



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. Workshop Uncertainty meets Explainability, ECML-PKDD 2023, Sep 2023, Torino, Italy. hal-04194943v2

HAL Id: hal-04194943

<https://hal.science/hal-04194943v2>

Submitted on 2 Oct 2024

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.

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. Identifying sources of uncertainty in an image classifier is a crucial challenge. Indeed, the decision process of those models is opaque and does not necessarily correspond to what we might expect. To help characterize classifiers, generative models can be used as they allow the control of visual attributes. Here we use a generative adversarial network to generate images corresponding to how a classifier sees the image. More specifically, we consider the classifier maximum softmax probability as an uncertainty estimation and use it as an additional input to condition the generative model. This allows us to generate images that result in uncertain predictions, giving us a global view of which images are harder to classify. We can also increase the uncertainty of a given image and observe the impact of an attribute, providing a more local understanding of the decision process. We perform experiments on the MNIST dataset, augmented with corruptions. We believe that generative models are a helpful tool to explain the behavior and uncertainties of image classifiers.

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. [2, 15]. 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 [20, 6], which aim to identify what minimal and meaningful input modification will lead to a desired prediction change. In particular, several works [23, 19, 13, 10] leverage generative models such as GANs (Generative Adversarial Networks) [5] or diffusion models [9]. Generative models have also been used to quantify the uncertainty of a classifier [17] or discover causes of failures [21, 14].

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.

2 Method

Generative Adversarial Networks (GANs) [5] are a type of generative model known for their generation quality and the controllability offered by their latent (input) space. In particular, they can generate full size images. They are composed of two neural networks: a generator that generates fake images and a discriminator that distinguishes fake images from real ones from the training data. The training is a competition between the two: the generator tries to fool the discriminator, which seeks not to be fooled. During this process, the two improve until the discriminator cannot distinguish any more real from fake data. GAN training is known to be unstable, so an equilibrium has to be found: if the discriminator becomes much better than the generator, it “wins”, and it’s hard for the generator to improve and fill the gap, and vice-versa. Losses and regularizations have been developed to fix the issue [1]. After training, we discard the discriminator and use the generator to generate images from noise vectors (“latent codes”). Interestingly, the model is structured so that interpolations between two latent codes result in a smooth semantic shift of an image into another; for instance, a digit image of “8” is progressively transformed into a mixture of “3” and “8” before ultimately becoming a “3”, which is not the case if the interpolation is done in the pixel space.

We use the model StyleGAN2 [11, 12], widely used for high-quality face generation and edition. It has a unique architecture. The input latent space \mathcal{Z} is mapped through fully connected layers to an intermediate latent space \mathcal{W} . The image is generated progressively at different scales, starting from an initial constant tensor with a size of 4^2 and 512 channels, which is up-sampled and transformed by residual convolution layers, and results in images of up to

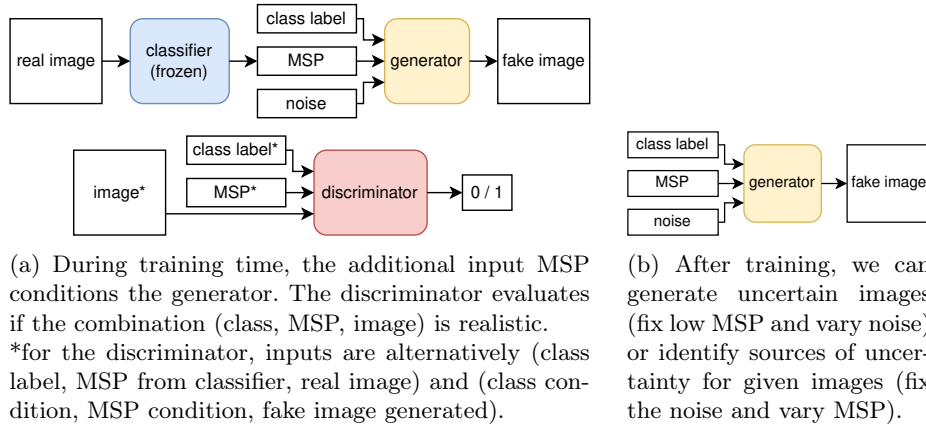


Fig. 1: Training process and structure of the generator.

1024² pixels. Latent codes $\mathbf{w} \in \mathcal{W}$ are transformed into *styles* $\mathbf{s} \in \mathcal{S}$ through learned affine transformations. Those styles will scale the convolution weights of each feature map for each generator layer. Styles applied at low resolution affect high-level aspects of the face (pose, hairstyle...), while at higher resolutions, they affect small details (microstructure...). Latent spaces \mathcal{W} and \mathcal{S} are highly disentangled, meaning that they encode distinct visual attributes along different dimensions. This allows image editing, one attribute at a time [22]. In particular, we can also use the latent space to characterize classifiers [13, 17, 14]. Here, we exploit generative models more straightforwardly by conditioning the generation with the classifier uncertainty, so that it becomes an input of the generator.

Our model architecture is depicted in Fig. 1a. A conditional GAN [16] 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., encoded as 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) [8], or true class probability [3]. We use the imperfect but simple MSP as an uncertainty estimation. We add it as an input condition to the generator. 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, which is not the same as what the classifier would output because of the imperfect generation. To condition the generator, we apply the MSPs of random real images using the classifier to track the real distribution of

MSP values. Otherwise, the discriminator would more easily make the difference between real and fake data, causing the generator training to be harder. 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.

3 Preliminary experiments

Two-dimensional moons data. We first illustrate the approach with a simple problem using the moons dataset [18]. 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. The network has 5 layers with a latent space of dimension 8. The conditioning is a concatenation of the class information as a one-hot embedding vector (of dimension 2) and the MSP as a continuous value.

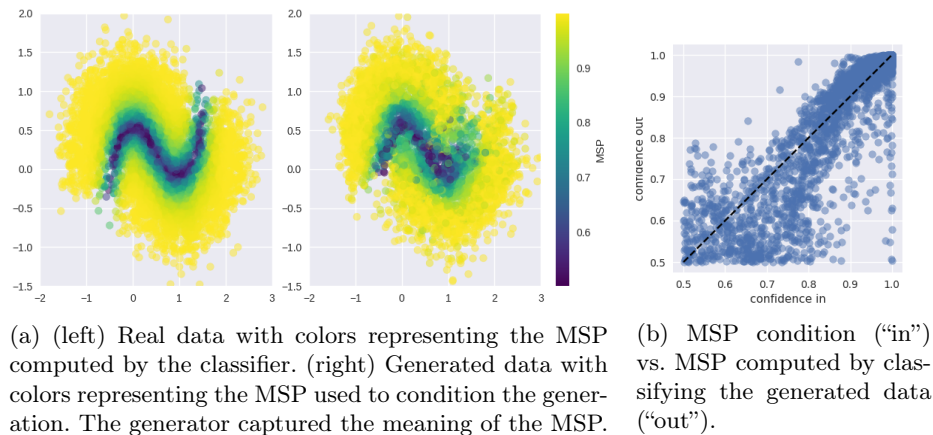
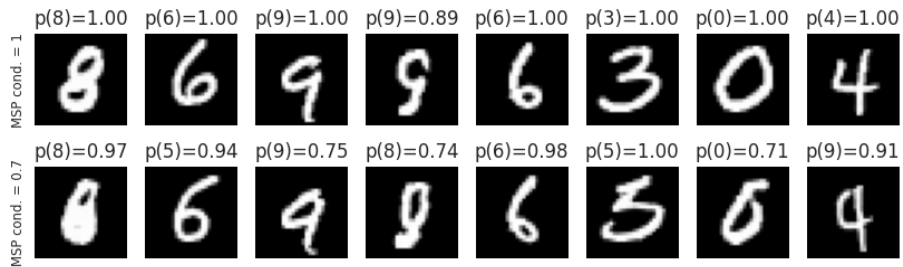


Fig. 2: Qualitative and quantitative results for moons dataset. Uncertainty conditioning works well; the MSP condition corresponds roughly to the real MSP.

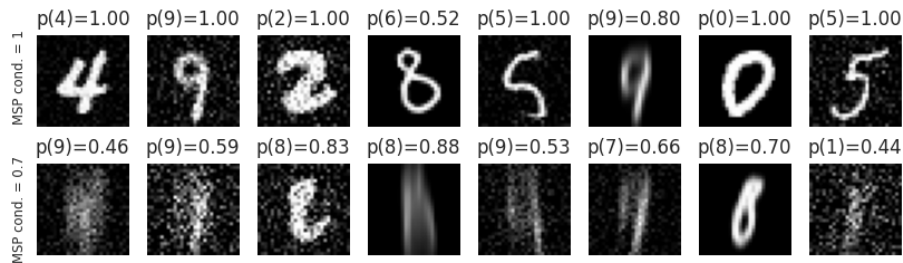
Fig. 2a on the left shows the data, with colors representing the MSP obtained when classifying the data. We can see that the MSP is close to 1, where the classes do not mix but gets lower in the middle area where the classes mix, representing higher uncertainty. We can note that this uncertainty is mostly aleatoric: data can be of either class in the middle region. Whereas, in 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.

MNIST. Let us now consider more complex data: images. We use the MNIST dataset [4], 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. We consider clean MNIST data, but to make the problem more realistic, we also choose to corrupt MNIST images. We use Gaussian blur and noise similarly to ImageNet-C [7]. These corruptions are applied on a random half of the images, with a random severity level out of 5 possible levels.



(a) clean MNIST



(b) corrupted MNIST

Fig. 3: Samples of images generated with MSP condition fixed at 1 (top) and 0.7 (bottom). Above each image is shown the classifier prediction and probability. Images at the bottom look harder, and the classifier is more uncertain.

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 [12, 11] architecture to handle images, with additional conditioning for the MSP. We keep the default latent space dimension of 512 (for the noise), as reducing it makes the training more difficult. The conditioning is a concatenation of a class embedding and the MSP value.

We can generate uncertain images by fixing a low MSP value and varying the noise input, as illustrated in Fig. 3a and 3b bottom. Also, comparing Fig. 3a and 3b top versus 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 perturbrates the classifier.

The qualitative results in Fig. 3a and 3b 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. 4 shows that “input” MSP and “output” MSP can be quite 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.

4 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. Leveraging generative models is a promising way to improve explainability when uncertain data is rare.

We identified some ideas for future work. The MSP might not contain sufficient information to capture the classifier behavior, so more information, like the full vector, could be considered. The constraint put on the condition during training should be reinforced, for instance with an additional loss term. Furthermore, the MSP might not be calibrated: the probability might be overestimated and require a recalibration.

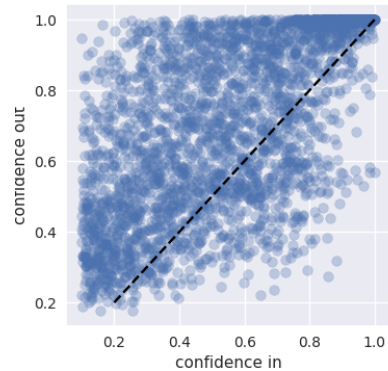


Fig. 4: MSP condition (“in”) vs. MSP computed by classifying the generated data (“out”).

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. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International conference on machine learning. pp. 214–223. PMLR (2017)
2. 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>
3. 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)
4. 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>
5. 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>
6. 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>
7. Hendrycks, D., Dietterich, T.: Benchmarking Neural Network Robustness to Common Corruptions and Perturbations (Mar 2019). <https://doi.org/10.48550/arXiv.1903.12261>
8. Hendrycks, D., Gimpel, K.: A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. arXiv:1610.02136 [cs] (Oct 2018)
9. Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models (Dec 2020). <https://doi.org/10.48550/arXiv.2006.11239>
10. 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)
11. 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)
12. 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)
13. Lang, O., Gandselman, 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)

14. 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)
15. 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>
16. Mirza, M., Osindero, S.: Conditional Generative Adversarial Nets. arXiv:1411.1784 [cs, stat] (Nov 2014)
17. 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>
18. 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)
19. Sauer, A., Geiger, A.: Counterfactual Generative Networks. arXiv:2101.06046 [cs] (Jan 2021)
20. 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>
21. 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>
22. Wu, Z., Lischinski, D., Shechtman, E.: Stylespace analysis: Disentangled controls for stylegan image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12863–12872 (2021)
23. Zhao, Z., Dua, D., Singh, S.: Generating Natural Adversarial Examples (Feb 2018). <https://doi.org/10.48550/arXiv.1710.11342>