On Gradients of Deep Generative Models for Representation-Invariant Anomaly Detection

Sam Dauncey [1], **Chris Holmes** [2,3], **Christopher Williams** [2] **& Fabian Falck** [2,3]
[1]University of Edinburgh, [2]University of Oxford, [3]The Alan Turing Institute
s.dauncey@ed.ac.uk, {cholmes, williams, fabian.falck}@stats.ox.ac.uk

Abstract

Deep generative models learn the distribution of training data, enabling to recognise the structures and patterns in it without requiring labels. Likelihood-based generative models, such as Variational Autoencoders (VAEs), flow-based models and autoregressive models, allow inferring the log-likelihood of a given data point and sampling from the learned distribution. A well-known fact about all of these models is that they can give higher log-likelihood values for structured out-of-distribution (OOD) data than for in-distribution data that they were trained on, rendering likelihood-based OOD detection infeasible. We provide further evidence for the hypothesis that this is due to a strong dependence on the counter-intuitive nature of volumes in the high-dimensional spaces under which one chooses to represent the input data, and provide theoretical results illustrating that the gradient of the log-likelihood is invariant under this choice of representation. We then present a first gradient-based anomaly detection method which exploits our theoretical results. Experimentally, our proposed method performs well on image-based OOD detection, illustrating its potential.

1 Introduction

Neural networks can be highly confident but incorrect when given inputs very different to the distribution of data they were trained on (1). One approach to make such a machine learning system more robust is by first filtering out out-of-distribution (OOD) inputs before deploying them in the wild, especially for safety-critical applications. Many deep generative models are straight-forward candidates for this task as they can learn to infer the likelihood of a given data point under the training distribution. However, (2) found that many such likelihood-based deep generative models such as flow-based models, variational autoencoders (VAEs), and PixelCNNs will give higher log-likelihoods for OOD data points than for a large fraction of in-distribution or training data points (see Fig. 3 in appendix B where we replicate these results). This result is very surprising given that these models are trained to maximise the log-likelihood of the training data, and issues persist even when the models can generate realistic-looking samples from their learned distribution. It marks an open problem of using deep generative models for OOD detection, rendering their direct application based on the likelihood metric infeasible.

Recent work tackles this problem mainly from two angles: (a) explaining why the log-likelihoods from these models fail to discriminate, and whether this indicates that these models do not fully understand the structure of their training data (3; 4; 5), and (b) proposing likelihood-based anomaly detection methods or adaptations which may overcome these shortcomings (6; 7; 8; 9; 10; 11; 12). With respect to (a), Le Lan et al. note that choosing a different way of representing the data can lead to very different likelihood estimates. Changing the representation affects the volumes of sets in our data space, meaning that the probability density associated with the points in these sets would have to increase or decrease (by the change-of-variables formula which we discuss in Sec. 2.2).

Assuming this was the primary cause of the observed pathologies of likelihood-based deep generative models better modelling of the distribution would neither solve this problem, as Le Lan et al. prove, there would always be some representation under which higher likelihoods are assigned to OOD data. Concluding, Le Lan et al. propose the *principle of invariance* desideratum: a given anomaly detection method should not depend on how the data is represented (4).

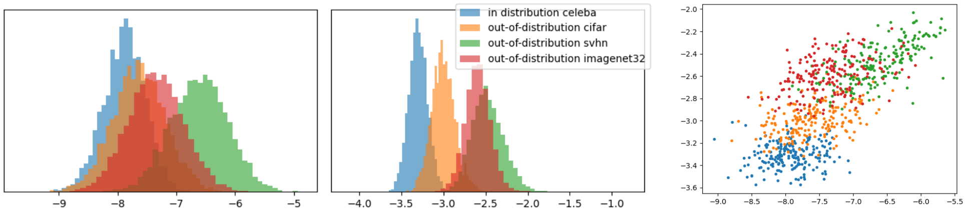

Figure 1: Layer-wise gradients of the log-likelihood, the score, is highly informative for OOD detection. We select 2 layers [left and middle] from the 1280 layers λ of a Glow (13) model trained on CelebA, a dataset that has previously proved challenging for OOD detection in previous work (12). We then evaluate this model on batches of five samples $\boldsymbol{x}_1 \ldots \boldsymbol{x}_5$ from several OOD image datasets and compute the layer-wise L^2-norm gradients of the log-likelihood $f_\lambda(\boldsymbol{x}_1 \ldots \boldsymbol{x}_5) = \left\|\nabla_{\boldsymbol{\theta}_\lambda}(\log p^{\boldsymbol{\theta}}(\boldsymbol{x}_1) + \cdots + \log p^{\boldsymbol{\theta}}(\boldsymbol{x}_5))\right\|^2$ (x-axis of histogram is on log-scale, the norms for the two layers are plotted against each other in a scatter plot [right]).

With respect to (b), most recently proposed methods such as Typicality (12) and energy-based models (10) *do* depend on the representation of data, and hence suffer from the observations made by Le Lan et al. . One current method which does satisfy representation invariance is likelihood ratios (6), which require also modelling a “background” distribution of all structured data to compare to the model of the in distribution data, and in practical scenarios, choosing a background distribution presents a challenge which does not necessarily generalise well across data domains. This paper presents the theoretical result that the gradient of the log-likelihood, also known as the *score* vector, is representation-invariant. We subsequently provide further theoretical results justifying and motivating its use for anomaly detection, and experimentally demonstrate that the score is informative in this regard. In appendix E we analyse the small amount of previous work on gradient-based anomaly detection, linking seemingly independent discoveries of other authors.

The preliminary results we present show promise for the use of gradient-based methods of anomaly detection, and also provide theoretical foundations for expanding on their use. With more results across a more diverse range of data sets and models, this would show that these are reliable methods for using deep generative models to draw an in vs out of distribution decision boundary which agrees with the intuitive semantic properties of the modelled distribution.

2 Theory

2.1 Motivation for the use of gradients in anomaly detection

In regression analysis, the leverage of a given data point, also known as the hat-value, is defined by the derivative $\frac{d\hat{y}}{dy}$ of the model’s prediction $\hat{y}$ with respect to the observation y which the model has been fitted to. Here, the idea is that high-leverage points, which can be intuitively understood as points for which inclusion in the fitting process has a large effect on the resulting model parameters, can be classified as outliers. We extend this notion to deep learning by considering gradient of a deep generative models w.r.t. its parameters, its score vector. In Fig. 1, we summarise its information through its layer-wise norm, here illustrated for two layers of a flow-based Glow (13) model trained on CelebA, computed for both in-distribution and OOD data points. Interestingly, we find a similar effect as for hat-values: The layer-wise score norms seem to differ between OOD and in-distribution data points, here being higher for OOD data points, and are hence informative. In the following, we will use this key insight to construct a simple method for the purpose of anomaly detection.

2.2 Invariance of the score vector under invertible transformations

Suppose we have set $\mathcal{X}$ in which our data points $\boldsymbol{x} \in \mathcal{X}$ are represented. If we endow $\mathcal{X}$ with a model distribution for our data $p^{\boldsymbol{\theta}}$ and a way to measure volumes we can define a probability density function in $\mathcal{X}$ which we will denote by $p^{\boldsymbol{\theta}}_{\mathcal{X}}(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{X}$. Suppose that we instead choose to represent our data points as belonging to some other set $\mathcal{T}$ with $T : \mathcal{X} \rightarrow \mathcal{T}$ being an invertible transformation that links these representations. We see that the probability density for a given point $\boldsymbol{t} = T(\boldsymbol{x})$ may be *different* from the probability density of $\boldsymbol{x}$ as the density is being scaled by the determinant Jacobian of T via the change-of-variables formula

$$p^{\boldsymbol{\theta}}_{\mathcal{T}}(\boldsymbol{t}) = p^{\boldsymbol{\theta}}_{\mathcal{X}}(\boldsymbol{x}) \left| \frac{\partial T^{-1}}{\partial \boldsymbol{x}} \right| . \tag{1}$$

Intuitively, the determinant Jacobian of T measures a change of volumes between the spaces $\mathcal{X}$ and $\mathcal{T}$. This result shows that merely changing the data representation may lead to contradicting likelihood estimates for deep generative models. However, if we do a log-transform and take the gradient with respect to the model parameters $\boldsymbol{\theta}$ on both sides of Eq. 1, we get the scores

$$\nabla_{\boldsymbol{\theta}}\{\log p^{\boldsymbol{\theta}}_{\mathcal{T}}\}(\boldsymbol{t}) = \nabla_{\boldsymbol{\theta}}\{\log p^{\boldsymbol{\theta}}_{\mathcal{X}}\}(\boldsymbol{x}), \tag{2}$$

noting that the determinant Jacobian is not dependent on $\boldsymbol{\theta}$. From this straight-forward calculation, we hence observe that the score vector is the same whether computed under $\mathcal{T}$ or $\mathcal{X}$. In other words, the score vector is satisfies satisfying the principle of invariance, i.e. it is not affected by how we choose to represent our data. This result also implies that optimization with stochastic gradient descent, which is used for learning deep generative models, is likewise representation-invariant, potentially explaining why the likelihood can still be used to train deep generative models and allow them to generate convincing samples, even though the likelihood itself appears uninformative. In Appendix A we show a similar invariance result applies for the Evidence Lower Bound (ELBO) of a VAE.

2.3 The Fisher Information Metric

We are now interested in formulating a method which uses the intuitively plausible (see Sec. 2.1) and data representation-invariant (see Sec. 2.2) score vector $\nabla_{\boldsymbol{\theta}} l(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}}\{\log p^{\boldsymbol{\theta}}_{\mathcal{X}}\}(\boldsymbol{x})$ for anomaly detection. A naïve, highly summarising approach would be to measure the size of the score vector by computing the L^2 norm $\|\nabla_{\boldsymbol{\theta}} l(\boldsymbol{x})\|_2^2$ as similarly discussed in (14). However, we empirically find that the size of this value is dominated by certain layers and parameters which have very large values, destroying the signal from other layers and parameters, as can be seen in figure 1 where the L^2-norm of the gradients for the layer on the left are orders of magnitude lower than those for the layer on the right. A perhaps mathematically more natural way to measure the score vector's size is to use the norm induced by the *Fisher information metric* (15), defined as

$$\|\nabla_{\boldsymbol{\theta}} l(\boldsymbol{x})\|_{FIM} = \nabla_{\boldsymbol{\theta}} l(\boldsymbol{x})^T F_{\boldsymbol{\theta}}^{-1} \nabla_{\boldsymbol{\theta}} l(\boldsymbol{x}), \tag{3}$$

where $F_\theta = E_{\boldsymbol{y} \sim p^{\boldsymbol{\theta}}}(\nabla_{\boldsymbol{\theta}} l(\boldsymbol{y}) \nabla_{\boldsymbol{\theta}} l(\boldsymbol{y})^T)$ is called the *Fisher Information Matrix (FIM)*. Intuitively, the FIM re-scales the gradients to give more equal weighting to the parameters which typically have smaller gradients, and is in fact independent of how the parameters are scaled. This is a result of the fact that the Fisher information metric is independent of how the model distribution is parameterized, in theory preventing dependence on representation in the "gradient space". Rao (1948) showed that, assuming the model parameters $\boldsymbol{\theta}$ are maximum likelihood estimates, $\|\nabla_{\boldsymbol{\theta}} l(\boldsymbol{x})\|_{FIM}$ should follow a χ^2 distribution, and thus can be used for a statistical test known as the *score test* (15).

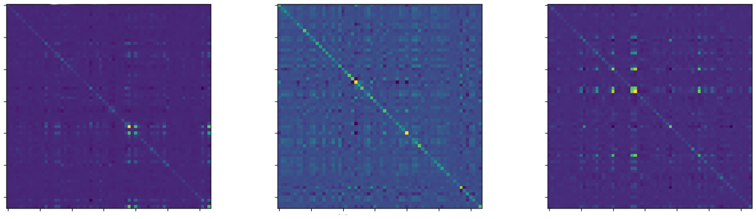

Figure 2: *Diagonal dominance of the FIM.* The three plots show heatmaps of Monte-Carlo approximations of FIMs for different layers λ in a PixelCNN (16) model. Lighter colours in cell ij correspond to higher values for $(F_{\boldsymbol{\theta}_\lambda})_{ij}$. Details of the approximation are in Appendix B

Table 1: AUROC values for our method (with batch size $B = 1$) compared to raw likelihood estimates using Glow models trained and tested on different image dataset pairings.

	Our method ($B = 1$)				Negative log likelihood $-l(\boldsymbol{x})$			
test dataset ↓ *train dataset* →	CIFAR-10	SVHN	CelebA	ImageNet32	CIFAR-10	SVHN	CelebA	ImageNet32
CIFAR-10	-	0.96	0.95	0.49	-	0.99	0.76	0.41
SVHN	0.82	-	0.99	0.56	0.08	-	0.08	0.06
CelebA	0.47	0.98	-	0.40	0.60	0.99	-	0.38
ImageNet32	0.65	1.00	0.99	-	0.65	1.00	0.99	-

3 Methodology

In practise, the FIM is too large to store for most deep generative models as it is a $P \times P$ matrix, where $P = |\boldsymbol{\theta}|$ is the number of parameters of the model. In Figure 1 [right], we illustrate a scatter plot of the gradients of Glow (13) for two layers (the two colors) for in-distribution and OOD datapoints. We observe that *within* a given neural network layer the gradients are of a similar order of magnitude and *across* layers, the size of the gradients tends to not correlate strongly. This indicates that each layer's gradient, rather than the gradient's overall size is informative for anomaly detection, and simply using the score's norm as done in (14) would not utilise this information. With this in mind, we decide to split our score vector layer-by-layer and take the L^2 norm of the score vector over each layer which we extract as features for anomaly detection. Our method generalises to an OOD test for batches $\boldsymbol{x}_1, \boldsymbol{x}_2 \dots \boldsymbol{x}_B$ of datapoints drawn from the same distribution. Setting $B = 1$ gives a method for deciding, given only one datapoint, whether to classify it as OOD. Specifically, for a given layer λ, we define our features f_λ as

$$f_\lambda(\boldsymbol{x}_1, \boldsymbol{x}_2 \dots \boldsymbol{x}_B) = \left\|\nabla_{\boldsymbol{\theta}_\lambda} \left(l(\boldsymbol{x}_1) + l(\boldsymbol{x}_2) + \cdots + l(\boldsymbol{x}_B)\right)\right\|^2, \tag{4}$$

where $\boldsymbol{\theta}_\lambda$ are the parameters for layer λ. This corresponds to approximating the FIM restricted to a given layer $F_\lambda^{\boldsymbol{\theta}}$ as a multiple of the identity matrix. In Fig. 2 we illustrate the FIM for three small layers λ of a PixelCNN (16) model and show a moderate diagonal dominance, justifying this somewhat crude approximation. We can combine all layer-wise L^2 norm features as computed in Eq. 4 and provide them as input to a decision algorithm which classifies anomalies. One choice would be to sum up all layer-wise L^2 norms and decide based on a threshold of the summed magnitude, which would be similar to the approach taken in the concurrent work in (8), where the authors approximate the FIM as diagonal and try to directly approximate the score statistic. A choice which we opt for here is to use a simple generative model, namely a one-class Support Vector Machine (SVM), which would allow for a more flexibility in the face of assumptions not always being met: For example, in rare cases we empirically observe the out-of-distribution gradient L^2 norms in a layer can be *higher* than the in-distribution L^2 norms, as illustrated in Figure 4 in Appendix B, in these cases a threshold-based approach would fail. We provide our proposed method as algorithms in Appendix C.

4 Experiments

We evaluate our method using the Area Under the Receiver Operating Curve (AUROC) on image datasets. We train a Glow (13) model on the same image datasets used in related work (2; 17; 7), namely CIFAR-10, SVHN, CelebA and ImageNet32, and then fit a OneClassSVM to features extracted via Eq. 4 from in-distribution data. Our results for single sample OOD detection ($B = 1$) are reported in Table 1, and are compared to the results when using raw negative log-likelihood $-l(\boldsymbol{x})$ as an anomaly score. In particular, we observe good performance for a model trained on CelebA and evaluated on CIFAR-10, a dataset pairing for which most likelihood-based methods have poor or have not reported performance for, as first reported (12). We observe poor performance for single sample OOD detection for Glow models trained on more visually diverse datasets such as CIFAR-10 and ImageNet32. However, we would like to challenge the assumption that one should expect a single sample OOD detection method trained on such visually diverse, highly overlapping data distributions the reject the other as OOD. We also present results for our proposed method as used in a goodness-of-fit test ($B = 5$) in Table 2, where we observe excellent performance for all training and test dataset pairings. Appendix D presents our results for PixelCNN and Glow models trained on FashionMNIST, compared against the results from (7) and (8).

ACKNOWLEDGEMENTS

We thank Joost van Amersfoort for his code repository and useful discussions. Chris Holmes acknowledges support from the Medical Research Council Programme Leaders award MC_UP_A390_1107, The Alan Turing Institute, Health Data Research, U.K., and the U.K. Engineering and Physical Sciences Research Council through the Bayes4Health programme grant. Christopher Williams acknowledges support from the Defence Science and Technology (DST) Group and from an EPSRC DTP Studentship. Fabian Falck acknowledges the receipt of studentship awards from the Health Data Research UK-The Alan Turing Institute Wellcome PhD Programme (Grant Ref: 218529/Z/19/Z), and the Enrichment Scheme of The Alan Turing Institute under the EPSRC Grant EP/N510129/1.

REFERENCES

[1] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[2] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019.

[3] Lily Zhang, Mark Goldstein, and Rajesh Ranganath. Understanding failures in out-of-distribution detection with deep generative models. In *International Conference on Machine Learning*, pages 12427–12436. PMLR, 2021.

[4] Charline Le Lan and Laurent Dinh. Perfect density models cannot guarantee anomaly detection. *Entropy*, 23(12), 2021.

[5] Anthony L. Caterini and Gabriel Loaiza-Ganem. Entropic issues in likelihood-based OOD detection. In Melanie F. Pradier, Aaron Schein, Stephanie Hyland, Francisco J. R. Ruiz, and Jessica Z. Forde, editors, *Proceedings on "I (Still) Can't Believe It's Not Better!" at NeurIPS 2021 Workshops*, volume 163 of *Proceedings of Machine Learning Research*, pages 21–26. PMLR, 13 Dec 2022.

[6] Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[7] Hyunsun Choi, Eric Jang, and Alexander A Alemi. Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018.

[8] Federico Bergamin, Pierre-Alexandre Mattei, Jakob Drachmann Havtorn, Hugo Senetaire, Hugo Schmutz, Lars Maaløe, Soren Hauberg, and Jes Frellsen. Model-agnostic out-of-distribution detection using combined statistical tests. In *International Conference on Artificial Intelligence and Statistics*, pages 10753–10776. PMLR, 2022.

[9] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. Deep anomaly detection with outlier exposure. *CoRR*, abs/1812.04606, 2018.

[10] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

[11] Jakob D. Havtorn, Jes Frellsen, Søren Hauberg, and Lars Maaløe. Hierarchical vaes know what they don't know. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4117–4128. PMLR, 18–24 Jul 2021.

[12] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, and Balaji Lakshminarayanan. Detecting out-of-distribution inputs to deep generative models using typicality, 2019.

[13] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[14] Quoc Phong Nguyen, Kar Wai Lim, Dinil Mon Divakaran, Kian Hsiang Low, and Mun Choon Chan. Gee: A gradient-based explainable variational autoencoder for network anomaly detection. In *2019 IEEE Conference on Communications and Network Security (CNS)*, pages 91–99. IEEE, 2019.

[15] C. Radhakrishna Rao. Large sample tests of statistical hypotheses concerning several parameters with applications to problems of estimation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 44(1):50–57, 1948.

[16] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1747–1756, New York, New York, USA, 20–22 Jun 2016. PMLR.

[17] Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F Núñez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. In *International Conference on Machine Learning*, 2020.

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of The 33rd International Conference on Machine Learning*, 2014.

[19] Gukyeong Kwon, Mohit Prabhushankar, Dogancan Temel, and Ghassan AlRegib. Backpropagated gradient representations for anomaly detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[20] Tommi Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11. MIT Press, 1998.

[21] Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. *Advances in Neural Information Processing Systems*, 34:677–689, 2021.

[22] Jinsol Lee and Ghassan AlRegib. Gradients as a measure of uncertainty in neural networks. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 2416–2420. IEEE, 2020.

[23] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

Table 2: AUROC values for our method (with batch size $B = 5$) compared to raw likelihood estimates using Glow models trained and tested on different image dataset pairings.

test dataset ↓ *train dataset* →	CIFAR-10	SVHN	CelebA	ImageNet32
CIFAR-10	-	1.00	1.00	0.83
SVHN	1.00	-	1.00	1.00
CelebA	0.99	1.00	-	0.97
ImageNet32	0.99	1.00	1.00	-

A Representation-Invariance of the gradient of the ELBO

Assume the same setup as in section 2.2, but this time with a Variational AutoEncoder (18) with decoder probability density $p^{\boldsymbol{\theta}}_{\mathcal{X}}(\boldsymbol{x}|\boldsymbol{z})$ and encoder probability density $q^{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, noting that the decoder probability density is that which depends on $\mathcal{X}$. The Evidence Lower Bound on the log-likelihood $p^{\boldsymbol{\theta}}_{\mathcal{X}}(\boldsymbol{x})$ is given by:

$$ELBO^{\boldsymbol{\theta},\boldsymbol{\phi}}_{\mathcal{X}}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{z}\sim q^{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left(\log \frac{p^{\boldsymbol{\theta}}_{\mathcal{X}}(\boldsymbol{x},\boldsymbol{z})}{q^{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right)$$

Noting that $\quad p^{\boldsymbol{\theta}}_{\mathcal{T}}(\boldsymbol{t},\boldsymbol{z}) = p^{\boldsymbol{\theta}}_{\mathcal{X}}(\boldsymbol{x},\boldsymbol{z})\left|\frac{\partial T^{-1}}{\partial \boldsymbol{x}}\right| \quad$ while $\quad q^{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{t}) = q^{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \quad$ gives that:

$$ELBO^{\boldsymbol{\theta},\boldsymbol{\phi}}_{\mathcal{T}}(\boldsymbol{t}) = ELBO^{\boldsymbol{\theta},\boldsymbol{\phi}}_{\mathcal{X}}(\boldsymbol{x}) + \log\left|\frac{\partial T^{-1}}{\partial \boldsymbol{x}}\right|$$

and taking the gradient wrt. $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ gives the result that the gradient of the ELBO with respect to the VAE's parameters is representation-invariant.

B Supplementary results

In Figure 2 we plot approximations the Fisher Information Matrix restricted to certain layers via Monte Carlo estimate, by taking $N = 1024$ samples $\boldsymbol{y}_i$ drawn from the model and computing:

$$\mathbb{E}_{\boldsymbol{y}\sim p^{\boldsymbol{\theta}}}((\nabla_{\boldsymbol{\theta}_\lambda} l(\boldsymbol{y})\nabla_{\boldsymbol{\theta}_\lambda} l(\boldsymbol{y})^T)) \approx \frac{1}{N}\left(\nabla_{\boldsymbol{\theta}_\lambda} l(\boldsymbol{y}_1)\nabla_{\boldsymbol{\theta}_\lambda} l(\boldsymbol{y}_1)^T \cdots + \nabla_{\boldsymbol{\theta}_\lambda} l(\boldsymbol{y}_N)\nabla_{\boldsymbol{\theta}_\lambda} l(\boldsymbol{y}_N)^T\right) \tag{5}$$

We observe that the approximated, layer-wise FIM has a moderate diagonal dominance, with the diagonal elements being roughly five times the size of the off-diagonal elements in absolute value, justifying our empirical FIM approximation in Eq. 4.

Fig. 3 shows a recreation of the results first described in (2) using the models we used for our experiments. Fig. **??** provides a supplementary empirical justification for our choice to split the score vector layer-by-layer.

Fig. 4[left] illustrates the rare phenomenon discussed in 3, where we see that for certain layers of a model trained on CIFAR-10, the L^2 norms of gradients from a batch of CIFAR-10 samples (in blue) are actually larger than those from CelebA.

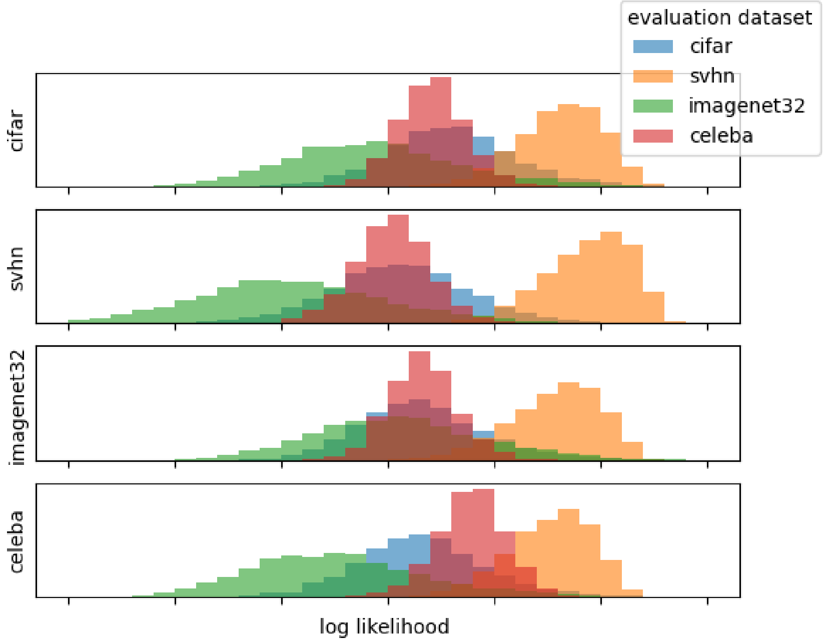

Figure 3: *Counter-intuitive properties of the log-likelihood.* Histogram of the log-likelihoods produced from Glow models trained on certain datasets and evaluated on other datasets, replicating results from (2). Note that the training dataset (labelled on the y axis) has a counter-intuitively small impact on the negative-log-likelihoods evaluated on that dataset, even though the visual qualities of the samples produced are very different. The authors of this seminal paper also observe that svhn has much higher log-likelihood assigned to it than to cifar even on an untrained model, and that the images with the highest log-likelihood assigned to them are constant.

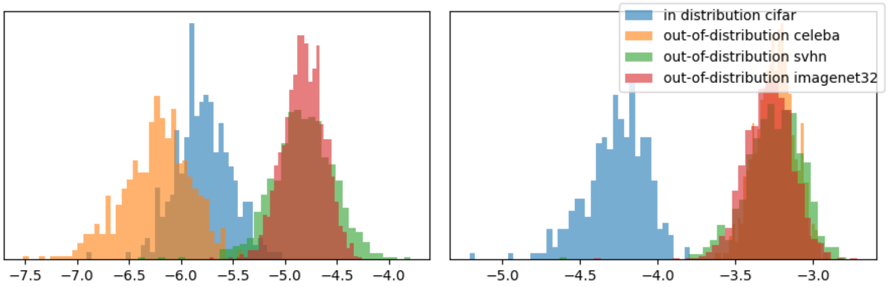

Figure 4: A replication of figure 1 but with a glow model trained on CIFAR-10 and $B = 32$ to illustrate that *occasionally for certain layers* the size of the gradients are lower for out-of-distribution samples.

C Algorithmic implementation

Algorithm 1 gives a qualitative description of how we extract features from the score vector, algorithms 2 and 3 describe how we use these features for anomaly detection. In practise, our method is straightforward to implement, and computing the features $f_1, \dots f_\ell$ requires only a few lines of PyTorch code.

Algorithm 1 Algorithm for computing features

```
Require: Deep generative model $M_D$, with parameters $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots \boldsymbol{\theta}_\ell$ in each of its $\ell$ layers.

  function GRADIENT FEATURES($\boldsymbol{x}_1 \dots \boldsymbol{x}_B$)
      for $\boldsymbol{x_i}$ in batch do
          $l(\boldsymbol{x}_i) \leftarrow M_D(\boldsymbol{x}_i)$                ▷ Compute the likelihood
      end for
      $S_{\boldsymbol{\theta}} \leftarrow \nabla_{\boldsymbol{\theta}}(l(\boldsymbol{x}_1) + \dots + l(\boldsymbol{x}_B))$   ▷ Compute the score via backpropagation
      for $\lambda \leftarrow 1 \dots \ell$ in layers do
          $f_\lambda \leftarrow \|S_{\boldsymbol{\theta}_\lambda}\|_2^2$                ▷ Store the layer-wise $L^2$ norms
      end for
  end function
```

Algorithm 2 Algorithm for training models

```
  Train a deep generative model $M_D$, with parameters $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots \boldsymbol{\theta}_\ell$ in each of its $\ell$ layers.
  for Batch $\boldsymbol{x}_1^n \dots \boldsymbol{x}_B^n$ in training data do
      $f_1^n \dots f_\ell^n \leftarrow$ GRADIENT FEATURES($\boldsymbol{x}_1^n \dots \boldsymbol{x}_B^n$)
  end for
  Fit a simple generative model $M_S$ to $f_1^n \dots f_\ell^n$ to predict the layer-wise $L^2$ norms $f_1 \dots f_\ell$
```

Algorithm 3 Algorithm for detecting anomalies

```
  Given new batch of samples $\boldsymbol{y}_1 \dots \boldsymbol{y}_B$
  $f_1 \dots f_\ell \leftarrow$ GRADIENT FEATURES($\boldsymbol{y}_1 \dots \boldsymbol{y}_B$)
  Use the anomaly score $M_S(f_1(\boldsymbol{y}) \dots f_\ell(\boldsymbol{y}))$
```

D Supplementary results

We also evaluate our method using PixelCNN (16) and Glow models trained on FashionMNIST and evaluated with AUROC on MNIST and Omniglot, with our results reported in 2, and compare them against two other methods (7) and (17). There is a lack of standardization in implementation of datasets and the underlying deep generative models which undermines the validity direct comparison of performance of anomaly detection methods across papers. For example, (8) note that the rescaling algorithm used on CelebA has a siginificant effect on performance, supported by the finding in (11) that inperceptible “low-level" features can have a significant effect on the log-likelihood.

Table 3: Table of results comparing the performance of our method on for a model trained on FashionMNIST at detecting OOD grayscale images to the performances of the S-score reported in (17) and Watanabe-Akaike Information Criterion reported in (7)

Method	MNIST	Omniglot
WAIC	0.766	0.796
S using PixelCNN++ and FLIF	0.967	1.000
PixelCNN Gradient norms (OneClassSVM) (ours)	0.979	1.000
S using Glow and FLIF	0.998	1.000
Glow Gradient norms (OneClassSVM) (ours)	0.819	1.000

E Related work on gradients for anomaly detection

With a search of the literature we found instances of authors who had seemingly independently converged on the usefulness of gradients of log-likelihoods from deep generative models for anomaly detection. Our theoretical results provide a basis for the efficacy of these similar methods, opening avenues for more pointed expansion on this work. With our experimental results, we seek to determine how much information can be extracted from the score vector alone without supplementary information from sources such as likelihood.

Concurrent work (8) uses a batch of statistical tests for anomaly detection, one of which being a score test where they approximate the Fisher Information Matrix as diagonal. In this work, we try to see how much signal can be extracted from the gradients alone, with our experimental and theoretical results nicely complementing the efficacy they find with the score statistic.

In, (14) use the L^2 norm of the difference in gradients of the ELBO from a Variational Auto Encoder $\nabla_{\boldsymbol{\theta}} l(\boldsymbol{x}) - \overline{\nabla_{\boldsymbol{\theta}} l(\boldsymbol{y})}$ as part of a batch of tests for detecting anomalous web traffic, where $\boldsymbol{x}$ is the test datapoint and $\overline{\nabla_{\boldsymbol{\theta}} l(\boldsymbol{y})}$ is the average gradient of a datapoint $\boldsymbol{y}$ that has been labelled as an attack; our method differs in that it is completely unsupervised and requires no such labels.

In (19) the authors compute the cosine similarity between the gradients in the decoder of a VAE and the typical gradients used during training as part of an anomaly score. We advocate for the use of the metrics related to the size of the score vector $\nabla_{\boldsymbol{\theta}} l(\boldsymbol{x})$ rather than metrics related to it's angle based on the intuition that for a well-trained network and the angle of the score vector $\nabla_{\boldsymbol{\theta}} l(\boldsymbol{x})$ should be close to random, limiting the amount of information that can be extracted from it.

In (12) the authors note that the Maximum Mean and Kernelized Stein Discrepancy tests they use to benchmark their typicality test are only effective when they use the Fisher kernel (20) $k(x_i, x_j) = \nabla_\theta l(x_i) \nabla_\theta l(x_j)^T$.

For completeness, we also include methods based on (non-generative) classifier models that use gradients, although our theoretical results do not apply in this case. In order to compute gradients without the target label, (21) computes the KL-divergence between the soft-max output of the classifier and a uniform distribution, then backpropagates gradients from this value and computes the L^2-norm of the gradient vector. (22) uses a similar method to compute the gradients and then trains a binary classifier on the layer wise L^2-norms of both the in-distribution and OOD data, and is hence not fully unsupervised. (23) propose a method using the gradients with respect to the *data*, which they dub ODIN, in which they backpropagate gradients to the input data to see how much of an input perturbation can change the softmax outcome from a classifier, the idea being that OOD inputs are more “fragile".