



HAL
open science

Explaining an image classifier with a generative model conditioned by uncertainty

Adrien Le Coz, Stéphane Herbin, Faouzi Adjed

► To cite this version:

Adrien Le Coz, Stéphane Herbin, Faouzi Adjed. Explaining an image classifier with a generative model conditioned by uncertainty. Uncertainty meets Explainability | Workshop and Tutorial @ ECML-PKDD 2023, Sep 2023, Torino, Italy. hal-04194943v1

HAL Id: hal-04194943

<https://hal.science/hal-04194943v1>

Submitted on 11 Nov 2023 (v1), last revised 2 Oct 2024 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Explaining an image classifier with a generative model conditioned by uncertainty

Adrien Le Coz^{1,2}, Stéphane Herbin²^[0000-0002-3341-3018], and Faouzi
Adjed¹^[0000-0002-0100-9352]

¹ IRT SystemX, Palaiseau, France

{adrien.le-coz, faouzi.adjed}@irt-systemx.fr

² DTIS, ONERA, Université Paris Saclay F-91123 Palaiseau - France

stephane.herbin@onera.fr

Abstract. We propose to condition a generative model by a given image classifier uncertainty in order to analyze and explain its behavior. Preliminary experiments on synthetic data and a corrupted version of MNIST dataset illustrate the idea.

Keywords: Generative model · Explainability of failure conditions · Uncertainty · Computer vision.

1 Introduction

Context: explaining the behavior of image classifiers. The growing use of image classifiers in many, sometimes critical, applications (e.g., medical diagnosis, autonomous driving, autonomous aircraft landing) reinforces the need to understand their behaviors. A key issue is to identify the conditions under which such systems are likely to fail, in order to ensure the safety of their use. With this objective in mind, one can consider uncertainty as a measure of potential failure: the question of failure condition identification can be translated into the problem of describing the nature of uncertain data for a given classifier.

Explainability is currently thought of as a tool to improve the trustworthiness of AI predictive systems. [1, 14]. In this paper, we propose to provide an explanation of the global classifier behavior as a representation of its uncertain data by using a generative model.

Explainability studies have mainly focused on providing so-called “post-hoc” explanations that are expected to somehow justify the actual prediction of a trained model. Very few studies have addressed the issue of identifying failure conditions. A related explanatory strategy is the design of counterfactuals [19, 5], which aim to identify what minimal and meaningful input modification will lead to a desired prediction change. In particular, several works [21, 18, 12, 9] leverage generative models such as GANs (Generative Adversarial Networks) [4] or diffusion models [8]. Generative models have also been used to quantify the uncertainty of a classifier [16] or discover causes of failures [20, 13].

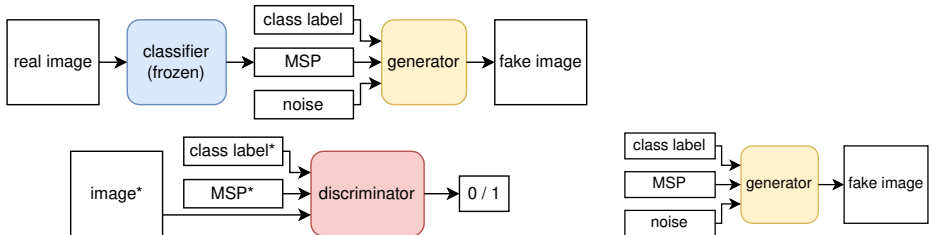
Main idea: GAN conditioned by the uncertainty of a classifier. Here we propose to explicitly create a generator of uncertain data. This is done by conditioning a generative model on the uncertainty of a given classifier. Such a

generative model can generate infinite amounts of uncertain data (as seen by the classifier) and provides a representation – an explanation – of what makes some data hazardous for the classifier. We expect to benefit from the learned model’s generalization capacity and use the generative model’s latent space – the “noise” – as a compact data representation.

The model architecture is depicted in Fig. 1a. A conditional GAN [15] takes a noise vector as input and a condition. Typically, this can be a one-hot embedding of the class to generate samples of a selected class. A simple way of conditioning a GAN is to concatenate the condition, e.g. one-hot embedding, to the noise vector as inputs for the generator, and also concatenate the condition to the real or fake image as inputs for the discriminator.

There are several ways to define the prediction uncertainty, e.g. entropy, maximum softmax probability (MSP) [7], or true class probability [2]. We use the imperfect but simple MSP as an uncertainty estimation. We add it as a condition for the generator to learn during training. Then after training, the model can generate uncertain data to get a global overview of the uncertainty. We also manipulate data to increase or decrease the uncertainty and exhibit sources of uncertainty.

MSP values are computed with the classifier (with frozen weights). For the discriminator used on real images, we compute their associated MSP first. For the discriminator used on fake images, we take the MSP used as a condition for the generator. To condition the generator, we apply the MSPs of random real images using the classifier to track the real distribution of MSP values. However, it is important to mention that we do not distinguish between aleatoric and epistemic uncertainty: the generative model is used to sample globally uncertain data.



(a) During training time, the additional input MSP conditions the generator. The discriminator evaluates if the combination (class, MSP, image) is realistic.

*for the discriminator, inputs are alternatively (class label, MSP from classifier, real image) and (class condition, MSP condition, fake image generated).

(b) After training, we can generate uncertain images (fix low MSP and vary noise) or identify sources of uncertainty for given images (fix the noise and vary MSP).

Fig. 1: Training process and structure of the generator.

2 Preliminary experiments

Two-dimensional moons data. We first illustrate the approach with a simple problem using the moons dataset [17]. The data is 2-dimensional and looks like

two interleaving half-circles corresponding to the upper and lower moon classes. The noise level can be adjusted, and we fix it to 0.3 to have an area where the two classes are mixed. We train a simple fully-connected neural network as a classifier. We use a simple generator based on a fully connected network conditioned by one-hot class embedding and the MSP.

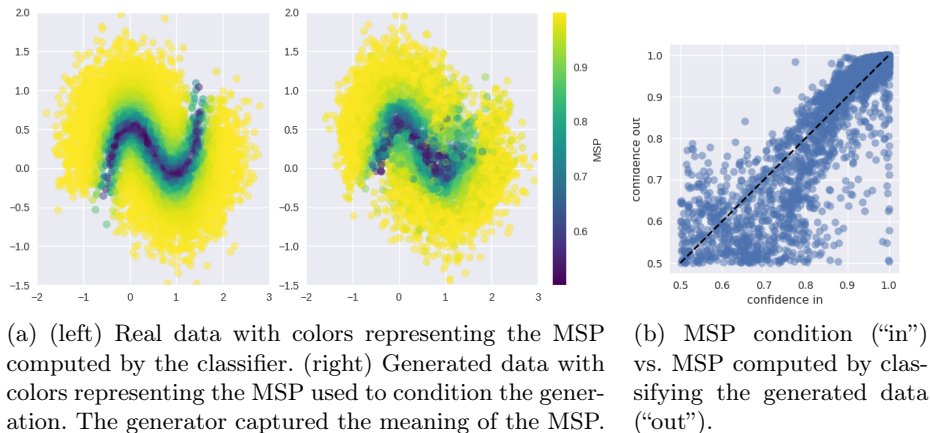


Fig. 2: Qualitative and quantitative results for moons dataset. Uncertainty conditioning works well; the MSP condition corresponds roughly to the real MSP.

Fig. 2a on the left shows the data, with colors representing the MSP obtained when classifying the data. We can see that the MSP is close to 1, where the classes do not mix, but gets lower in the middle area where the classes mix, representing higher uncertainty (mostly aleatoric). Whereas, fig. 2a on the right shows synthetic data conditioned by MSP. The values are sampled from MSP computed on real data to follow the same distribution. We can see similarities between the locations of real data with high MSP and synthetic data conditioned by high MSP, and likewise for low MSP. The generator captured which kind of data is uncertain and can generate such data when conditioned with low MSP. For more quantitative results, we follow this process: fix some MSP values as conditions (“input confidence”), generate fake data, classify it, and obtain the MSP of the classifier (“output confidence”). Ideally, both values should be the same every time. As seen in Fig. 2b, it is not necessarily the case, especially for lower values. Yet, the two are correlated.

Corrupted MNIST. Let us now consider more complex data: images. We use the MNIST dataset [3], which contains black and white images of handwritten digits with ten classes (digits from 0 to 9). We train a Convolutional Neural Network to classify digits from images. As the task of digit classification of clean MNIST images is too simple (the classifier is almost perfect and highly confident), we choose to corrupt MNIST images to make the problem more realistic. We use Gaussian blur and noise similarly to ImageNet-C [6]. These corruptions are applied on a random half of the images, with a random severity

level out of 5 possible levels. We found that it results in a reasonable accuracy reduction compared to clean MNIST: now 94.0% on the train set and 93.2% on the validation set (instead of 98.8% and 98.5%, respectively). Also, MSP values are more spread out. The GAN is now based on the StyleGAN2 [11, 10] architecture to handle images, with additional conditioning for the MSP. The conditioning is a concatenation of a class embedding and the MSP value.

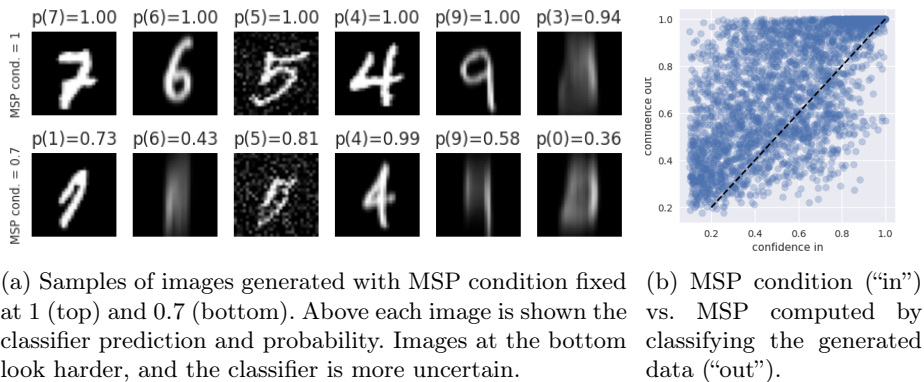


Fig. 3: Qualitative and quantitative results for corrupted MNIST dataset. Uncertainty conditioning works but not precisely as for moons data.

We can generate uncertain images by fixing a low MSP value and varying the noise input, as illustrated in Fig. 3a bottom. Also, comparing Fig. 3a top and bottom, we gain insight into the classifier’s sources of uncertainty by observing what makes given images more uncertain (by fixing noise inputs and lowering the MSP condition). In this case, it is primarily shape, Gaussian noise, and blur that perturbs the classifier.

The qualitative results in Fig. 3a show that images generated with the conditioning of $\text{MSP} = 1$ mainly result in “output” MSPs close to 1. We get more spread-out “output” MSP values when conditioned with $\text{MSP} = 0.7$. Fig. 3b shows that “input” MSP and “output” MSP can be pretty different. While not as good as on the moons dataset, we still observe some correlation. We hypothesize that the MSP is much less well-defined on MNIST images than on the moons dataset. More substantial constraints on the conditioning should be considered to improve the results.

3 Conclusion

We created an explicit generator of uncertain data that can be used in several ways. It can give a global outlook of uncertain images by generating them on demand. It can also corrupt images (transform them into their more uncertain form) to visualize sources of local uncertainty. The results are preliminary but encouraging. The MSP might not contain sufficient information to capture the classifier behavior, and the constraint put on the condition during training should be reinforced. Leveraging generative models is a promising way to improve explainability when uncertain data is rare.

Acknowledgments

This work has been supported by the French government under the "Investissements d'avenir" program, as part of the SystemX Technological Research Institute.

This work was granted access to the HPC/AI resources of IDRIS under the allocation 2022-AD011013372 made by GENCI.

References

1. Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., Herrera, F.: Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI (Dec 2019). <https://doi.org/10.48550/arXiv.1910.10045>
2. Corbière, C., Thome, N., Saporta, A., Vu, T.H., Cord, M., Pérez, P.: Confidence Estimation via Auxiliary Models. *arXiv:2012.06508 [cs, stat]* (May 2021)
3. Deng, L.: The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine* **29**(6), 141–142 (Nov 2012). <https://doi.org/10.1109/MSP.2012.2211477>
4. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Networks (Jun 2014). <https://doi.org/10.48550/arXiv.1406.2661>
5. Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., Lee, S.: Counterfactual Visual Explanations (Jun 2019). <https://doi.org/10.48550/arXiv.1904.07451>
6. Hendrycks, D., Dietterich, T.: Benchmarking Neural Network Robustness to Common Corruptions and Perturbations (Mar 2019). <https://doi.org/10.48550/arXiv.1903.12261>
7. Hendrycks, D., Gimpel, K.: A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. *arXiv:1610.02136 [cs]* (Oct 2018)
8. Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models (Dec 2020). <https://doi.org/10.48550/arXiv.2006.11239>
9. Jeanneret, G., Simon, L., Jurie, F.: Adversarial Counterfactual Visual Explanations. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 16425–16435 (2023)
10. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training Generative Adversarial Networks with Limited Data. *arXiv:2006.06676 [cs, stat]* (Oct 2020)
11. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and Improving the Image Quality of StyleGAN. *arXiv:1912.04958 [cs, eess, stat]* (Mar 2020)
12. Lang, O., Gandelsman, Y., Yarom, M., Wald, Y., Elidan, G., Hassidim, A., Freeman, W.T., Isola, P., Globerson, A., Irani, M., Mosseri, I.: Explaining in Style: Training a GAN to explain a classifier in StyleSpace. *arXiv:2104.13369 [cs, eess, stat]* (Sep 2021)
13. Le Coz, A., Herbin, S., Adjed, F.: Leveraging generative models to characterize the failure conditions of image classifiers. In: *The IJCAI-ECAI-22 Workshop on Artificial Intelligence Safety (AISafety 2022)* (Jul 2022)

14. Linardatos, P., Papastefanopoulos, V., Kotsiantis, S.: Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy* **23**(1), 18 (Jan 2021). <https://doi.org/10.3390/e23010018>
15. Mirza, M., Osindero, S.: Conditional Generative Adversarial Nets. arXiv:1411.1784 [cs, stat] (Nov 2014)
16. Oberdiek, P., Fink, G.A., Rottmann, M.: UQGAN: A Unified Model for Uncertainty Quantification of Deep Classifiers trained via Conditional GANs (Oct 2022). <https://doi.org/10.48550/arXiv.2201.13279>
17. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É.: Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12**(85), 2825–2830 (2011)
18. Sauer, A., Geiger, A.: Counterfactual Generative Networks. arXiv:2101.06046 [cs] (Jan 2021)
19. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR (Mar 2018). <https://doi.org/10.48550/arXiv.1711.00399>
20. Wiles, O., Albuquerque, I., Goyal, S.: Discovering Bugs in Vision Models using Off-the-shelf Image Generation and Captioning (Aug 2022). <https://doi.org/10.48550/arXiv.2208.08831>
21. Zhao, Z., Dua, D., Singh, S.: Generating Natural Adversarial Examples (Feb 2018). <https://doi.org/10.48550/arXiv.1710.11342>