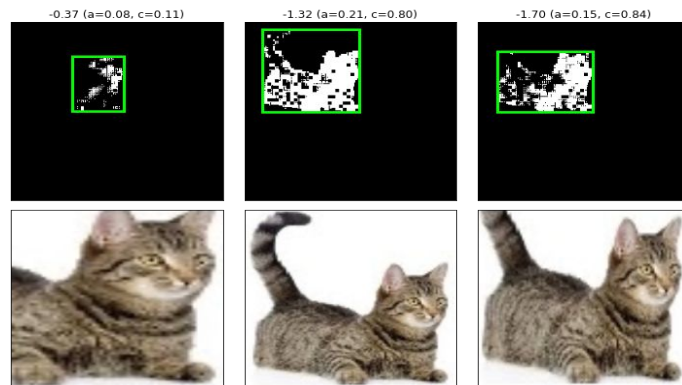
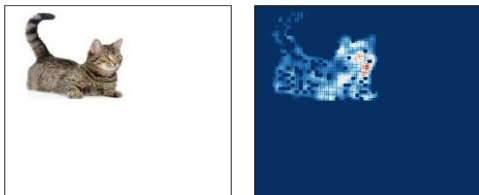
# Quantitative Analysis: LSC Score

1. Convert a continuous valued saliency map to a boolean mask.
2. Find the tightest rectangular crop for the boolean mask.
3. Compute the confidence of the classifier on the cropped image.
4. Compute the LSC score. Do multiple thresholding and report the best score.

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

$a$  = crop\_size / image\_size

$c$  = confidence of the classifier on the cropped image

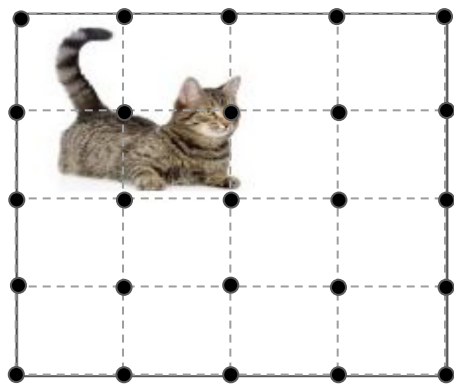


\* Dabkowski et al., 2017

# OptBox

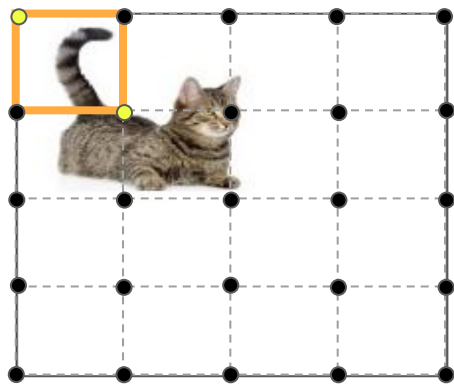


# OptBox



- Discretize the image into subgrids. For ImageNet we divide the image into a 10 x 10 grid.

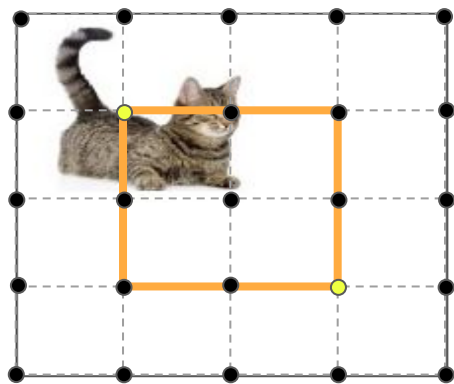
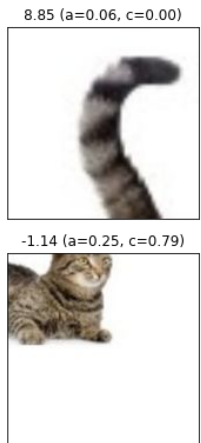
# OptBox



- Discretize the image into subgrids. For ImageNet we divide the image into a 10 x 10 grid.
- Pick any 2 points (to represent opposite corners of a rectangle) and evaluate the cropped region.

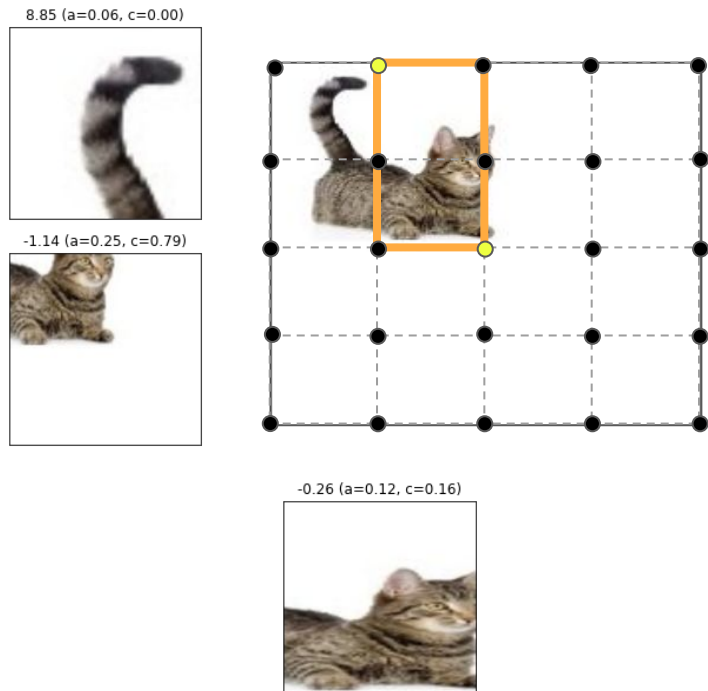


# OptBox



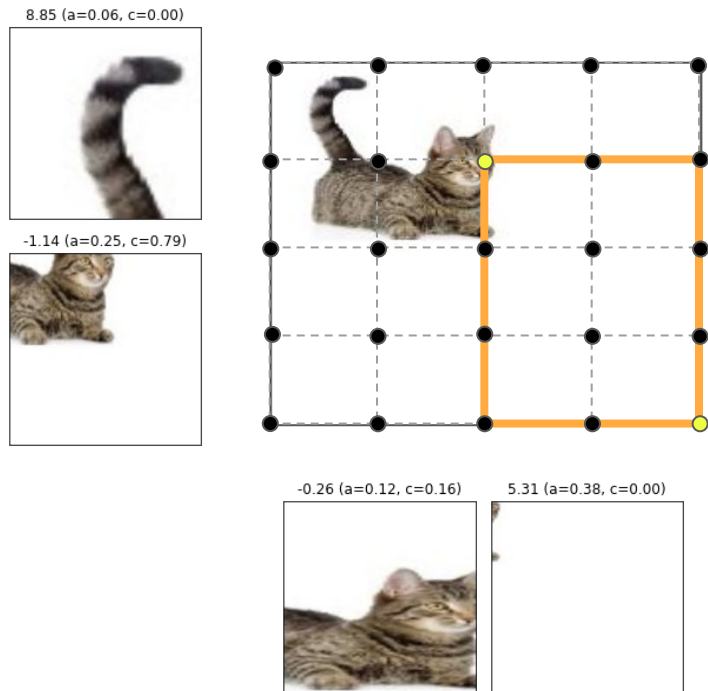
- Discretize the image into subgrids. For ImageNet we divide the image into a 10 x 10 grid.
- Pick any 2 points (to represent opposite corners of a rectangle) and evaluate the cropped region.

# OptBox



- Discretize the image into subgrids. For ImageNet we divide the image into a 10 x 10 grid.
- Pick any 2 points (to represent opposite corners of a rectangle) and evaluate the cropped region.

# OptBox



- Discretize the image into subgrids. For ImageNet we divide the image into a 10 x 10 grid.
- Pick any 2 points (to represent opposite corners of a rectangle) and evaluate the cropped region.

# OptBox

8.85 (a=0.06, c=0.00)



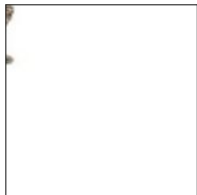
-1.14 (a=0.25, c=0.79)



-0.26 (a=0.12, c=0.16)



5.31 (a=0.38, c=0.00)



- Discretize the image into subgrids. For ImageNet we divide the image into a 10 x 10 grid.
- Pick any 2 points (to represent opposite corners of a rectangle) and evaluate the cropped region.
- Evaluate all possible crops and report the best LSC score.

# Quantitative Analysis: Sparsity and Win%

**Sparsity:** Fraction of Image area which the boolean mask covers.

**Win%:** Fraction of the total images for which a given saliency method achieved the highest score.

# Image Saliency: Quantitative Analysis

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

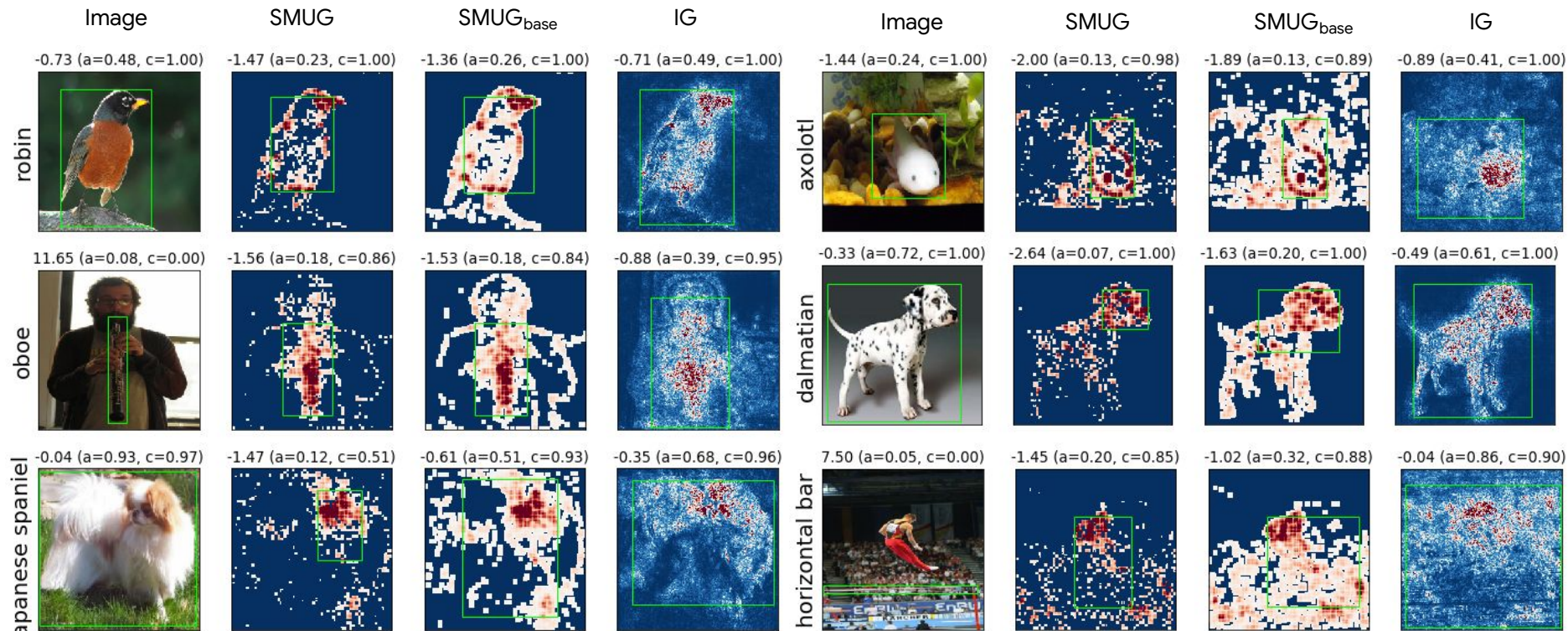
a = crop\_size / image\_size

c = confidence of the classifier on the cropped image

Method	SMUG	SMUG <sub>base</sub>	GROUNDTRUTH	IG	GRADCAM	CENTERBOX	MAXBOX	OPTBOX
<b>LSC</b> ↓	<b>-1.26</b> <sup>-0.75</sup> <sub>-1.80</sub>	<b>-1.23</b> <sup>-0.71</sup> <sub>-1.76</sub>	<b>-0.34</b> <sup>0.04</sup> <sub>-0.81</sub>	<b>-0.29</b> <sup>-0.05</sup> <sub>-0.62</sub>	<b>-1.10</b> <sup>-0.50</sup> <sub>-1.67</sub>	<b>-0.64</b> <sup>-0.29</sup> <sub>-0.69</sub>	<b>0.04</b> <sup>0.23</sup> <sub>0.00</sub>	<b>-2.27</b> <sup>-1.79</sup> <sub>-2.71</sub>
<b>Win%</b> ↑	<b>40.9</b> ± 1.68	33.5 ± 1.61	3.6 ± 0.64	1.7 ± 0.44	37.8 ± 1.65	2.6 ± 0.54	0.2 ± 0.16	-
<b>Sparsity%</b> ↓	<b>17.7</b> ± 0.10	43.3 ± 0.32	50.7 ± 0.98	-	-	50.0±0.99	100.0±0.0	8.9±0.0

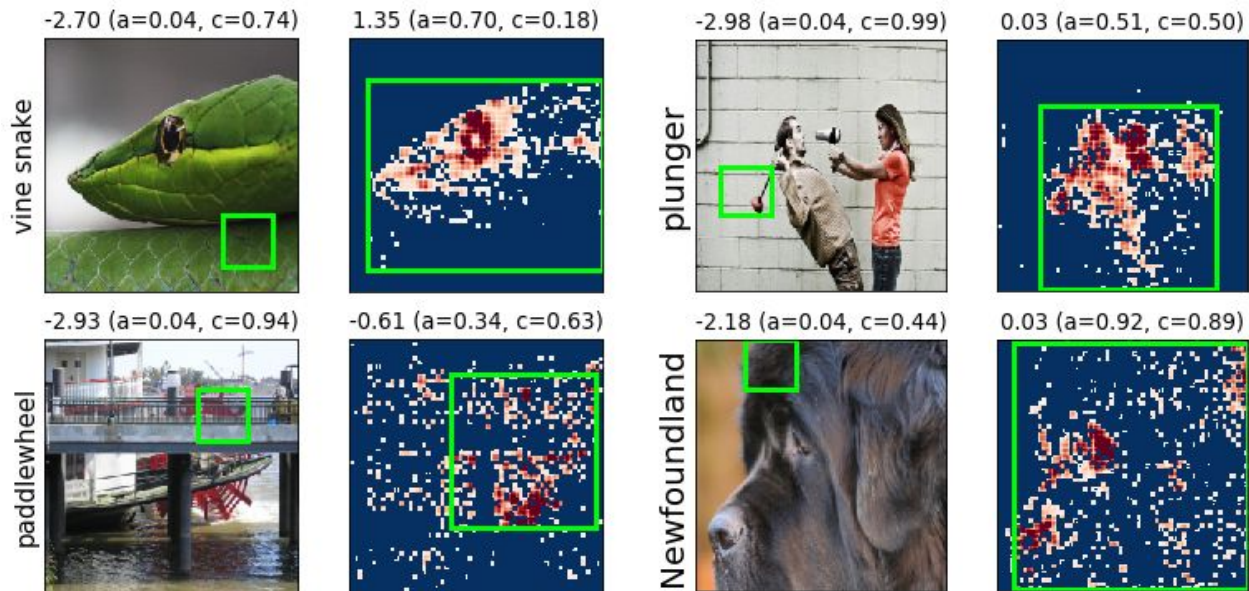
- Scores are reported for 3304 images which were classified correctly in the ImageNet’s Test set.
- SMUG and SMUG\_base achieve similar LSC scores, though the former produces much sparser saliency maps, because the LSC score considers the size of the cropped region and not the sparsity of the saliency map.
- OptBox isn’t faithful to the model and achieves a very high negative score by gaming the LSC score as demonstrated in the 2-cats experiment.

# Image Saliency



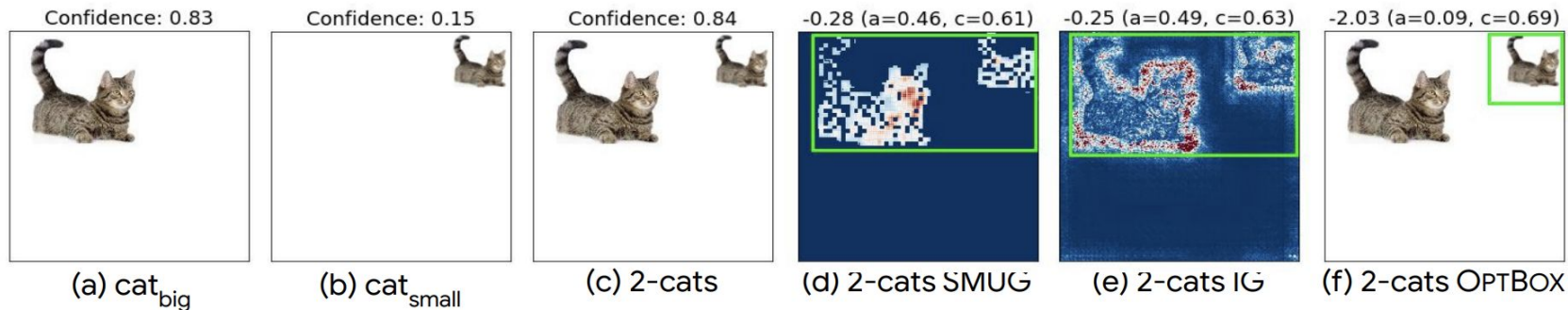


# SMUG vs OptBox



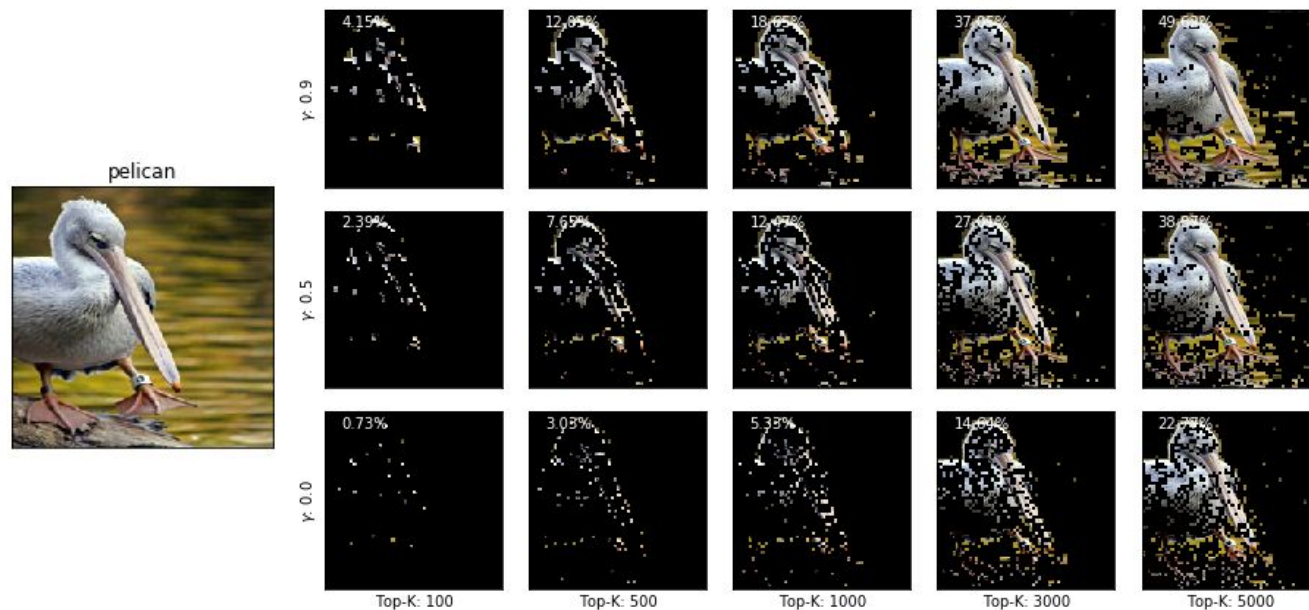


# Analysing LSC Metric and OptBox

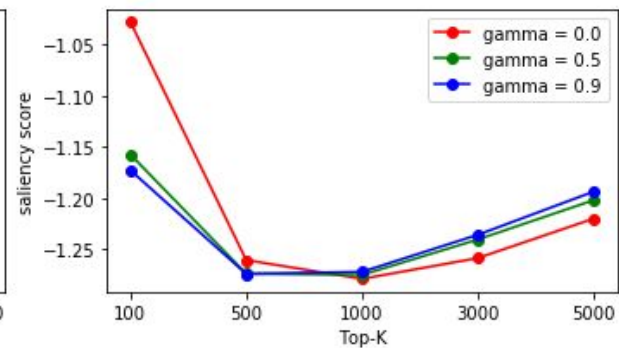
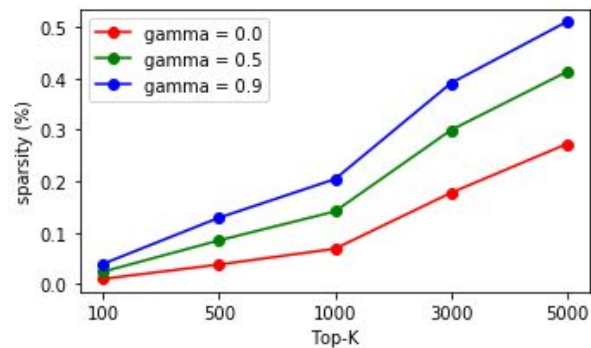
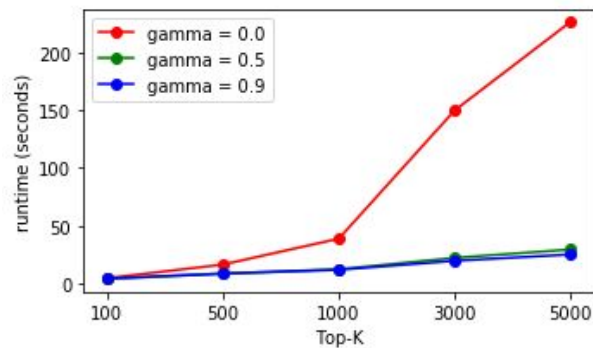


(a) shows an image of a cat ( $\text{cat}_{\text{big}}$ ) placed on a white background that is classified with a confidence of 0.83. (b) shows an image of the same cat ( $\text{cat}_{\text{small}}$ ), scaled to a quarter of its original size, that is classified with a confidence of 0.15. (c) By placing  $\text{cat}_{\text{big}}$  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 object rescaling in LSC, favoring the more compact object.

# TopK vs Gamma



# TopK vs Gamma



# Text Saliency

SIS

11.2 oz bottle split and poured into a new belgium globe . 4.3 % abv , 4°c - 6°c , 40199 on label . a - faintly cloudy pink in color with a dense white head . frilly lace drapes across the glass delicately . § - fresh bushels of raspberries just rinsed gives the aroma an inviting nose . soft wheat and lemonade pull through and 'fake-up' the aroma . it almost begins smelling like a berry weiss-sunset wheat combo . initially good though ... † - fruity raspberry is far too syrupy with a lingering corn syrup finish . light wheat in the background balanced just a bit but it 's overly fake in it 's flavor profile . it almost seems as if lemonade is blended in . m - sugary sweet and slight with a highly carbonated finish and light body . o - overly sweet and fake , it 's got a good aroma initially but definitely lacking elsewhere . too syrupy but not awful .

SMUG

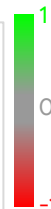
11.2 oz bottle split and poured into a new belgium globe . 4.3 % abv , 4°c - 6°c , 40199 on label . a - faintly cloudy pink in color with a dense white head . frilly lace drapes across the glass delicately . § - fresh bushels of raspberries just rinsed gives the aroma an inviting nose . soft wheat and lemonade pull through and 'fake-up' the aroma . it almost begins smelling like a berry weiss-sunset wheat combo . initially good though ... † - fruity raspberry is far too syrupy with a lingering corn syrup finish . light wheat in the background balanced just a bit but it 's overly fake in it 's flavor profile . it almost seems as if lemonade is blended in . m - sugary sweet and slight with a highly carbonated finish and light body . o - overly sweet and fake , it 's got a good aroma initially but definitely lacking elsewhere . too syrupy but not awful .

SMUG<sub>base</sub>

11.2 oz bottle split and poured into a new belgium globe . 4.3 % abv , 4°c - 6°c , 40199 on label . a - faintly cloudy pink in color with a dense white head . frilly lace drapes across the glass delicately . § - fresh bushels of raspberries just rinsed gives the aroma an inviting nose . soft wheat and lemonade pull through and 'fake-up' the aroma . it almost begins smelling like a berry weiss-sunset wheat combo . initially good though ... † - fruity raspberry is far too syrupy with a lingering corn syrup finish . light wheat in the background balanced just a bit but it 's overly fake in it 's flavor profile . it almost seems as if lemonade is blended in . m - sugary sweet and slight with a highly carbonated finish and light body . o - overly sweet and fake , it 's got a good aroma initially but definitely lacking elsewhere . too syrupy but not awful .

IG

11.2 oz bottle split and poured into a new belgium globe . 4.3 % abv , 4°c - 6°c , 40199 on label . a - faintly cloudy pink in color with a dense white head . frilly lace drapes across the glass delicately . § - fresh bushels of raspberries just rinsed gives the aroma an inviting nose . soft wheat and lemonade pull through and 'fake-up' the aroma . it almost begins smelling like a berry weiss-sunset wheat combo . initially good though ... † - fruity raspberry is far too syrupy with a lingering corn syrup finish . light wheat in the background balanced just a bit but it 's overly fake in it 's flavor profile . it almost seems as if lemonade is blended in . m - sugary sweet and slight with a highly carbonated finish and light body . o - overly sweet and fake , it 's got a good aroma initially but definitely lacking elsewhere . too syrupy but not awful .



## Predict ratings for aroma of a beer from customer reviews

SIS: Sufficient Input Subsets (finds minimal subset such that the final confidence of the prediction is retained)

SMUG: SMT + gradients

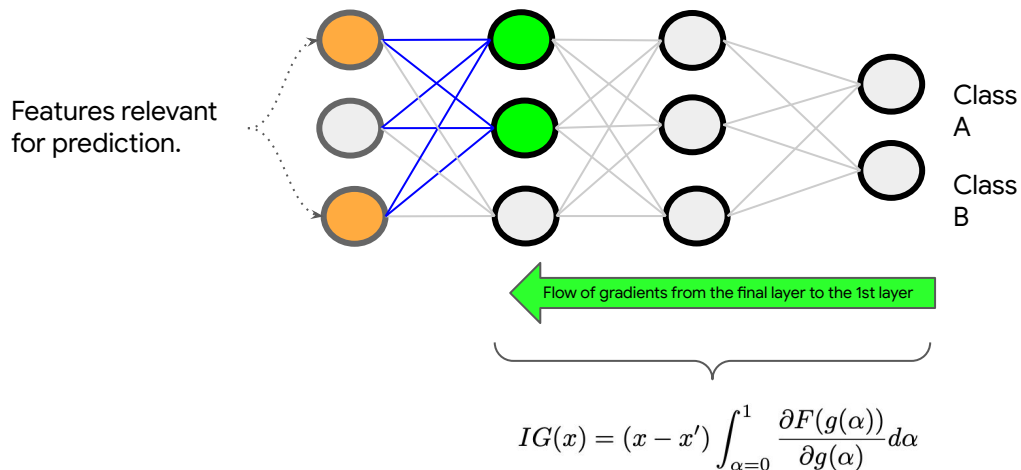
SMUG\_base: SMUG without minimization

IG: Integrated Gradients

Underlined words: Human rationale

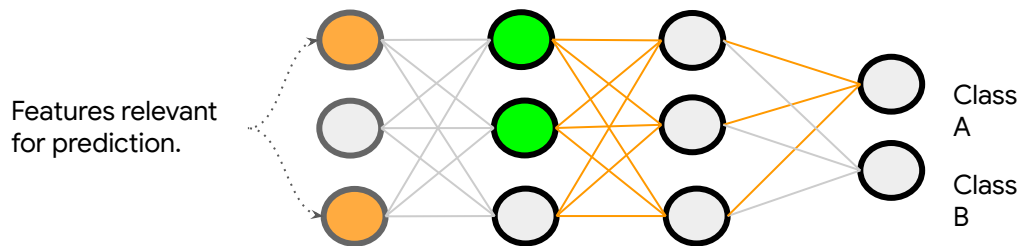
# Summary

- We present a technique (SMUG) to encode the minimal input feature discovery problem for neural model explanation using SMT solvers. Our approach, which does masking on linear equations also overcomes the issue of handling out-of-distribution samples.



# Summary

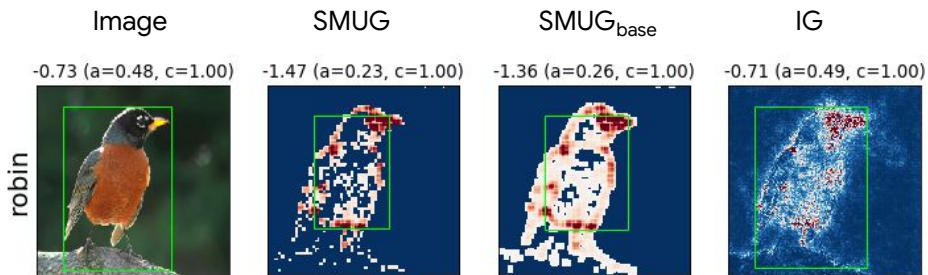
- We present a technique (SMUG) to encode the minimal input feature discovery problem for neural model explanation using SMT solvers. Our approach, which does masking on linear equations also overcomes the issue of handling out-of-distribution samples.
- Our approach uses gradient information to scale SMT-based analysis of neural networks to larger models and input features. Further, it also overcomes the issue of choosing a “baseline” parameter for Integrated Gradients (Kapishnikov et al., 2019; Sturmfels et al., 2020).



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

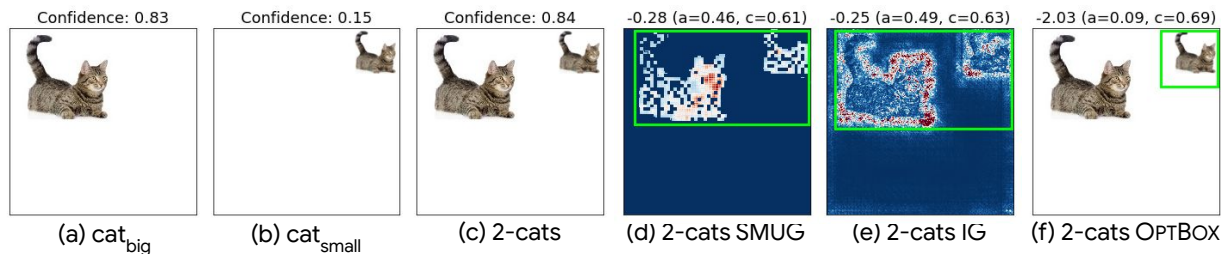
# Summary

- We present a technique (SMUG) to encode the minimal input feature discovery problem for neural model explanation using SMT solvers. Our approach, which does masking on linear equations also overcomes the issue of handling out-of-distribution samples.
- Our approach uses gradient information to scale SMT-based analysis of neural networks to larger models and input features. Further, it also overcomes the issue of choosing a “baseline” parameter for Integrated Gradients (Kapishnikov et al., 2019; Sturmfels et al., 2020).
- 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.



# Summary

- We present a technique (SMUG) to encode the minimal input feature discovery problem for neural model explanation using SMT solvers. Our approach, which does masking on linear equations also overcomes the issue of handling out-of-distribution samples.
- Our approach uses gradient information to scale SMT-based analysis of neural networks to larger models and input features. Further, it also overcomes the issue of choosing a “baseline” parameter for Integrated Gradients (Kapishnikov et al., 2019; Sturmfels et al., 2020).
- 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.
- To improve our understanding on saliency map evaluation, we show how the popular and widely used LSC metric (Dabkowski & Gal, 2017) can be gamed heuristically to generate explanations that are not necessarily faithful to the model.





# Code:

[https://github.com/google-research/google-research/tree/master/smug\\_saliency](https://github.com/google-research/google-research/tree/master/smug_saliency)

```
In [ ]: # Copyright 2021 The Google Research Authors.
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
```

## NOTE

Make sure that this notebook is using smug kernel

we use inception\_v1 model from TF slim here. Our paper used a slightly different variant of this model without batch norm, so visualizations and results may differ. At the bottom, we also show examples for inception\_v3.

```
In [ ]: import os
import matplotlib.pyplot as plt
import numpy as np
from PIL import Image
import saliency
import tensorflow.compat.v1 as tf
import tensorflow_hub as hub
import tf_slim as slim
tf.disable_eager_execution()

if not os.path.exists('models/research/slim'):
    !git clone https://github.com/tensorflow/models/

if not os.path.exists('inception_v1_2016_08_28.tar.gz'):
    !wget http://download.tensorflow.org/models/inception_v1_2016_08_28.tar.gz
    !tar -xvzf inception_v1_2016_08_28.tar.gz

old_cwd = os.getcwd()
os.chdir('models/research/slim')
from nets import inception_v1
os.chdir(old_cwd)

-- -- -- -- --
```