GeValDi: Generative Validation of Discriminative Models

Anonymous authors
Paper under double-blind review

Abstract

Evaluation of machine learning (ML) models is critically important for reliable use. Though typically done via unseen data, such validation datasets often need to be large and hard to procure; additionally, mutliple models may perform equally well on such datasets. To address these challenges, we offer GeValdi: a data-efficient method to validate discriminative classifiers by creating samples where such classifiers maximally differ. We demonstrate how such "maximally different samples" can be constructed and leveraged to probe the failure modes of classifiers and offer a hierarchically-aware metric to further support fine-grained, comparative model evaluation.

1 Introduction

In a typical machine learning (ML) problem, several ML models may perform equally well on training datasets (Breiman, 2001), but substantially differently on unseen data. How then are we to choose which model (or set of models) to deploy? Distinguishing where similarly performing models differ requires large, annotated validation datasets that are typically not available. We propose a data-efficient solution that probes the differences between comparably performing classifiers *without using validation data*. We *synthesise* data where predictions of two classifiers maximally differ. This is done by optimising in the latent space of a generative model. We call our approach GeValDi: Generative Validation of Discriminative classifiers. Using ImageNet (Deng et al., 2009) as a case study, we empirically investigate the ability of our approach to generate "maximally different samples."

2 GeValDi

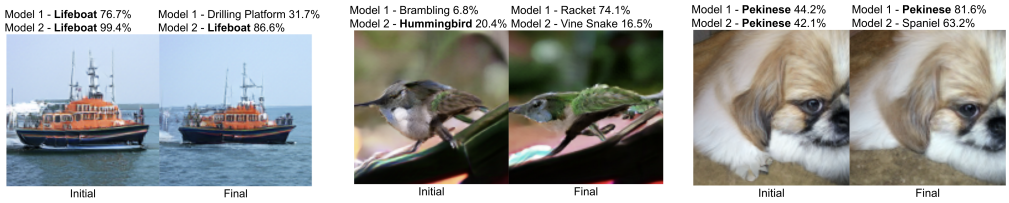
In this section, we introduce GeValDi. Given two disriminative classifiers, defined by their predictive probabilities $p_1(\mathbf{y}|\mathbf{x})$ and $p_2(\mathbf{y}|\mathbf{x})$, GeValDi generates data in the input space where the classifiers maximally differ, which is formalised as

$$\tilde{\mathbf{x}} = \arg\max_{\mathbf{x}\in\mathcal{X}} D\left[p_1(\mathbf{y}|\mathbf{x}) \| p_2(\mathbf{y}|\mathbf{x})\right] \tag{1}$$

where $\tilde{\mathbf{x}} \in \mathcal{X}$ denotes the *maximally different sample* (MDS), $\mathcal{X}$ denotes the input space and D denotes some divergence measure, such as the KL-divergence (Kullback & Leibler, 1951). However, without restricting the search space, optimising this objective will return data on which our classifiers are unlikely to be evaluated on in practice (i.e., under the true data distribution). Our key insight is to constrain the search by optimising in the latent space of a generative model, such as $p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$ where $p(\mathbf{z})$ denotes the prior, and $p(\mathbf{x}|\mathbf{z})$ is some conditional distribution with mean function $E[p(\mathbf{x}|\mathbf{z})] = g(\mathbf{z})$. Provided $p(\mathbf{x})$ accurately models the data distribution, we can formalise the problem of generating an MDS with high-probability under the true data distribution as

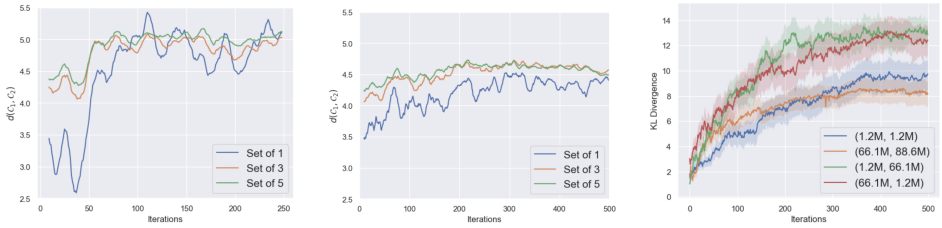
$$g^{-1}(\tilde{\mathbf{x}}) = \arg\max_{\mathbf{z}\in\mathcal{Z}} D\left[p_1(\mathbf{y}|g(\mathbf{z})) \| p_2(\mathbf{y}|g(\mathbf{z}))\right] + \lambda \log p(\mathbf{z}) \tag{2}$$

where λ is some regularisation constant which trades off flexibility of the optimisation problem with the likelihood of the generated sample under $\tilde{p}(\mathbf{x})$. By optimising in latent space, not only do we ensure samples are from the true data manifold, we only have to optimise over low-dimensional latent spaces.



(a) (b) (c)

Figure 1: Each figure shows images generated by the GAN before (left) and after (right) the latent space optimisation for Model 1 - GoogleNet (Szegedy et al., 2014) vs Model 2 - AlexNet (Krizhevsky et al., 2012). Bolded text highlights the BigGAN class-label used to generate samples.



(a) (b) (c)

Figure 2: (a) and (b) show evolution of set distance for top $n = 1, 3, 5$ predictions for a pair of models, where number of parameters (model 1, model 2) are (1.2 million, 1.2 million) and (66.1 million, 88.6 million) respectively. (c) shows evolution of KL divergence for four pairs of models of varying expressivities.

3 Results

We study the ability of GeValDi to generate realistic synthetic samples for which the predictions of two high-performing classifiers differ, on ImageNet (Deng et al., 2009). We employ BigGAN (Brock et al., 2018) for generation; a complete list of experimental details can be found in Appendix B.

In Figure 1, we compare the optimised MDS with the pre-optimised latent variable image, along with predicted class probabilities for our two classifiers. As expected, whilst photo-realistic, the optimised MDS deviate noticeably from the pre-optimised images. Furthermore, we are able to find failure modes of high-performance classifiers using synthetic data. For example, in Figure 1(a), although the final image is clearly a lifeboat, model 1 incorrectly classifies it as a drilling platform. In Figure 2 shows evolution of KL divergence, and prediction set distance for models with different expressivities. Prediction set distance measures the distance between two sets of predictions along label hierarchy (see Appendix C). In Figure 2, increases in set distance show that predictions shift laterally across the label hierarchy, and this shift is affected by classifier expressivity. Therefore, this allows us to contrast classifier predictions along the label hierarchy for classifier selection.

4 Conclusion

GeValDi is able to generate realistic samples of maximal difference, exposing failure modes of classifiers. Further, model expressivity affects the hierarchical set distances of predictions, pointing to the possibility that models with higher expressivities are able to learn label hierarchies better. Therefore, with GeValDi, we are not only able to find failure modes of high-performance classifiers, but also better understand how classifiers have learnt label hierarchies, offering a novel way of evaluating classifiers.

5 URM Statement

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2023 Tiny Papers Track.

References

Leo Breiman. Statistical modeling: The two cultures. *Statistical Science*, 16(3):199–231, 2001.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis, 2018. URL https://arxiv.org/abs/1809.11096.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Christiane Fellbaum (ed.). *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA, 1998. ISBN 978-0-262-06197-1.

Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and lt;0.5mb model size. 2016. doi: 10.48550/ARXIV.1602.07360. URL https://arxiv.org/abs/1602.07360.

Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8237. doi: 10.1561/2200000056. URL http://dx.doi.org/10.1561/2200000056.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. 2022. doi: 10.48550/ARXIV.2201.03545. URL https://arxiv.org/abs/2201.03545.

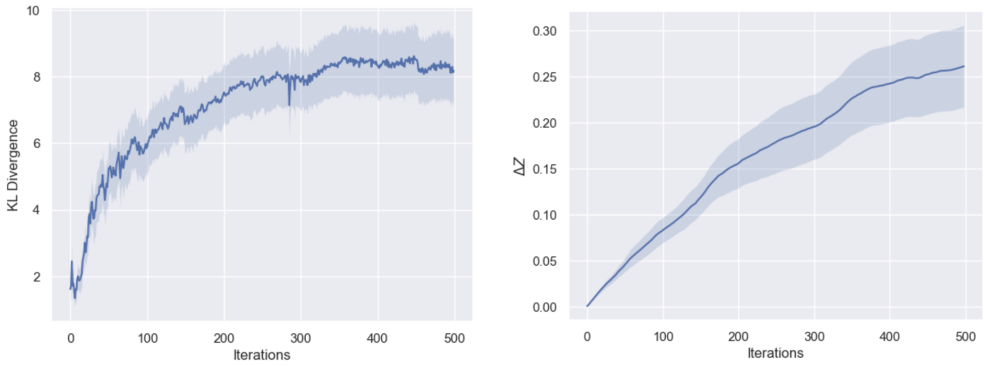
Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. URL https://arxiv.org/abs/1409.4842.

Rikiya Yamashita, Mizuho Nishio, Richard Kinh Do, and Kaori Togashi. Convolutional neural networks: An overview and application in radiology. *Insights into Imaging*, 9(4): 611–629, 2018. doi: 10.1007/s13244-018-0639-9.

A Validating the MDS Algorithm

In order to make sure the MDS algorithm works as it is supposed to, the path of KL divergence and the evolution of the latent space vector is checked.

Figure 3(a) shows that KL divergence is clearly increasing across iterations consistently across the various experiments. This means that we are actually able to learn samples that



(a) (b)

Figure 3: (a) shows the evolution of KL divergence across iterations of MDS algorithm. (b) shows the evolution of $\Delta\mathbf{z}$ across iterations of MDS algorithm

our models maximally disagree on, and are not just getting a random draw of samples from the latent space.

Furthermore, we need to understand whether these latent space samples are any different from the initial starting point, or are we just perturbing the initial point to generate 'diverse' samples. We do this by plotting the evolution of the squared difference between the initial starting point and the current latent space point in our MDS algorithm, $\Delta\mathbf{z} = E[\frac{||\mathbf{z}-\mathbf{z}_{\text{init}}||_2}{||\mathbf{z}||_2}]$. As evident in Figure 3(b), we see that the latent space point changes significantly to the initial starting point ($\Delta\mathbf{z} \sim 25\%$) and this shows us that our MDS algorithm actually explores the latent space to find maximally different samples.

B Experimental Setup

For our experiments involving the MNIST dataset (Deng, 2012), a Variational Autoencoder (VAE) (Kingma & Welling, 2019) is used as the latent variable model, with Convolutional Neural Networks (CNNs) (Yamashita et al., 2018) as the discriminative classifiers. Note that the CNNs classifiers have classification accuracies of 98.6% and 98.8%.

For experiments involving ImageNet (Deng et al., 2009), pretrained discriminative classifiers and Generative Adversarial Networks (GANs) were used. For the GAN, BigGAN by DeepMind (Brock et al., 2018), which generates images from the data distribution that generates the ImageNet dataset, is used. For the classifiers, pairwise comparisons are made between AlexNet (Krizhevsky et al., 2012), GoogleNet (Szegedy et al., 2014), SqueezeNet (Iandola et al., 2016) and ConvNeXt (Liu et al., 2022). Note that the choice of these classifiers provide a wide variety of model capacities that allow us to investigate the effect it has on MDS samples.

To explore the impact of model expressivity, we do 4 experiments (note that M denotes a million parameters): 2 models with similar, but low capacity (SqueezeNet0 - 1.2M, SqueezeNet1 - 1.2M), 2 models with similar, but high capacity (AlexNet - 66.1M, ConvNeXt Base - 88.6M), 2 models with different capacity (SqueezeNet 0 - 1.2M, AlexNet - 66.1M) and lastly, the same 2 models but with the ordering reversed (AlexNet - 66.1M, SqueezeNet 0 - 1.2M). The last experiment explores how changing the ordering of the models in our KL divergence changes the MDS we generate, since KL divergence is non-symmetric.

C Prediction Set Distance

We develop a metric to quantify the discrepancy between models' predictions which accounts for *hierarchical structure* in the data space. Specifically, our approach utilizes the pre-defined

class hierarchy in ImageNet, known as WordNet (Fellbaum, 1998); this taxonomy defines distance between classes related to their conceptual similarity. Our metric measures the distance between the top n classes predicted by two classifiers, $\mathcal{C}_1$ and $\mathcal{C}_2$. By analyzing the evolution of the distance metric as we progress through the MDS algorithm, we gain insights into how the predictions of two models differ.

$$d(\mathcal{C}_1, \mathcal{C}_2) = \frac{1}{|\mathcal{C}_1||\mathcal{C}_2|} \sum_{c_i \in \mathcal{C}_1} \sum_{c_j \in \mathcal{C}_2} H_w(c_i, c_j) \tag{3}$$

where $|\mathcal{C}|$ gives the size of set $\mathcal{C}$, and $H_w(a, b)$ computes the minimum distance from class a to class b along the hierarchy of classes, which is defined as *half the length of the shortest path from one class to another*.

D Additional MDS Examples

In this section, we present several more examples of MDS in Figure 4, with some interesting observations to reinforce the utility of this method.

We are able to identify a few types of failure modes of these classifiers, and we look at them in detail here. Firstly, consider Figure 4(b), where both models initially classify the images as "Harvestman" (i.e. the BigGAN class-label). After optimisation, model 2 flips the prediction to "Barn Spider" with more than 50% confidence, which is interesting as it illustrates a failure mode where there are certain input perturbations in the image space that nudge model 2 to flip its prediction. Furthermore, we notice that the set distance between "Harvestman" and "Barn Spider" is only 1.0, which means that flip in class occurred at the same level in the hierarchy.

Contrasting this to example Figure 4(e), where the the predictions evolve from "Orange" and "Granny Smith", to "Maraca" and "Granny Smith", where the set distance increases from 2.0 to 7.0 when model 1 flips its prediction from "Orange" to "Maraca", we see that this failure mode exhibits a large lateral shift of the model predictions along the hierarchy. We note that this category of failure modes are more serious than the previous failure mode because the prediction is more incorrect from the hierarchical perspective in the former case.

As per Figure 4(i), we observe that this latent space optimisation can sometimes point to regions in the image space that improve the model predictions. In other words, in this example, we essentially found perturbations of the image space that nudged both classifiers to correctly classify the image. This behaviour is also observed in Figure 4(k) where the classifiers become more correctly confident in their predictions.

Therefore, these examples point to using MDS to not only find failure modes, but also perturbations in image space that improve model predictions.

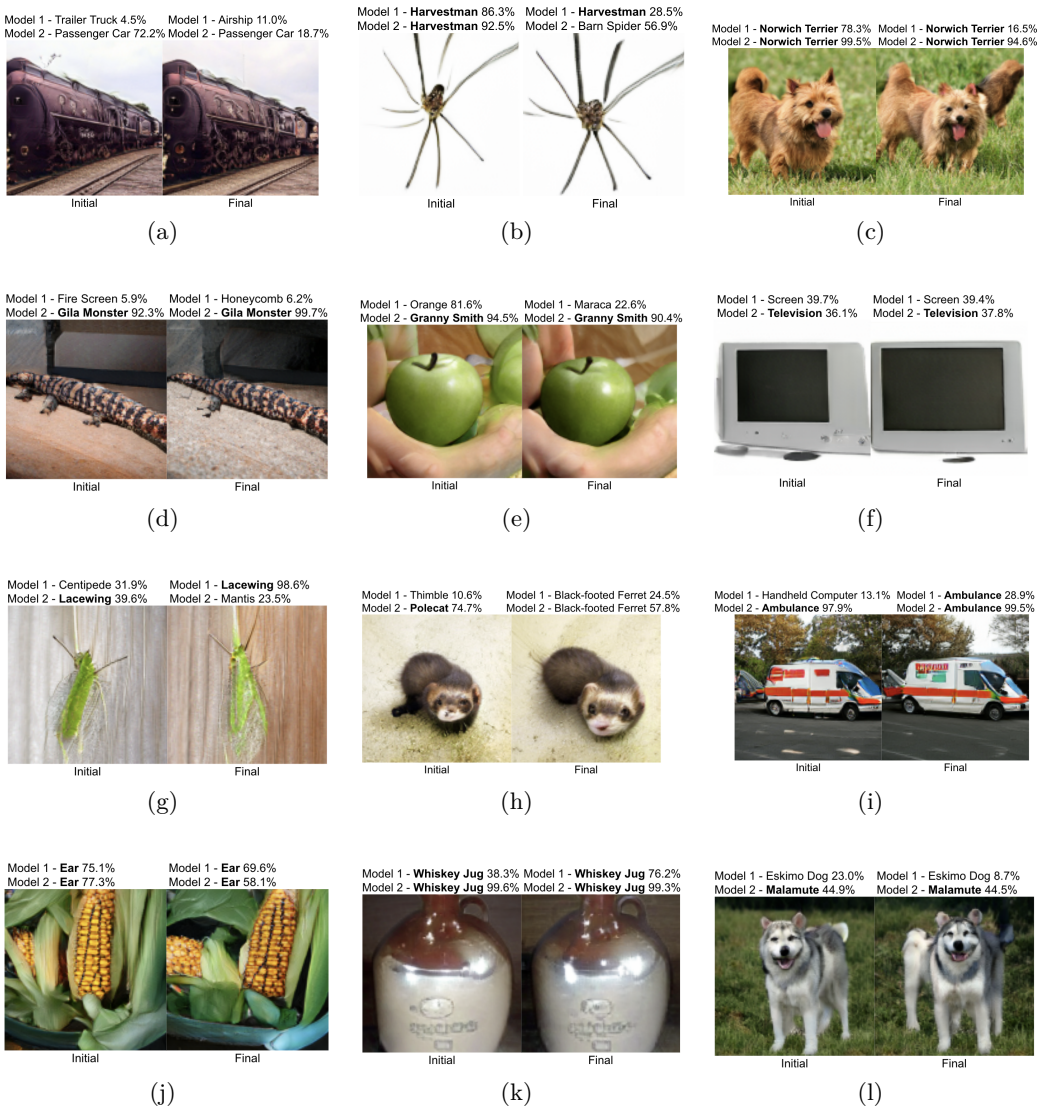


Figure 4: Each figure shows images generated by the GAN before (left) and after (right) the latent space optimisation for Model 1 (GoogleNet) vs Model 2 (AlexNet). Bolded text is used to highlight the BigGAN class-label used to generate samples.