On the Effect of Gradient Regularisation on Interpretability

Arn De Moor[2], Arne Gevaert[1,2], Jonathan Peck[1,2], and Yvan Saeys[1,2]

[1] Department of Mathematics, Computer Science and Statistics, Ghent University
[2] Data Mining and Modeling for Biomedicine, VIB Inflammation Research Center

Abstract. This paper explores the impact of input gradient regularisation on model interpretability. Although this technique has been known for some time to improve the generalisation ability of deep neural networks, it was only recently highlighted that regularising the input gradient may also enhance its interpretability as a saliency map. Prior works have only observed this effect subjectively, however, and a quantitative evaluation is currently lacking. We aim to fill this gap by quantifying the influence of gradient regularisation on the quality of gradient-based saliency maps across multiple metrics and datasets. We find that gradient regularisation can indeed increase the quality of saliency maps, although this effect is heavily dependent on the specific dataset and/or model. This finding implies that subjective observations regarding the quality of saliency maps are not guaranteed to generalise to different datasets.

Keywords: Gradient Regularisation · Saliency Map · Faithfulness · Robustness

1 Introduction

Deep neural networks have become increasingly popular as a solution for various tasks, such as natural language processing [4] and computer vision [32]. However, due to their black box nature, there have been concerns regarding their interpretability [10]. As complicated non-linear functions, it is often difficult or impossible to explain why a deep neural network produced a given output and which factors contributed to it. This is problematic when neural networks are used to make high-stakes decisions, such as approving bank loans or aiding in medical diagnosis. It is therefore crucial that we develop techniques to make neural networks more interpretable.

In this paper, we focus on one specific method known as *gradient regularisation*. Gradient regularisation was first introduced as a means to increase the smoothness and generalisation capacity of neural networks by penalising large input gradients [9]. It has also been used to decrease the susceptibility to adversarial perturbations [12,23]. More recently, it has been discovered that this method might improve interpretability [25]. However, to the best of our knowledge, no work has been done on quantifying this effect using concrete metrics across multiple datasets. In this work, we address this gap in the literature.

Numerous metrics have been proposed to evaluate the interpretability and quality of saliency maps, which are commonly used to explain machine learning models. According to Hedström et al. [15], these metrics can be roughly categorised into six classes: faithfulness, robustness, localisation, complexity, randomisation (sensitivity), and axiomatic metrics. The classes that are most relevant to our work are faithfulness, robustness and complexity.

To quantify faithfulness, we used an estimate inspired by the approach of Alvarez-Melis and Jaakkola [2], which verifies whether perturbing the pixels marked as 'relevant' by the saliency map truly impacts the output. Robustness measures whether the saliency is stable under small perturbations of the input. The local Lipschitz constant [1] and Max-Sensitivity [31] were used to quantify this. The fractional entropy of the pixels in the saliency map was used as a measure of complexity, as suggested by [5]. The impact of gradient regularisation on these metrics was analysed on five different image datasets using two different models and two different techniques to generate saliency maps.

To regularise the gradient, we use *double backpropagation* [9], which analytically computes the second-order derivative of the loss function with respect to the input and the model weights. While alternative methods for input gradient regularisation exist that may offer performance advantages [12], double backpropagation is the exact method that was also used in [25], which originally hypothesised the positive effect of gradient regularisation on the interpretability of saliency maps.

The authors of [25] have already noticed that gradient-based saliency maps for a regularised model on MNIST digits seem to be visually more interpretable. Our quantitative experiments confirm this observation, showing an increased faithfulness on MNIST for regularised models. However, we find that this improvement may not generalise to more complex datasets, as this increase in faithfulness disappears when testing on datasets like CIFAR-10 and ImageNette.

2 Related work

Double backpropagation was first introduced as a means of regularising the input gradient of a neural network [9]. In double backpropagation, a regularisation term is added to the loss function which contains the gradient of the original loss function with respect to the input features. As is the case for other regularisation techniques [26], double backpropagation has been shown to have a positive effect on the generalisation of the model when the correct regularisation strength is used. Despite this advantage, it never gained significant popularity for this purpose compared to other regularisation techniques such as dropout [29], batch normalisation [18] and regular ℓ_2 regularisation. As such, it is only since relatively recently that it has gained prominence, as regularising input gradients has been shown improve robustness to adversarial perturbations [25,12,23]. Furthermore, Ros and Doshi-Velez [25] suggested that this approach would also improve the interpretability of the model. Following this reasoning, some others have applied similar techniques for this purpose [11,22]. However, in these studies, verifica-

tion relied solely on a single dataset, possibly limiting the generalisability of their conclusions. In this work, the effects of input gradient regularisation on the interpretability of saliency maps are investigated across several datasets using multiple quantifiable metrics for a wide range of regularisation strengths. Therefore, we are able to draw more objective and generalisable conclusions.

3 Methods

In this section, we discuss the methods used in our work in more detail. We begin with a description of the experimental setup, datasets, and training procedure. Next we give an overview of the various metrics we used to quantify the quality of saliency maps.

3.1 Experimental Setup

All models were regularised using input gradient regularisation, similar to the approach described in [25]. The total loss function used is defined as follows:

$$\mathcal{L}(y, \hat{y}) = \mathrm{H}(y, \hat{y}) + \alpha \, \|\boldsymbol{\nabla} \, \mathrm{H}(y, \hat{y})\|^2$$

Here, $\mathrm{H}(y, \hat{y})$ is the cross-entropy between the model outputs y and the labels $\hat{y}$, $\alpha > 0$ is a predetermined regularisation coefficient, $\boldsymbol{\nabla}$ represents the gradient with respect to the input and $\|.\|^2$ is the square of the ℓ_2-norm. Similar to ℓ_2 regularisation, this loss function will primarily penalise large gradients. To assess the effect of the regularisation strength on the interpretability, models were trained for a wide range of α values. Next, different metrics were used to assess the quality of saliency maps as a function of α.

ResNet18 [14] models were trained using the CIFAR-10 [20] and Imagenette [17] datasets. LeNet-5 [21] was evaluated on the MNIST digits [8], Fashion-MNIST [30], and Kuzushiji-MNIST (KMNIST) datasets [6]. The Fashion-MNIST, KMNIST and MNIST datasets are very similar in the sense that all three contain 10 classes that are divided over 60000 28x28 grayscale images. The '320 px' version of Imagenette was used and the images were centre-cropped so that the resulting size is 320x320 pixels. Every dataset employed includes a pre-specified separation between training and testing sets.

For the MNIST, Fashion-MNIST, and KMNIST datasets, the data was normalised to the range [-1, 1]. In the case of CIFAR-10 and Imagenette, the data was normalised to the range [0, 1].

For models trained on the MNIST, Fashion-MNIST, and KMNIST datasets, training was performed over 25 epochs with a learning rate of 0.001. Models trained on CIFAR-10 and Imagenette underwent 50 epochs with a learning rate of 0.0001.

The Adam optimiser [19] was used for all tests. For all datasets except Imagenette, 50 models were trained with logarithmically spaced α values between 10^2 and 10^{10}. An exception was made for Imagenette as the accuracy 'tipping

point' is not clearly visible in the aforementioned range. Instead, 27 models were trained with logarithmically spaced α values between 1×10^3 and 1.29×10^{13}.

For every metric, the implementation provided by Quantus 0.5.3 [15] was used, while PyTorch [3] served as the underlying framework for the experiments. The metrics are evaluated on a batch of 128 samples for every trained model.

Two methods were used to create the saliency maps. The first method, calculating the input gradient, is the most simple one. Even though other methods exist that show better performance, it is often used for comparison as done in [11,25]. The second method used is DeepLIFT [27], which is a popular method to generate saliency maps [7].

3.2 Metrics

The metrics we used to quantify the quality of saliency map explanations can be divided into three categories, based on the specific aspect of the saliency map that they measure: faithfulness, robustness or complexity.

Faithfulness The metric used to evaluate the faithfulness of the saliency map was the Faithfulness Estimate (FE) used by [2]. This metric is designed to verify whether features that are marked as relevant by the saliency map truly affect the prediction score to a greater extent. The FE of a saliency map is computed by perturbing each feature individually, and recording the effect on the model output. More concretely, this effect is the difference between the confidence score for the correct class before and after perturbation of the feature. Finally, the correlation between this effect and the importance of the pixels according to the saliency map is calculated. In our case, the perturbation is computed for each feature as follows:

$$x_i' := \min_{1 \le j \le n} x_j + \max_{1 \le j \le n} x_j - x_i$$

where x_i and x_i' are respectively the original and updated values of the i-th feature in the image, and n is the total number of features. We denote the original image as $\mathbf{x}$, and the version of $\mathbf{x}$ where the i-th feature is perturbed in this way as $\mathbf{x}^{(i)}$. The effect of the perturbation on the i-th feature of an image x is then defined as:

$$d_i := f\left(\mathbf{x}^{(i)}\right) - f(\mathbf{x})$$

where $f(\mathbf{x})$ is the confidence score of the model f for image $\mathbf{x}$. Denoting the function that generates the attribution map for a given image $\mathbf{x}$ as $a(\mathbf{x})$, we compute the FE for an image $\mathbf{x}$ and attribution map $\mathbf{a} := a(\mathbf{x})$ as the Pearson correlation coefficient between the effects $(d_i \mid 1 \le i \le n)$ and attribution values $(a_i \mid 1 \le i \le n)$.

In order to assess whether the FE metric is actually increasing with α, we compute the p-values of the Spearman correlation test between the α values and the corresponding FE scores. The p-values are calculated using a two-sided permutation test with 9999 samples, with a conservative correction to account

for sampling uncertainty [24]. Note that the accuracy of the model starts decreasing past a certain value for α. At this point, the model can be considered over-regularised. We exclude these over-regularised models from the p-value calculation by choosing a threshold value t_α for each dataset and excluding all models with $\alpha > t_\alpha$. This threshold is defined as the minimal value such that $t_\alpha > 1e5$ and the accuracy of the model with $\alpha = t_\alpha$ is lower than 80% of the mean accuracy of all models with $\alpha < 1e5$.

Robustness The first metric used to evaluate the robustness of the saliency maps was the local Lipschitz constant (also used as a robustness metric in [1]). Models with a lower score are more robust to input perturbations [16]. In this work, the Quantus implementation [15] of this metric was used:

$$\sup_{\mathbf{x} \in B} \frac{\|a(\mathbf{x} + \epsilon) - a(\mathbf{x})\|_2}{\|\epsilon\|_2}$$

Where B is a batch of 200 random input samples, $a(\mathbf{x})$ is the function that generates the attribution map based on the input $\mathbf{x}$ and $\epsilon \sim \mathcal{N}(0, 0.1)$ is Gaussian noise centred around 0 and a standard deviation of 0.1 sampled for every pixel. Note that, as previously mentioned, for every trained model, this formula is evaluated 128 times, and the average is taken.

An additional robustness measure we used in our experiments is Max-Sensitivity [31]. It is defined by the following formula:

$$\max_{\|\epsilon\| \leq r} \|a(\mathbf{x} + \epsilon) - a(\mathbf{x})\|$$

This value is estimated by sampling the noise ϵ uniformly in the range $[-0.2, 0.2]$. Thus, in this case, the norm for $\|\epsilon\|$ is the L_∞ norm and $r = 0.2$. The norm for $\|a(\mathbf{x} + \epsilon) - a(\mathbf{x})\|$ was chosen to be the ℓ_2 norm. As done in [31], the sensitivity was normalised to allow for comparison. In our case, the sensitivity was divided by the ℓ_2 norm of the original saliency map $a(\mathbf{x})$. A batch of 128 samples was used to estimate the average max-sensitivity for a trained model and 200 noise samples were used to estimate the maximum in the max-sensitivity definition.

Complexity We describe an explanation as having low complexity if it highlights only a small fraction of the features as being important. This is quantified using the approach described in [5]. We first define a probability distribution based on the contributions of each of the input features:

$$\mathbb{P}(i) := \frac{|a_i|}{\sum_{1 \leq j \leq n} |a_j|}$$

If this probability distribution resembles a uniform distribution, then the explanation is complex. This can be quantified using the entropy of the distribution, where a lower entropy value corresponds to a less complex saliency map:

$$-\sum_{1\leq i\leq n} \mathbb{P}(i)\ln\mathbb{P}(i)$$

This result is averaged over a batch of 128 input samples.

4 Results

In this section, we discuss the results obtained in our experiments. We focus on the results on the MNIST, CIFAR-10 and Imagenette datasets, and using the input gradient to generate saliency maps. Results for the Fashion-MNIST and Kuzushiji-MNIST were very similar to those for MNIST (see Appendix A), and can be retrieved and reproduced from our code repository,[3] as well as the results obtained using DeepLIFT. We also include visual examples demonstrating the effects of gradient regularisation on saliency maps in Appendix B.

We begin by inspecting the effect of double backpropagation on the test accuracy of the model. The results can be viewed in Figure 1. For each dataset, we observe a clear cut-off point where the model is over-regularised and accuracy drops dramatically. Explanations that were generated for models past this cut-off point are therefore considered irrelevant, as the model itself for which the explanation was generated does not generalise.

In Figure 2, the FE for different values of α can be seen for MNIST, CIFAR-10 and Imagenette. The error bars represent the 95% confidence interval of the average FE score computed on a batch of 128 samples. To objectively quantify the effect of double backpropagation on the FE score, we compute the Pearson correlation between the logarithmically spaced α-values and the resulting FE score for each dataset. The resulting correlations and p-values are reported in the corresponding figure captions.

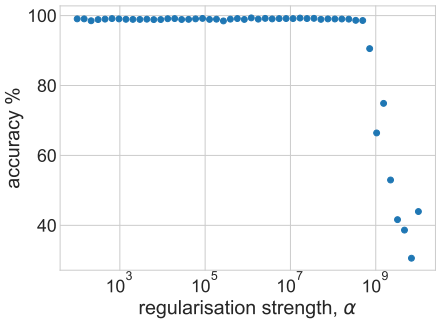
On MNIST, we see a notable improvement in FE score with increasing α, which is in accordance with the observations made visually in [25]. Once the model becomes over-regularised, FE drops again. However, this effect is much less visible on CIFAR-10 and Imagenette, suggesting that the effect on MNIST might not generalise to other, more complex datasets.

As noted in previous research [23,25,12], neural networks trained with input gradient regularisation tend to be more robust to adversarial perturbations. In Figure 3, we see that input gradient regularisation indeed results in a measurable improvement on the local Lipschitz constant and Max-Sensitivity of the explanations, although the magnitude of the effect again varies across datasets.

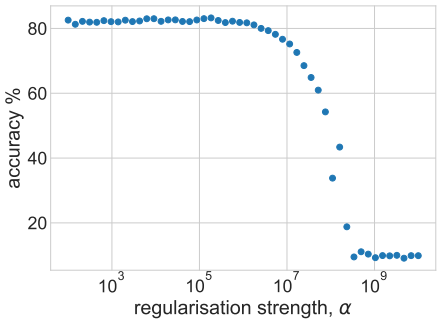
Finally, we investigate the complexity of explanations for increasing values of α. The results are shown in Figure 4. In contrast with the previous metrics, the effect of input gradient regularisation on the complexity does not seem to follow a clear trend. A slight decrease in complexity can generally be observed when using ResNet, although this decrease is not visible when using LeNet.

Additional results on a wider variety of datasets and models can be found in Appendix A.

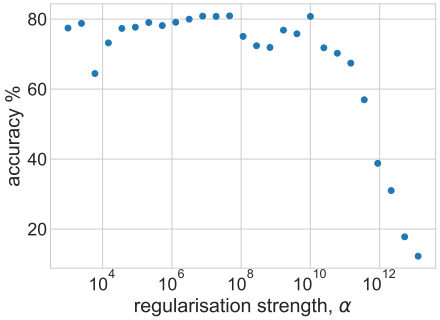
[3] `https://github.com/saeyslab/gradient-regularisation-interpretability`



(a) MNIST (LeNet).



(b) CIFAR-10 (ResNet18).



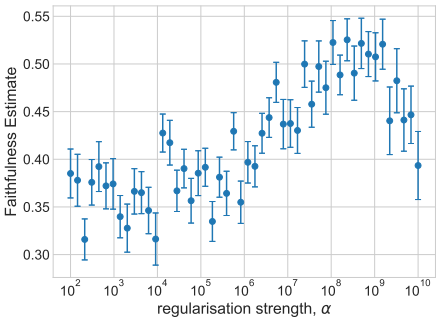
(c) ImageNette (ResNet18).

Fig. 1: Test set accuracy for varying values of α. Each dot corresponds to a separately trained model.

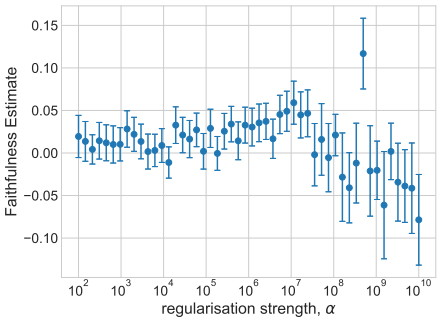
5 Conclusion

We have been able to quantitatively confirm the observation that input gradient regularisation improves the faithfulness of saliency maps on the MNIST dataset. However, we show that this effect does not seem to generalise to more complex datasets such as CIFAR-10 and ImageNette. Additionally, when focusing on different aspects of the quality of saliency, such as robustness and complexity, our results show that the effect of gradient regularisation is dependent on the specific dataset and/or model. Again, the effect seems to be strongly dependent on the dataset in question. We conclude that subjective observations regarding the quality of saliency maps are not guaranteed to generalise to different datasets.

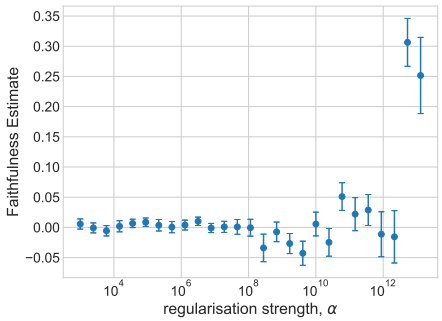
We hypothesise that the differences in results between datasets can be attributed to differences in input dimensionality. The Faithfulness metric, for example, perturbs each pixel individually. However, with increasing image resolution, a single pixel naturally has a diminishing influence on model outputs. Hence, the Faithfulness estimate is susceptible to a curse of dimensionality where increasing resolution causes the metric to be dominated by noise. Alternatively,



(a) MNIST (LeNet).
$\rho = 0.813, p = 0.0002$



(b) CIFAR-10 (ResNet18).
$\rho = 0.536, p = 0.0014$

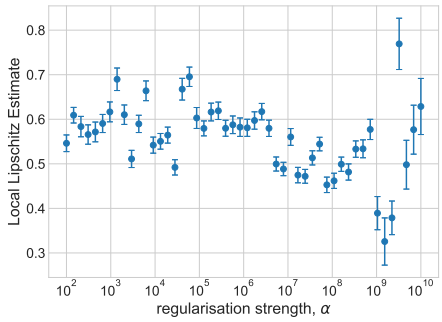


(c) Imagenette (ResNet18).
$\rho = 0.012, p = 0.9416$

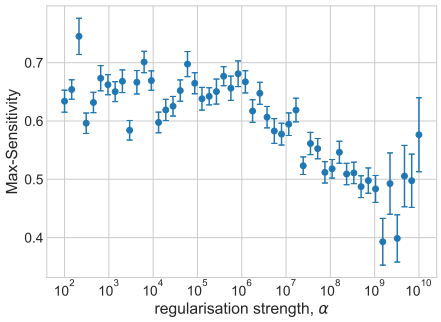
Fig. 2: Faithfulness Estimate for different datasets and varying values of α. Each dot corresponds to a separately trained model. The error bars represent the 95% confidence interval of the Faithfulness Estimate computed on 128 images. For each dataset, ρ and p are the Pearson correlation and corresponding p-value between α and the Faithfulness Estimate.

previous work has exhibited a similar problem for input gradients, which tend to become increasingly noisy in high-dimensional settings [13,28]. This suggests that pixel-based saliency mapping may itself be an inherently more difficult problem in high-dimensional settings. However, more research is needed to clarify the precise mechanisms underlying our observations.

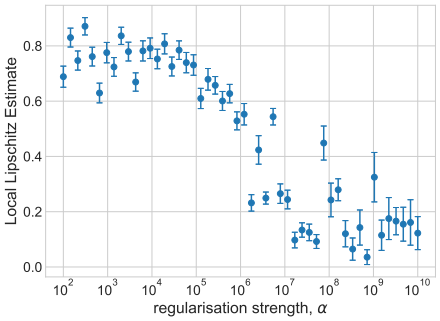
We argue that the quality of saliency maps should be quantified on a case-by-case basis, by computing the metrics of interest on the specific dataset and model of interest. Although we have shown that the influence of gradient regularisation on the quality of saliency maps depends on the specific combination of model and dataset, further research can be done to investigate which components of a given use case are most influential on this effect.



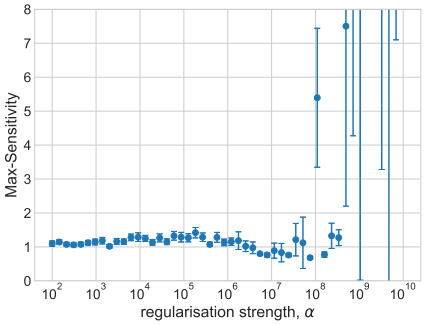
(a) Local Lipschitz constant MNIST (LeNet).
$\rho = -0.548, p = 0.0002$



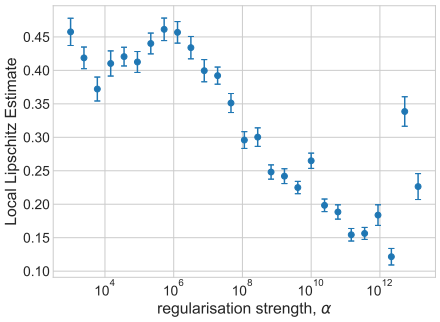
(b) Max-Sensitivity MNIST (LeNet).
$\rho = -0.705, p = 0.0002$



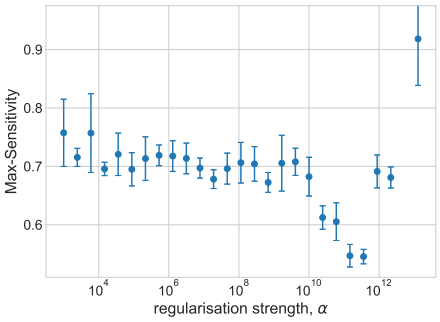
(c) Local Lipschitz constant CIFAR-10 (ResNet18).
$\rho = -0.824, p = 0.0002$



(d) Max-Sensitivity CIFAR-10 (ResNet18).
$\rho = -0.180, p = 0.2982$

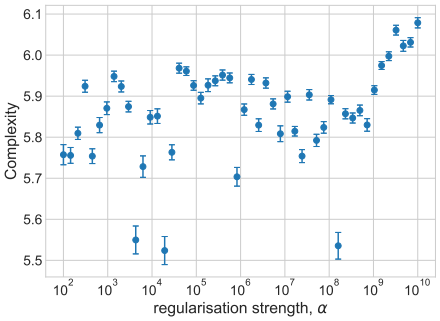


(e) Local Lipschitz constant Imagenette (ResNet18).
$\rho = -0.865, p = 0.0002$

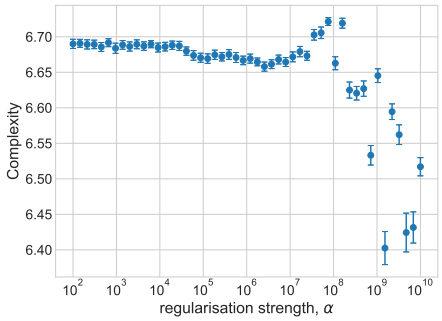


(f) Max-Sensitivity Imagenette (ResNet18).
$\rho = -0.763, p = 0.0002$

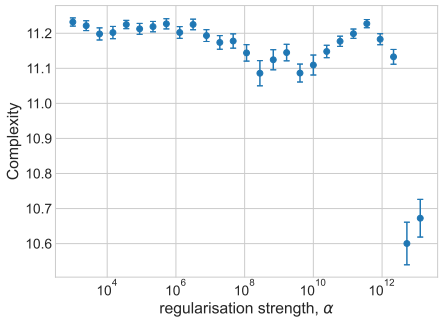
Fig. 3: Local Lipschitz constant and Max-Sensitivity for different datasets and varying values of α. Each dot corresponds to a separately trained model. The error bars represent the 95% confidence interval of the corresponding metric computed on 128 images.



(a) MNIST (LeNet).
$\rho = 0.049, p = 0.749$



(b) CIFAR-10 (ResNet18).
$\rho = -0.687, p = 0.0002$



(c) Imagenette (ResNet18).
$\rho = -0.530, p = 0.011$

Fig. 4: Entropy for different datasets and varying values of α. Each dot corresponds to a separately trained model. The error bars represent the 95% confidence interval of the entropy computed on 128 images.

References

1. Alvarez-Melis, D., Jaakkola, T.S.: On the robustness of interpretability methods (2018), https://arxiv.org/abs/1806.08049
2. Alvarez-Melis, D., Jaakkola, T.S.: Towards robust interpretability with self-explaining neural networks. p. 7786–7795. NIPS'18, Curran Associates Inc., Red Hook, NY, USA (2018)
3. Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., Hirsh, B., Huang, S., Kalambarkar, K., Kirsch, L., Lazos, M., Lezcano, M., Liang, Y., Liang, J., Lu, Y., Luk, C., Maher, B., Pan, Y., Puhrsch, C., Reso, M., Saroufim, M., Siraichi, M.Y., Suk, H., Suo, M., Tillet, P., Wang, E., Wang, X., Wen, W., Zhang, S., Zhao, X., Zhou, K., Zou, R., Mathews, A., Chanan, G., Wu, P., Chintala, S.: PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In: 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24). ACM (Apr 2024). https://doi.org/10.1145/3620665.3640366, https://pytorch.org/assets/pytorch2-2.pdf
4. Arkhangelskaya, E., Nikolenko, S.I.: Deep learning for natural language processing: a survey. Journal of Mathematical Sciences **273**(4), 533–582 (2023)
5. Bhatt, U., Weller, A., Moura, J.M.F.: Evaluating and aggregating feature-based model explanations. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. IJCAI'20 (2021), https://dl.acm.org/doi/abs/10.5555/3491440.3491857
6. Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Kazuaki, Y., Ha, D.: Deep learning for classical Japanese literature (2018). https://doi.org/10.20676/00000341, https://arxiv.org/abs/1812.01718
7. Das, A., Rad, P.: Opportunities and challenges in explainable artificial intelligence (xai): A survey (2020), https://arxiv.org/abs/2006.11371
8. Deng, L.: The MNIST database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine **29**(6), 141–142 (2012). https://doi.org/10.1109/MSP.2012.2211477
9. Drucker, H., Le Cun, Y.: Improving generalization performance using double backpropagation. IEEE Transactions on Neural Networks **3**(6), 991–997 (1992). https://doi.org/10.1109/72.165600
10. Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., Qian, B., Wen, Z., Shah, T., Morgan, G., et al.: Explainable AI (XAI): Core ideas, techniques, and solutions. ACM Computing Surveys **55**(9), 1–33 (2023)
11. Figueroa, F.T., Zhang, H., Sicre, R., Avrithis, Y., Ayache, S.: A learning paradigm for interpretable gradients (2024), https://arxiv.org/abs/2404.15024
12. Finlay, C., Oberman, A.M.: Scaleable input gradient regularization for adversarial robustness. Machine Learning with Applications **3**, 100017 (2021). https://doi.org/https://doi.org/10.1016/j.mlwa.2020.100017, https://www.sciencedirect.com/science/article/pii/S2666827020300177
13. Ghorbani, A., Abid, A., Zou, J.: Interpretation of neural networks is fragile. Proceedings of the AAAI Conference on Artificial Intelligence **33**(01), 3681–3688 (Jul 2019). https://doi.org/10.1609/aaai.v33i01.33013681, https://ojs.aaai.org/index.php/AAAI/article/view/4252

14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
15. Hedström, A., Weber, L., Krakowczyk, D., Bareeva, D., Motzkus, F., Samek, W., Lapuschkin, S., Höhne, M.M.C.: Quantus: An explainable AI toolkit for responsible evaluation of neural network explanations and beyond. Journal of Machine Learning Research **24**(34), 1–11 (2023), http://jmlr.org/papers/v24/22-0142.html
16. Hein, M., Andriushchenko, M.: Formal guarantees on the robustness of a classifier against adversarial manipulation. Advances in neural information processing systems **30** (2017)
17. Howard, J.: Imagenette. https://github.com/fastai/imagenette (2019), accessed: 2024-09-22
18. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. p. 448–456. ICML'15, JMLR.org (2015), https://dl.acm.org/doi/10.5555/3045118.3045167
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980
20. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. Rep. 0, University of Toronto, Toronto, Ontario (2009), https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
21. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
22. Liu, D., Wu, L.Y., Li, B., Boussaid, F., Bennamoun, M., Xie, X., Liang, C.: Jacobian norm with selective input gradient regularization for interpretable adversarial defense. Pattern Recognition **145**, 109902 (2024). https://doi.org/https://doi.org/10.1016/j.patcog.2023.109902, https://www.sciencedirect.com/science/article/pii/S0031320323006003
23. Lyu, C., Huang, K., Liang, H.N.: A unified gradient regularization family for adversarial examples. In: 2015 IEEE International Conference on Data Mining. pp. 301–309 (Nov 2015). https://doi.org/10.1109/ICDM.2015.84
24. Phipson, B., Smyth, G.K.: Permutation p-values should never be zero: Calculating exact p-values when permutations are randomly drawn. Statistical Applications in Genetics and Molecular Biology **9**(1) (2010). https://doi.org/doi:10.2202/1544-6115.1585, https://doi.org/10.2202/1544-6115.1585
25. Ross, A.S., Doshi-Velez, F.: Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence. AAAI'18/IAAI'18/EAAI'18, AAAI Press (2018), https://dl.acm.org/doi/10.5555/3504035.3504238
26. Santos, C.F.G.D., Papa, J.a.P.: Avoiding overfitting: A survey on regularization methods for convolutional neural networks. ACM Comput. Surv. **54**(10s) (Sep 2022). https://doi.org/10.1145/3510413, https://doi.org/10.1145/3510413

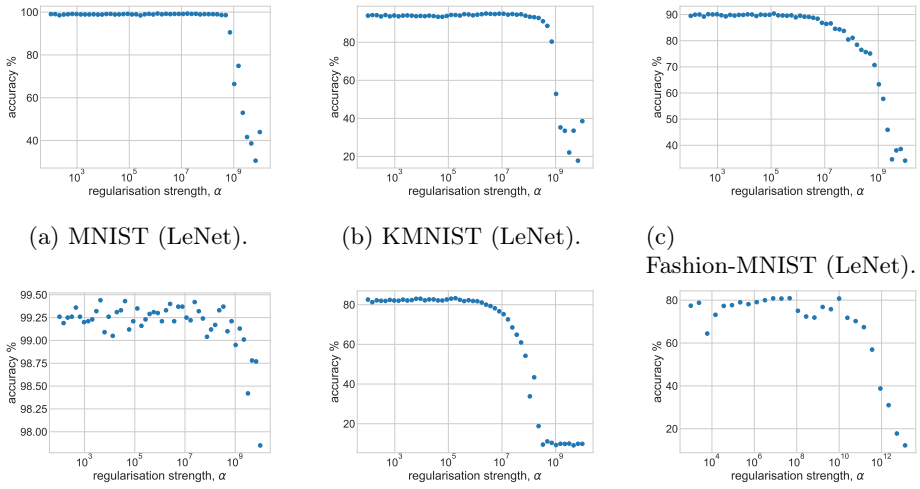
27. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. p. 3145–3153. ICML'17, JMLR.org (2017), https://dl.acm.org/doi/10.5555/3305890.3306006
28. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017)
29. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**, 1929–1958 (2014), https://api.semanticscholar.org/CorpusID:6844431
30. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms (2017), https://arxiv.org/abs/1708.07747
31. Yeh, C.K., Hsieh, C.Y., Suggala, A.S., Inouye, D.I., Ravikumar, P.: On the (in)fidelity and sensitivity of explanations. Curran Associates Inc., Red Hook, NY, USA (2019), https://dl.acm.org/doi/abs/10.5555/3454287.3455271
32. Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., Parmar, M.: A review of convolutional neural networks in computer vision. Artificial Intelligence Review **57**(4), 99 (2024)

A Additional data

To avoid clutter, the major part of the supporting data is not shown in the main report. We provide a more comprehensive view here. This includes additional datasets such as Fashion-MNIST and KMNIST, as well as alternative model-datasets combinations like MNIST using ResNet18. For all combinations, we report results using both gradient saliency maps and DeepLift. Overall, the choice between gradient saliency maps and DeepLift does not appear to significantly affect the observed trends across datasets and models. The main discussion focuses on a subset of the results for clarity. However, it is consistent with the broader set of data presented here, as the analysis used all results.

A.1 Accuracy

In this subsection the accuracy of all trained models is shown.

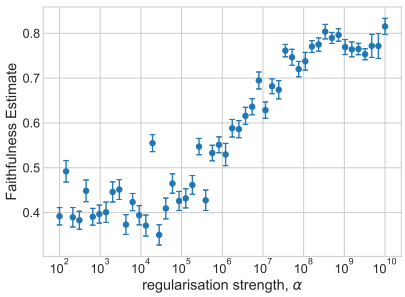


(a) MNIST (LeNet). (b) KMNIST (LeNet). (c) Fashion-MNIST (LeNet).

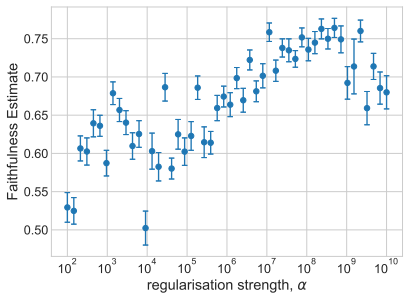
(d) MNIST (ResNet18). (e) CIFAR-10 (ResNet18). (f) ImageNette (ResNet18).

Fig. 5: Test set accuracy for varying values of α. Each dot corresponds to a separately trained model.

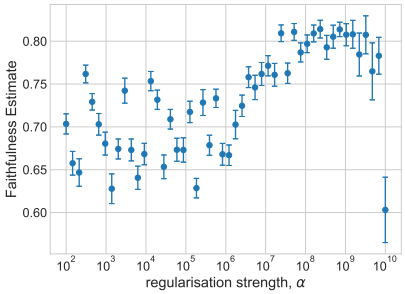
A.2 Faithfulness



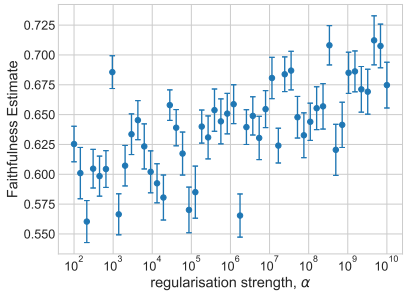
(a) Fashion-MNIST (LeNet).
$\rho = 0.887, p = 0.0002$



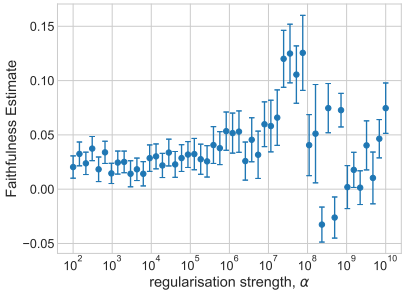
(b) KMNIST (LeNet).
$\rho = 0.834, p = 0.0002$



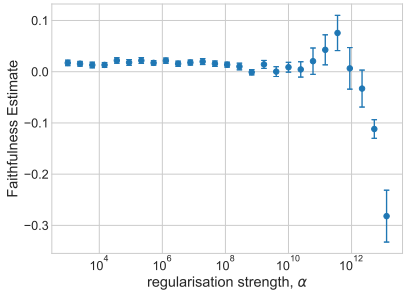
(c) MNIST (LeNet)
$\rho = 0.723, p = 0.0002$



(d) MNIST (ResNet).
$\rho = 0.670, p = 0.0002$

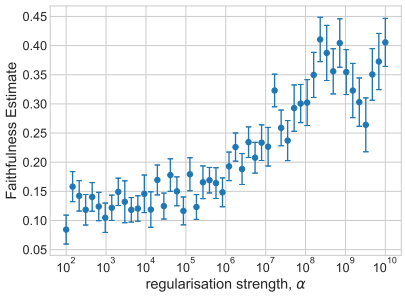


(e) CIFAR-10 (ResNet).
$\rho = 0.701, p = 0.0002$

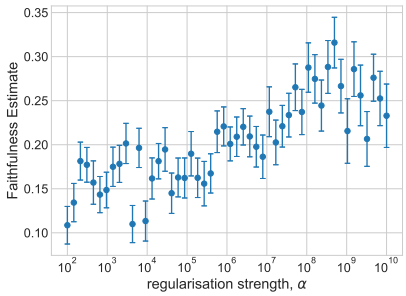


(f) Imagenette (ResNet).
$\rho = -0.054, p = 0.8076$

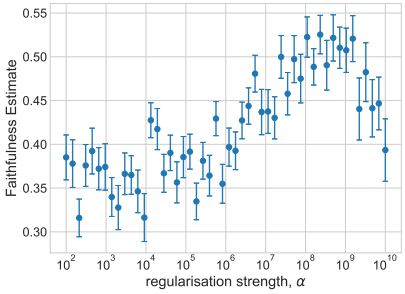
Fig. 6: Faithfulness Estimate for different datasets and varying values of α. Each dot corresponds to a separately trained model. The error bars represent the 95% confidence interval of the Faithfulness Estimate computed on 128 images. Saliency maps were calculated using DeepLift.



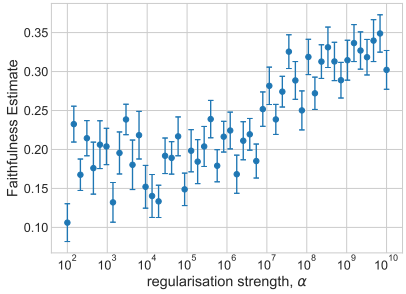
(a) Fashion-MNIST (LeNet).
$\rho = 0.890, p = 0.0002$



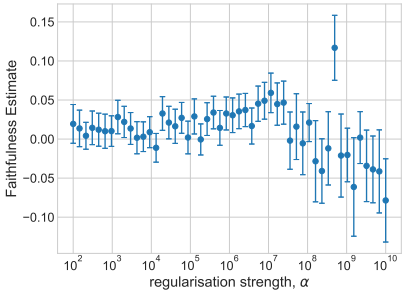
(b) KMNIST (LeNet).
$\rho = 0.839, p = 0.0002$



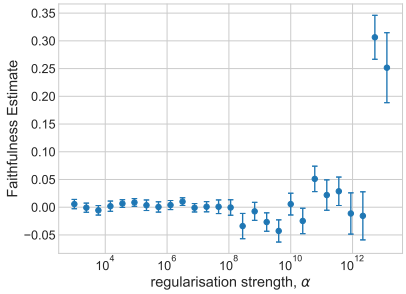
(c) MNIST (LeNet).
$\rho = 0.813, p = 0.0002$



(d) MNIST (ResNet).
$\rho = 0.808, p = 0.0002$



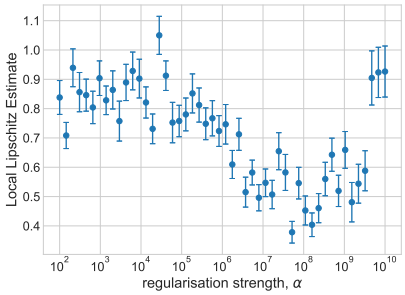
(e) CIFAR-10 (ResNet).
$\rho = 0.536, p = 0.0014$



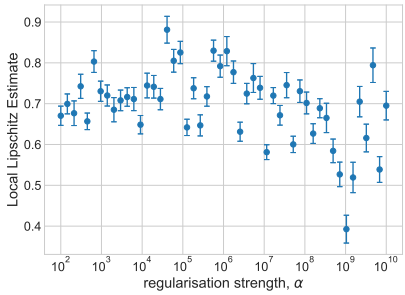
(f) Imagenette (ResNet).
$\rho = 0.012, p = 0.9416$

Fig. 7: Faithfulness Estimate for different datasets and varying values of α. Each dot corresponds to a separately trained model. The error bars represent the 95% confidence interval of the Faithfulness Estimate computed on 128 images. Saliency maps were calculated using the gradient.

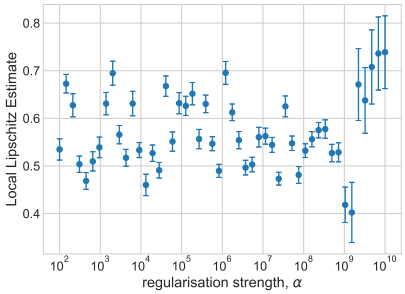
A.3 Robustness



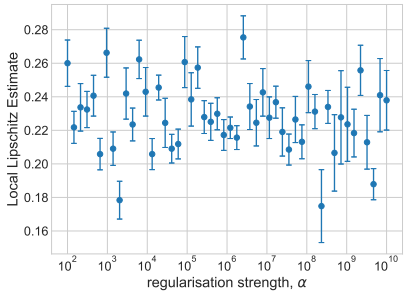
(a) Fashion-MNIST (LeNet).
$\rho = -0.813, p = 0.0002$



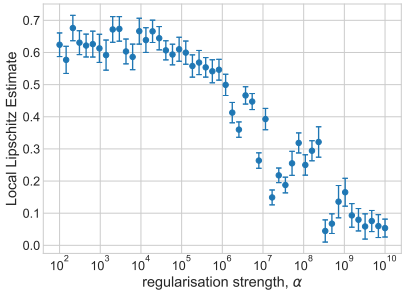
(b) KMNIST (LeNet).
$\rho = -0.267, p = 0.0812$



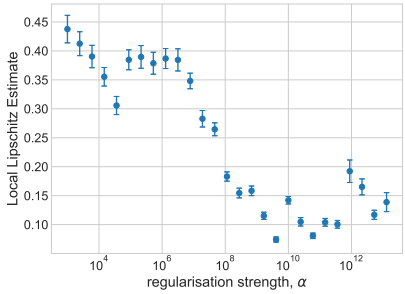
(c) MNIST (LeNet).
$\rho = -0.155, p = 0.3286$



(d) MNIST (ResNet).
$\rho = -0.135, p = 0.3496$

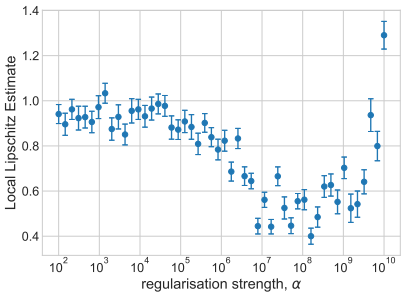


(e) CIFAR-10 (ResNet).
$\rho = -0.834, p = 0.0002$

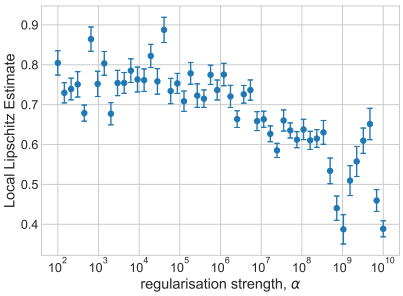


(f) Imagenette (ResNet).
$\rho = -0.932, p = 0.0002$

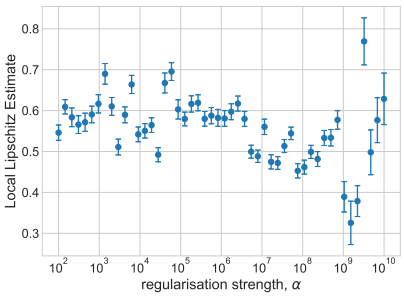
Fig. 8: Local Lipschitz constant for different datasets and varying values of α. Each dot corresponds to a separately trained model. The error bars represent the 95% confidence interval of the corresponding metric computed on 128 images. Saliency maps were calculated using DeepLift.



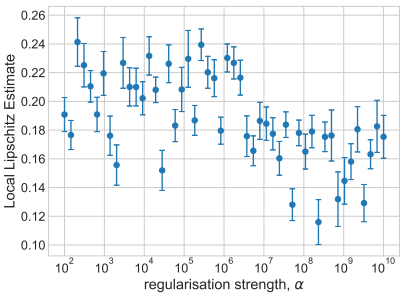
(a) Fashion-MNIST (LeNet).
$\rho = -0.851, p = 0.0002$



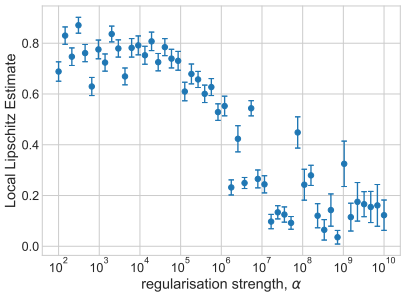
(b) KMNIST (LeNet).
$\rho = -0.755, p = 0.0002$



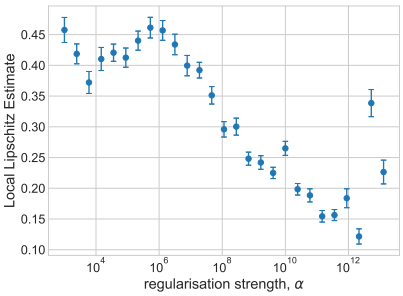
(c) MNIST (LeNet).
$\rho = -0.548, p = 0.0002$



(d) MNIST (ResNet).
$\rho = -0.575, p = 0.0002$

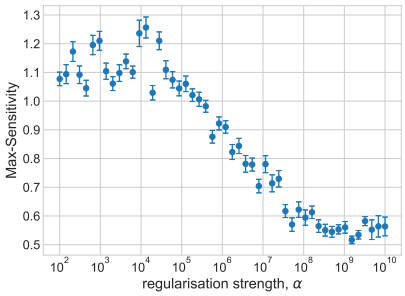


(e) CIFAR-10 (ResNet).
$\rho = -0.824, p = 0.0002$

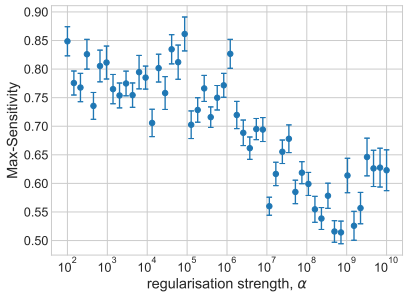


(f) Imagenette (ResNet).
$\rho = -0.865, p = 0.0002$

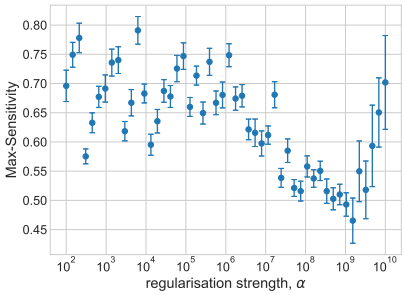
Fig. 9: Local Lipschitz constant for different datasets and varying values of α. Each dot corresponds to a separately trained model. The error bars represent the 95% confidence interval of the corresponding metric computed on 128 images. Saliency maps were calculated using the gradient.



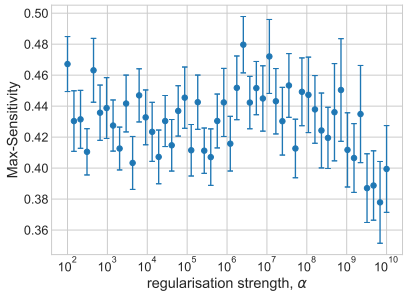
(a) Fashion-MNIST (LeNet).
$\rho = -0.906, p = 0.0002$



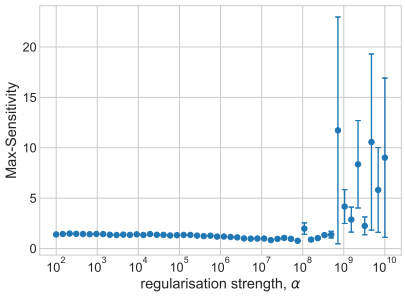
(b) KMNIST (LeNet).
$\rho = -0.818, p = 0.0002$



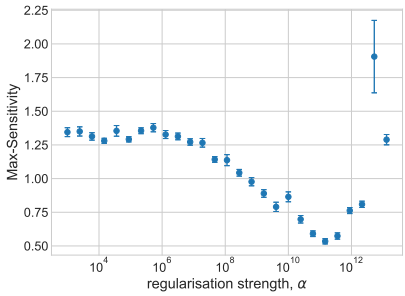
(c) MNIST (LeNet).
$\rho = -0.688, p = 0.0002$



(d) MNIST (ResNet).
$\rho = -0.155, p = 0.2756$

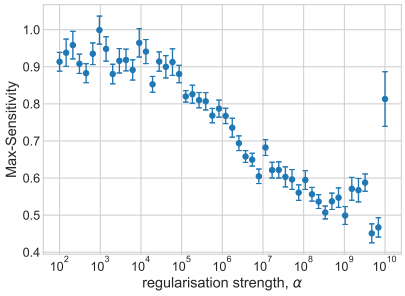


(e) CIFAR-10 (ResNet), Some datapoints with $\alpha > 1e8$ were clipped as the y-value was too high.
$\rho = -0.939, p = 0.0002$

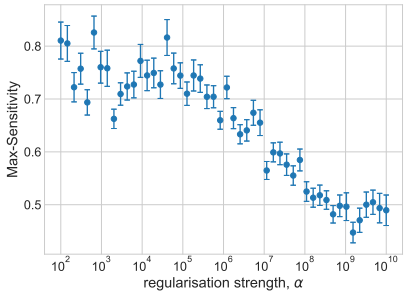


(f) Imagenette (ResNet).
$\rho = -0.913, p = 0.0002$

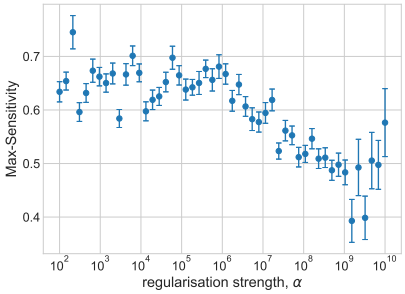
Fig. 10: Max-Sensitivity for different datasets and varying values of α. Each dot corresponds to a separately trained model. The error bars represent the 95% confidence interval of the corresponding metric computed on 128 images. Saliency maps were calculated using DeepLift.



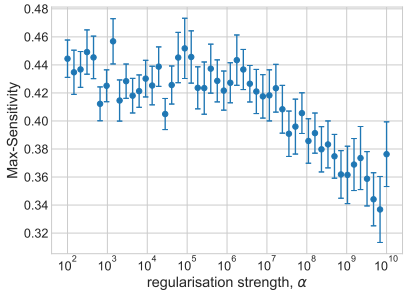
(a) Fashion-MNIST (LeNet).
$\rho = -0.939, p = 0.0002$



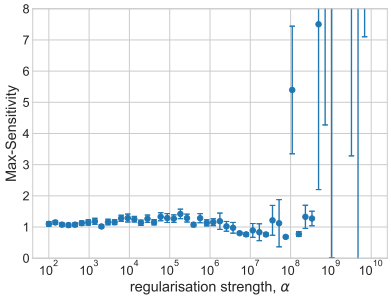
(b) KMNIST (LeNet).
$\rho = -0.868, p = 0.0002$



(c) MNIST (LeNet).
$\rho = -0.705, p = 0.0002$

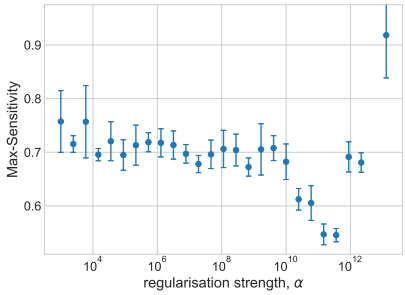


(d) MNIST (ResNet).
$\rho = -0.779, p = 0.0002$



(e) CIFAR-10 (ResNet), Some datapoints with $\alpha > 1e8$ were clipped as the y-value was too high.

$\rho = -0.180, p = 0.2982$

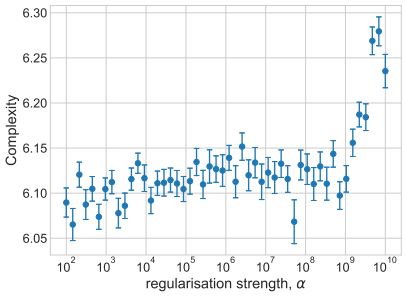


(f) Imagenette (ResNet), a datapoint with $\alpha = 5.275e12$ was clipped as the y-value, 2.25, was too high.

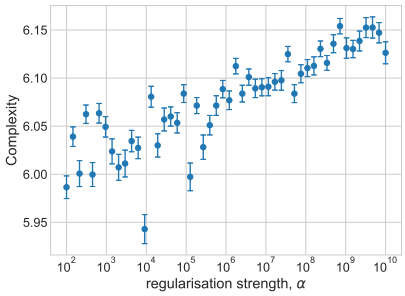
$\rho = -0.763, p = 0.0002$

Fig. 11: Max-Sensitivity for different datasets and varying values of α. Each dot corresponds to a separately trained model. The error bars represent the 95% confidence interval of the corresponding metric computed on 128 images. Saliency maps were calculated using the gradient.

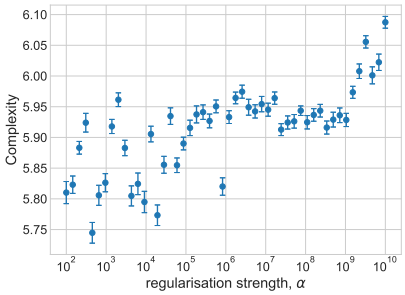
A.4 Complexity



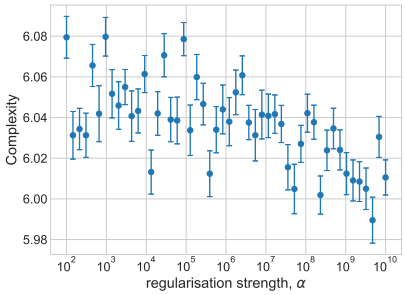
(a) Fashion-MNIST (LeNet).
$\rho = 0.448, p = 0.0032$



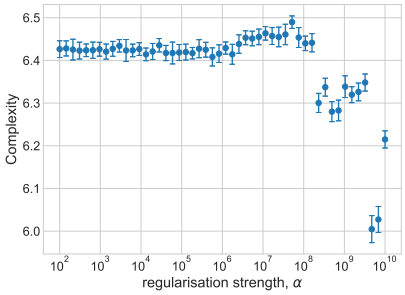
(b) KMNIST (LeNet).
$\rho = 0.876, p = 0.0002$



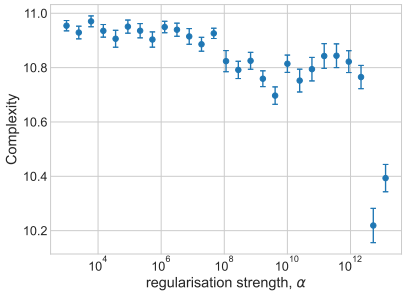
(c) MNIST (LeNet).
$\rho = 0.581, p = 0.0002$



(d) MNIST (ResNet).
$\rho = -0.624, p = 0.0002$

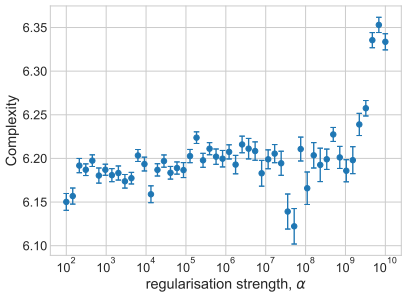


(e) CIFAR-10 (ResNet).
$\rho = 0.355, p = 0.0344$

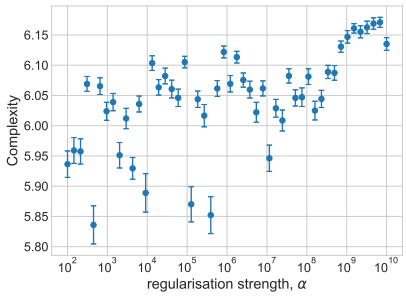


(f) Imagenette (ResNet).
$\rho = -0.805, p = 0.0002$

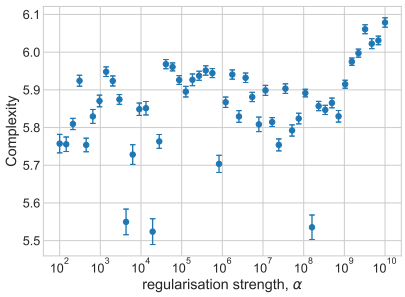
Fig. 12: Entropy for different datasets and varying values of α. Each dot corresponds to a separately trained model. The error bars represent the 95% confidence interval of the entropy computed on 128 images. Saliency maps were calculated using DeepLift.



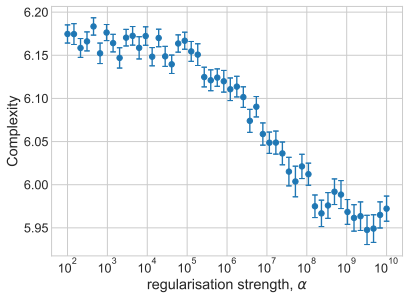
(a) Fashion-MNIST (LeNet).
$\rho = 0.418, p = 0.007$



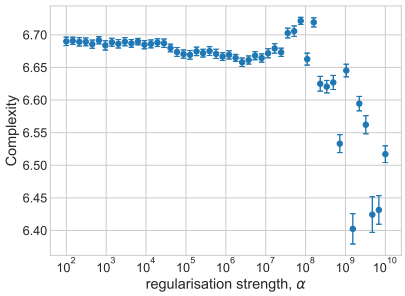
(b) KMNIST (LeNet).
$\rho = 0.445, p = 0.0034$



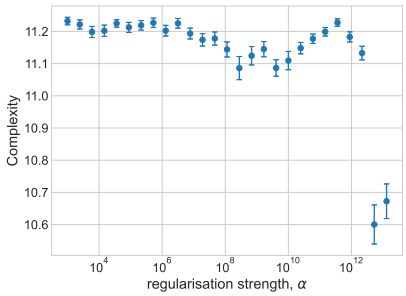
(c) MNIST (LeNet).
$\rho = 0.049, p = 0.749$



(d) MNIST (ResNet).
$\rho = -0.954, p = 0.0002$



(e) CIFAR-10 (ResNet).
$\rho = -0.687, p = 0.0002$

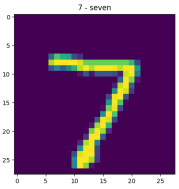


(f) Imagenette (ResNet).
$\rho = -0.530, p = 0.011$

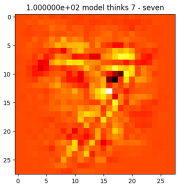
Fig. 13: Entropy for different datasets and varying values of α. Each dot corresponds to a separately trained model. The error bars represent the 95% confidence interval of the entropy computed on 128 images. Saliency maps were calculated using the gradient.

B Visual results

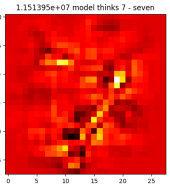
This appendix presents a visual comparison of saliency maps produced by models trained with different levels of regularisation. Even though it may not be possible to objectively determine improvement in saliency maps without using concrete metrics, it does seem that saliency maps on MNIST, KMNIST and Fashion-MNIST (visible in Figures 14, 16, 17) appear more visually pleasing when the input gradient of the trained model was regularised. This observation is consistent with findings from [25]. However, we refrain from drawing the same conclusion for CIFAR-10 (Figure 18) and Imagenette (Figure 19), as the effect seems to appear less obvious in these cases, and it is difficult to assess such differences based on visual inspection. Interesting to note is that, on MNIST, even nonsensical models seem to show readable numbers when trained using a very high gradient regularisation. An example of this is Figure 15, which shows saliency maps generated on MNIST digits for a model trained on Fashion-MNIST.



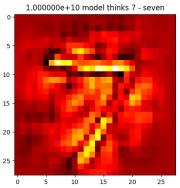
(a) First sample produced by an unshuffled dataloader on the test set of MNIST.



(b) Saliency sample of MNIST when $\alpha = 100$.

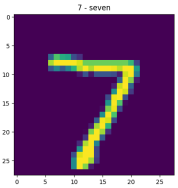


(c) Saliency sample of MNIST when $\alpha = 1.15e7$.

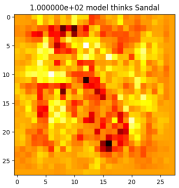


(d) Saliency sample of MNIST when $\alpha = 1.00e10$.

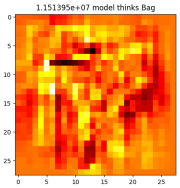
Fig. 14: DeepLIFT saliency evaluation on MNIST data using LeNet.



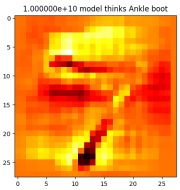
(a) Sample of MNIST, the corresponding Fashion-MNIST class is Sneaker.



(b) Saliency sample of MNIST when $\alpha = 100$.

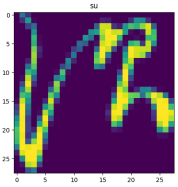


(c) Saliency sample of MNIST when $\alpha = 1.15e7$.

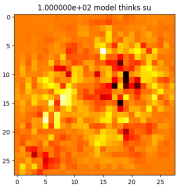


(d) Saliency sample of MNIST when $\alpha = 1.00e10$.

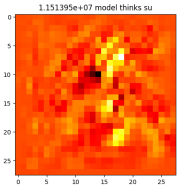
Fig. 15: DeepLIFT saliency evaluation on MNIST data using LeNet trained on Fashion-MNIST.



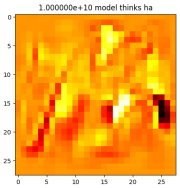
(a) First sample produced by an unshuffled dataloader on the test set of KMNIST.



(b) Saliency sample of KMNIST when $\alpha = 100$.

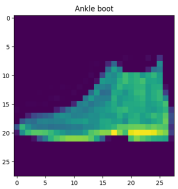


(c) Saliency sample of KMNIST when $\alpha = 1.15e7$.

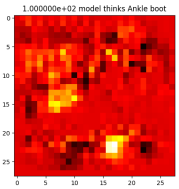


(d) Saliency sample of KMNIST when $\alpha = 1.00e10$.

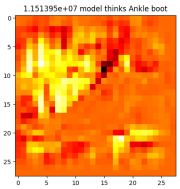
Fig. 16: DeepLIFT saliency evaluation on KMNIST data using LeNet.



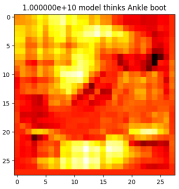
(a) First sample produced by an unshuffled dataloader on the test set of Fashion-MNIST.



(b) Saliency sample of Fashion-MNIST when α = 100.

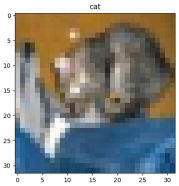


(c) Saliency sample of Fashion-MNIST when α = 1.15e7.

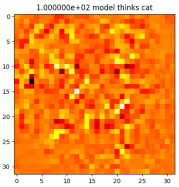


(d) Saliency sample of Fashion-MNIST when α = 1.00e10.

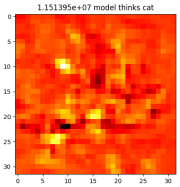
Fig. 17: DeepLIFT saliency evaluation on Fashion-MNIST data using LeNet.



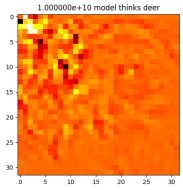
(a) First sample produced by an unshuffled dataloader on the test set of CIFAR-10.



(b) Saliency sample of CIFAR-10 when α = 100.

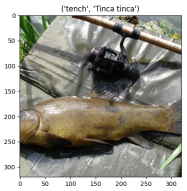


(c) Saliency sample of CIFAR-10 when α = 1.15e7.

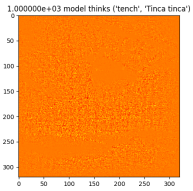


(d) Saliency sample of CIFAR-10 when α = 1.00e10.

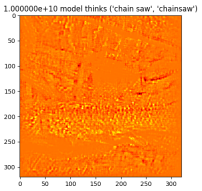
Fig. 18: DeepLIFT saliency evaluation on CIFAR-10 data using ResNet.



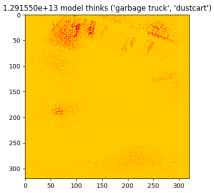
(a) First sample produced by an unshuffled dataloader on the test set of Imagenette.



(b) Saliency sample of Imagenette when $\alpha = 1000$.



(c) Saliency sample of Imagenette when $\alpha = 1.00e7$.



(d) Saliency sample of Imagenette when $\alpha = 1.29e13$.

Fig. 19: DeepLIFT saliency evaluation on Imagenette data using ResNet.