GeValDi: Generative Validation of Discriminative Models

Anonymous authors
Paper under double-blind review

Abstract

The evaluation of machine learning (ML) models is a core tenet of trustworthy use. Evaluation is typically done via a held-out dataset. However, such validation datasets often need to be large and are hard to procure; further, multiple models may perform equally well on such sets. To address these challenges, we offer GeValdi: an efficient method to validate discriminative classifiers by creating samples where such classifiers maximally differ. We demonstrate how such "maximally different samples" can be constructed via and leveraged to probe the failure mode of classifiers and offer a hierarchically-aware metric to further support fine-grained, comparative model evaluation.

1 Introduction

Many different machine learning (ML) models may be able to perform comparably well on observed data (Black et al., 2022); however, at test-time these models may deviate substantially in their predictions on unseen data. How should we choose which model (or set of models) we wish to deploy? Identifying where seemingly-comparable models differ typically requires large, annotated validation datasets – which may not be readily available. We propose a more data-efficient solution for probing the differences between comparably performing classifiers *without using validation data*.

We *synthesise* data for which the predictions of two classifiers differ maximally. To do so, we optimise in the latent space of a generative model. We dub 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 method to generate 'maximally different samples.' Furthermore, we explore the path of classifier predictions along the latent space optimisation path, and how model expressivity affects it.

2 Related Work

Prior work has leveraged generating synthetic data to explore the properties of classifiers (Hittmeir et al., 2019). Yousefzadeh & O'Leary (2019) studies the characteristics of classifiers by finding points in input space about which small perturbations changes the predicted class (i.e., "flip points"). These "flip points" help identify uncertainty in classifications and determine minimum input perturbations required for a class flip. Adversarial example generation finds small perturbations of existing images such that the predicted class label changes (Dalvi et al., 2004; Goodfellow et al., 2014; Szegedy et al., 2013; Papernot et al., 2016). Here, we first consider characterising the differences in the predictions made by two classifiers, rather than characterising a single classifier (Demšar, 2006; Coston et al., 2021).

Further, we perform optimisation *in the latent space of a generative model*, rather than the original image space (Upadhyay & Mukherjee, 2021; Creswell et al., 2017) which allows us to stay in the data manifold. Relatedly, Antorán et al. (2020) identifies changes in input space by performing latent space optimisation such that classifier uncertainty decreases. Roth et al. (2022) uses points where models substantially disagree to iteratively improve models. Our work bridges these directions to support model differentiation at evaluation-time.

3 GeValDi

In this section, we introduce GeValDi, a framework for performing generative validation of discriminative classifiers. Provided two discriminative classifiers, defined by their predictive probability distributions, $p_1(\mathbf{y}|\mathbf{x})$ and $p_2(\mathbf{y}|\mathbf{x})$, GeValDi generates data in the input space where the classifiers differ maximally, 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 which such classifiers are unlikely to be evaluated on in practice (i.e., under the true data distribution). Thus, it is necessary to constrain the optimisation to the data manifold. Our key insight is to model this data distribution – and optimise directly *in the latent space*. Specifically, consider the latent variable model:

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z} \tag{2}$$

where $p(\mathbf{z})$ denotes the prior over the latent variable $\mathbf{z}$, 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

$$g^{-1}(\tilde{\mathbf{x}}) = \arg\max_{\mathbf{z} \in \mathcal{Z}} D\left[p_1(\mathbf{y}|g(\mathbf{z})) \| p_2(\mathbf{y}|g(\mathbf{z}))\right] + \lambda \log p(\mathbf{z}) \tag{3}$$

where λ is some regularisation constant which trades off flexibility of the optimisation problem with the likelihood of the generated sample under $\tilde{p}(\mathbf{x})$. The intuition behind this additional regularisation term is that, provided $p(\mathbf{x}) \approx \tilde{p}(\mathbf{x})$, for any latent variable $\mathbf{z}$ with high-probability under $p(\mathbf{z})$, $g(\mathbf{z})$ is high-probability under $\tilde{p}(\mathbf{x})$.

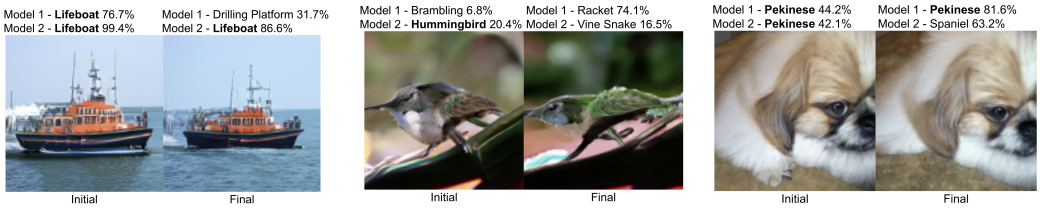
By optimising in latent space, we ensure samples are from the true data manifold. As latent spaces are typically designed to be low-dimensional, our optimisation offers a computationally cheap way to generate such points. Our overall GeValDi framework is then the following: given two discriminative classifiers and a generative model, we create MDS that maximise divergence between classifier predictions.

4 Experiments

We now probe the ability of GeValDi to generate realistic synthetic images for which the predictions of two high-performing, pre-trained classifiers differ. For the generative model, we employ a variational autoencoder (VAE) (Kingma & Welling, 2019) and pre-trained BigGAN (Brock et al., 2018) for experiments on MNIST (Deng, 2012) and ImageNet (Deng et al., 2009), respectively. For a complete list of experimental details, see Appendix B.

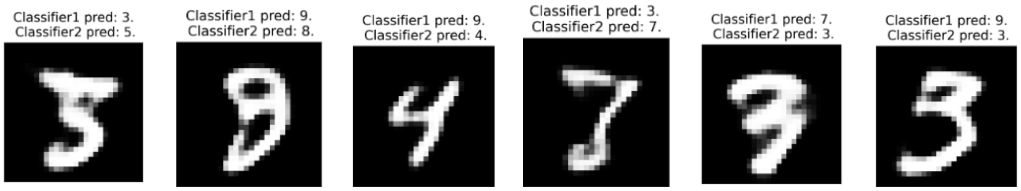
4.1 Examples of Maximally Different Samples

In Figures 1 and 2 we compare the optimised MDS with the image corresponding to the pre-optimised latent variable for ImageNet and MNIST, together with the predicted class probabilities for the two pre-trained classifiers. In both figures, we see that the optimised MDS remain photo-realistic whilst deviating noticeably from the pre-optimised image, indicating the latent space optimisation works as intended. The resulting MDS are able to expose intriguing characteristics of the classifiers involved. For example, in Figure 1(a) the final image is clearly a lifeboat, yet model 1 incorrectly classifies it as a drilling platform. This demonstrates the utility of our approach in exposing potential shortcomings of high-performance pre-trained classifiers using synthetic data. However, it is also important to note that there also exist cases where the true class label of the MDS is not so obvious. Examples include Figures 2(a) and 2(b). Please see Appendix C for more examples.



(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 is used to highlight the BigGAN class-label used to generate samples.



(a) (b) (c) (d) (e) (f)

Figure 2: Each figure shows images generated by the VAE (Kingma & Welling, 2019) after latent space optimisation. Text on top gives the predictions by Classifier 1 and 2.

4.2 Predictive Class Path

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. For instance, the distance between two dog breeds is closer than that of a dog and an inanimate object (e.g., a table). 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.

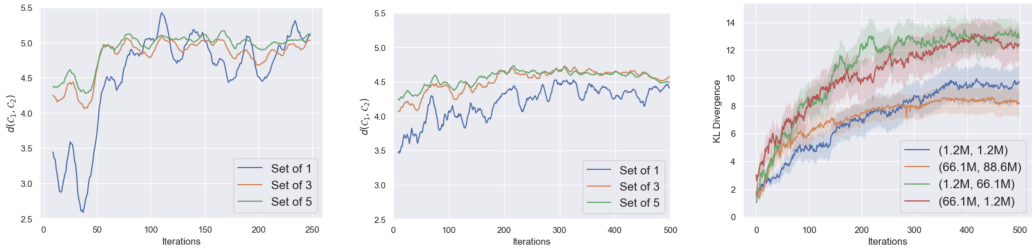
$$d(\mathcal{C}_1, \mathcal{C}_2) = \frac{1}{|\mathcal{C}_1||\mathcal{C}_2|} \sum_{c_i \in \mathcal{C}_1} \sum_{c_j \in \mathcal{C}_2} H_w(c_i, c_j) \tag{4}$$

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*.

Figure 3(a) illustrates the set distance path between $\mathcal{C}_1$ and $\mathcal{C}_2$ for different set sizes (1, 3, and 5). Initially, set distances increase with set size because larger sets span larger portions of the hierarchy. However, interestingly, set distance decreases with set size across iterations. This is because the disagreement between predicted classes in a set reduces the overall set distance between $\mathcal{C}_1$ and $\mathcal{C}_2$. In Figure 2(b), the set distance between predicted labels of the "Brambling" and "Hummingbird" models is 2.5 at the initial point. However, it increases to 9.5 in iteration 500 for "Racket" and "Vine Snake". Qualitatively, MDS enables users to gain insight into the behavior of two models not only probabilistically, but also conceptually in a more interpretable way.

4.3 Classifier Expressivity

When using the MDS algorithm, we are finding directions in latent space that maximise KL divergence between model predictions. This means that our latent space path has a



(a) (b) (c)

Figure 3: (a) and (b) show evolution of set path distance for $n = 1, 3, 5$ for 2 pairs of models, with (number of parameters in model 1, number of parameters in model 2) being (1.2 million, 1.2 million) and (66.1 million, 88.6 million) respectively. (c) shows evolution of KL divergence across iterations for four pairs of models with different capacities.

high dependence on model structure and parameters. So, we need to consider the effects of variations in the models involved. We consider classifiers of varying expressivity (Hu et al., 2021) to understand how it affects the MDS generated. The expressivity of a classifier is the size of the set of functions spanned by its parameters Bengio & Delalleau (2011); Liang et al. (2019); Montufar et al. (2014); Raghu et al. (2017). In particular, we use the number of parameters as a proxy for classifier expressivity here. To explore this, we do 4 experiments, with details in Appendix B.

Figure 3(c) shows the KL divergence path across iterations for our four experiments. From this, we see that increasing the difference in expressivity of classifier pairs increases both the convergent value and rate of growth of KL divergence. Figures 3(a) and 3(b) show that increasing classifier expressivity decreases the overall set distance between classifier predictions. This is especially evident in comparing the set distance of top one predictions, where the increase in set distance grows in the following order: comparing two models of high expressivity, comparing one model of high expressivity vs one model of low expressivity, comparing two models of low expressivity. An important implication of this is that classifiers with low expressivity have a higher probability of producing perceptually inconsistent predictions than models with higher expressivity. Since classifiers implicitly learn label hierarchies by learning the sets of predictions that are most probable, this implication is especially critical, as we can perform model selection by contrasting how well two models learn the hierarchy.

5 Conclusion

We propose GeValDi as a method to circumvent the need for extensive amounts of validation data to compare two classifiers. Instead, we generate samples from the true data distribution, where the predicitons of two classifiers maximally differ. Our method is able to identify failure modes – including differences in pairs of models' *paths through label space hierarchies* – offering a novel way to evaluate classifiers. Further, we observe that expressivity affects the divergence of predictions and hierarchical set distances, pointing to the possibility that models with higher expressivity are able to learn the label hierarchy better than models with lower expressivity (Cerri et al., 2014; Sadat & Caragea, 2022).

Next steps with GeValDi include running human subject experiments (HSEs) on MDS samples to study how the samples alter human preferences about model selection, as well as testing the method on other modalities (e.g., audio data). With MDS, we can not only find and correct failure modes of high-performance classifiers, but also make informed decisions on which models have best learnt label hierarchies.

References

Javier Antorán, Umang Bhatt, Tameem Adel, Adrian Weller, and José Miguel Hernández-Lobato. Getting a clue: A method for explaining uncertainty estimates. 2020. doi: 10.48550/ARXIV.2006.06848. URL https://arxiv.org/abs/2006.06848.

Yoshua Bengio and Olivier Delalleau. On the expressive power of deep architectures. In *Algorithmic Learning Theory: 22nd International Conference, ALT 2011, Espoo, Finland, October 5-7, 2011. Proceedings 22*, pp. 18–36. Springer, 2011.

Emily Black, Manish Raghavan, and Solon Barocas. Model multiplicity: Opportunities, concerns, and solutions. *2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022. doi: 10.1145/3531146.3533149.

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.

Ricardo Cerri, Rodrigo C. Barros, and André C.P.L.F. de Carvalho. Hierarchical multi-label classification using local neural networks. *Journal of Computer and System Sciences*, 80 (1):39–56, 2014. ISSN 0022-0000. doi: https://doi.org/10.1016/j.jcss.2013.03.007. URL https://www.sciencedirect.com/science/article/pii/S0022000013000718.

Amanda Coston, Ashesh Rambachan, and Alexandra Chouldechova. Characterizing fairness over the set of good models under selective labels. In *International Conference on Machine Learning*, pp. 2144–2155. PMLR, 2021.

Antonia Creswell, Anil A Bharath, and Biswa Sengupta. Latentpoison-adversarial attacks on the latent space. *arXiv preprint arXiv:1711.02879*, 2017.

Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 99–108, 2004.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, dec 2006. ISSN 1532-4435.

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.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. 2014. doi: 10.48550/ARXIV.1412.6572. URL https://arxiv.org/abs/1412.6572.

Markus Hittmeir, Andreas Ekelhart, and Rudolf Mayer. On the utility of synthetic data: An empirical evaluation on machine learning tasks. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, ARES '19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450371643. doi: 10.1145/3339252.3339281. URL https://doi.org/10.1145/3339252.3339281.

Xia Hu, Lingyang Chu, Jian Pei, Weiqing Liu, and Jiang Bian. Model complexity of deep learning: A survey. 2021. doi: 10.48550/ARXIV.2103.05127. URL https://arxiv.org/abs/2103.05127.

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.

Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-rao metric, geometry, and complexity of neural networks. In *The 22nd international conference on artificial intelligence and statistics*, pp. 888–896. PMLR, 2019.

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.

Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. *Advances in neural information processing systems*, 27, 2014.

Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *international conference on machine learning*, pp. 2847–2854. PMLR, 2017.

Aaron Roth, Alexander Tolbert, and Scott Weinstein. Reconciling individual probability forecasts. 2022. doi: 10.48550/ARXIV.2209.01687. URL https://arxiv.org/abs/2209.01687.

Mobashir Sadat and Cornelia Caragea. Hierarchical multi-label classification of scientific documents. 2022. doi: 10.48550/ARXIV.2211.02810. URL https://arxiv.org/abs/2211.02810.

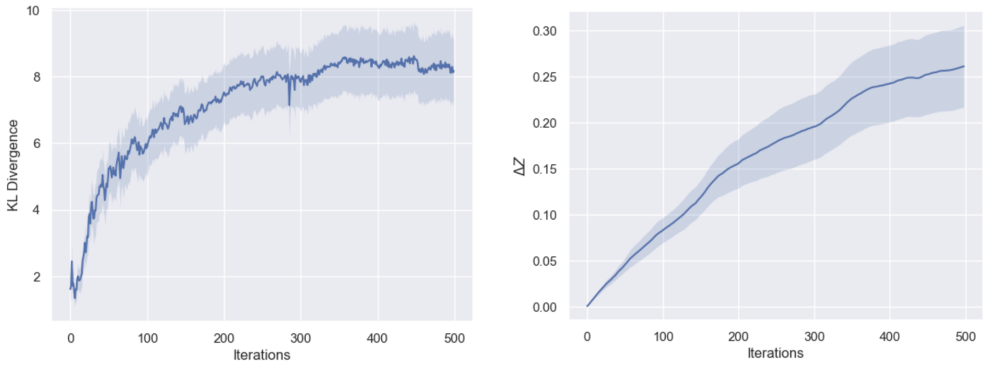
Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. 2013. doi: 10.48550/ARXIV.1312.6199. URL https://arxiv.org/abs/1312.6199.

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.

Ujjwal Upadhyay and Prerana Mukherjee. Generating out of distribution adversarial attack using latent space poisoning. *IEEE Signal Processing Letters*, 28:523–527, 2021.

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.

Roozbeh Yousefzadeh and Dianne P. O'Leary. Interpreting neural networks using flip points. 2019. doi: 10.48550/ARXIV.1903.08789. URL https://arxiv.org/abs/1903.08789.



(a) (b)

Figure 4: (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

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 4(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 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 4(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

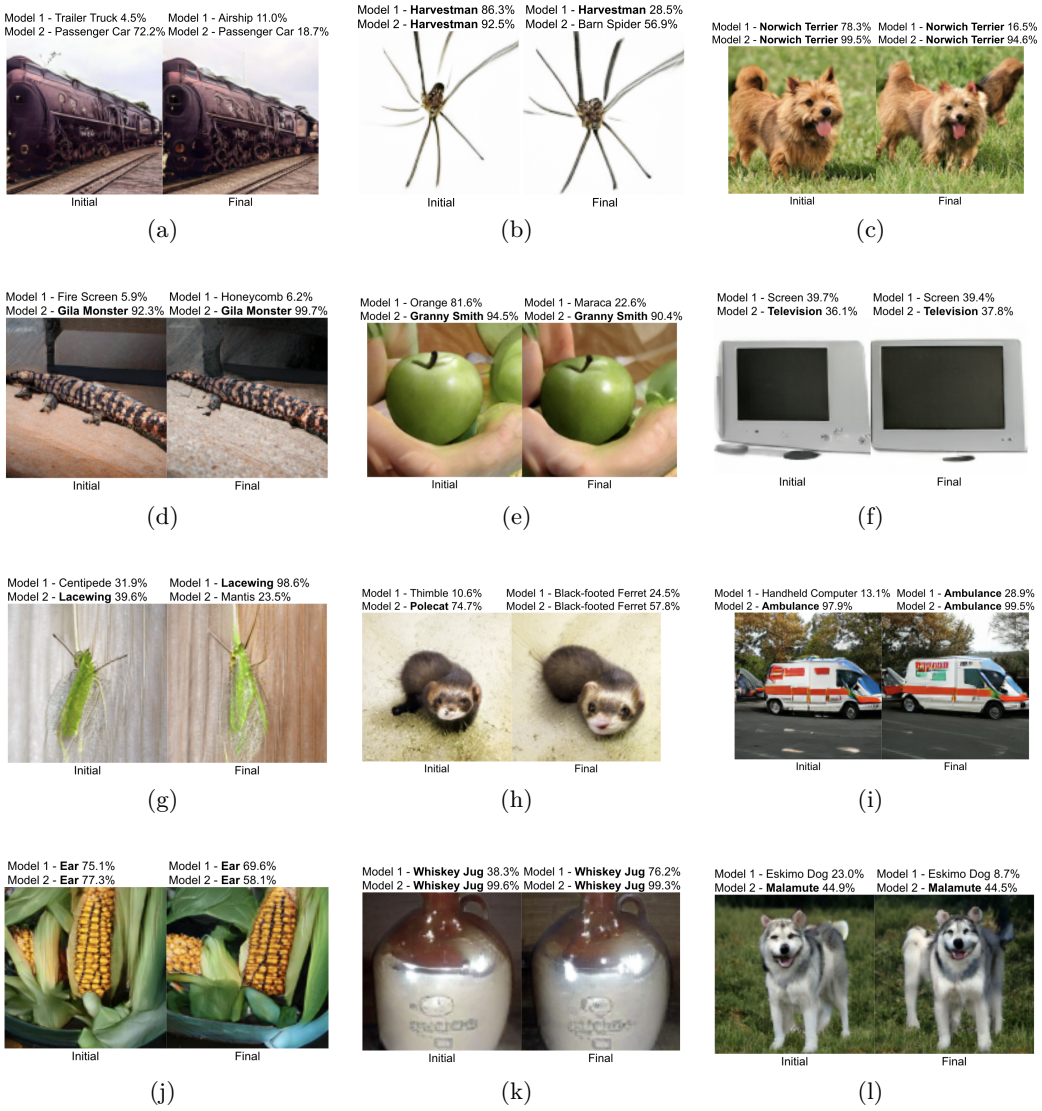


Figure 5: Each figure shows images generated by the GAN before (left) and after (right) the latent space optimisation for Model 1 (GoogleNet) vs Model 2 (AlexNet). Bolded text is used to highlight the BigGAN class-label used to generate samples.

- 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 Additional MDS Examples

In this section, we present several more examples of MDS in Figure 5, 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 5(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 5(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 5(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 5(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.