Robust outlier detection by de-biasing VAE likelihoods

Kushal Chauhan[1], Pradeep Shenoy[1], Manish Gupta[1], and Devarajan Sridharan*[2]

[1]Google Research
{kushalchauhan, shenoypradeep, manishgupt}@google.com
[2]Center for Neuroscience, and Computer Science and Automation, Indian Institute of Science
sridhar@iisc.ac.in

Abstract

Deep networks often make confident, yet, incorrect, predictions when tested with outlier data that is far removed from their training distributions. Likelihoods computed by deep generative models (DGM) are a candidate metric for outlier detection with unlabeled data. Yet, DGM likelihoods are readily biased and unreliable. Here, we examine outlier detection with variational autoencoders (VAEs), among the simplest of DGMs. We show that an analytically-derived correction ameliorates a key bias with VAE likelihoods. The bias correction is sample-specific, computationally inexpensive, and readily computed for various visible distributions. Next, we show that a well-known preprocessing technique, contrast stretching, extends the effectiveness of bias correction to improve outlier detection performance. We evaluate our approach comprehensively with nine (grayscale and natural) image datasets, and demonstrate significant advantages, in terms of speed and accuracy, over four state-of-the-art methods.

1 Introduction

Deep neural networks are notorious for their confident, yet incorrect predictions when tested with data whose statistics are far removed from the training data distribution [16]. Developing robust methods for outlier detection is, therefore, an important challenge with critical real-world implications.

Deep generative models (DGMs), like variational autoencoders (VAEs [4]) or flow-based models (e.g. Glow [3]), are increasingly used for outlier detection, especially with label-free data. Yet, several previous studies have shown that likelihoods computed by DGMs, including VAEs, are unreliable for outlier detection [13, 11, 1, 19], and are readily biased by differences in low-level image statistics [13, 11]. A few solutions have been proposed, but these suffer from computational bottlenecks [13, 19, 10], or theoretical limitations [1] (see Section 5, Related Work).

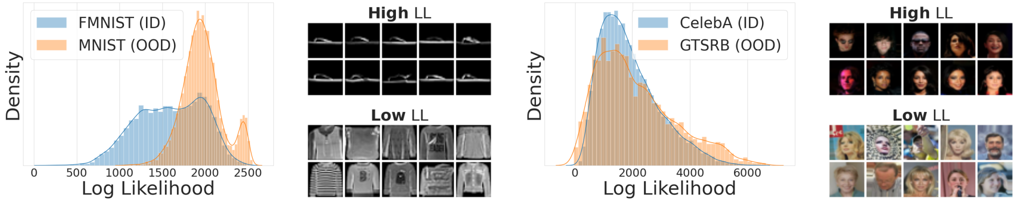
Here, we explore outlier detection with VAEs, arguably among the simplest of deep generative models. We propose two efficient remedies that achieve or approach state-of-the-art for outlier detection, both with grayscale and natural image datasets. Our key contributions are as follows:

- We present an easy-to-compute bias correction for VAE likelihoods that can be computed *post hoc* during evaluation time.
- We show that a standard preprocessing step – contrast normalization – enables bias correction to achieve state-of-the-art accuracies.
- We present, to the best of our knowledge, the most comprehensive evaluation of outlier detection with VAEs to date, with 9 datasets and 4 competing approaches [15, 13, 19, 1].

2 The challenge with outlier detection using VAE likelihoods

We illustrate the challenge of outlier detection with VAE likelihoods, by taking a fresh look at two previously reported sources of bias [13, 11].

Bayesian Deep Learning workshop, NeurIPS 2021.



(a) (b) (c) (d)

Figure 1: VAE likelihoods are unreliable for outlier detection.

Bias arising from pixel intensity. As a first example, we train a VAE on grayscale FMNIST images [18] and compute the likelihoods for in-distribution (ID) FMNIST and out-of-distribution (OOD) MNIST [6] test samples (continuous Bernoulli visible distribution [9]; model details in Appendix A). We replicate the well-known issue with VAE likelihoods: FMNIST VAE likelihoods are higher for OOD (MNIST) samples as compared to ID (FMNIST) samples (Fig. 1a). The highest likelihoods are assigned to FMNIST samples with a large number of black pixels (Fig. 1b, top row), whereas the lowest likelihoods are assigned to samples with many intermediate (gray) pixel values (Fig. 1b, bottom row), consistent with previous reports [13]. On simulated images with different (constant) pixel intensities, we find a U-shape trend in likelihood bias (Fig. 2, blue line).

Bias arising from channel variance or image contrast. Next, we train a VAE on the CelebA dataset [8], and compute ID (CelebA) and OOD (GTSRB [17]) likelihoods. Again, the VAE assigns higher likelihoods to OOD samples (Fig. 1c). Faces with dark backgrounds and high contrast between the face and background are assigned the highest likelihoods (Fig. 1d, top row), and vice versa for low contrast faces (Fig. 1d, bottom row). With simulated images, we observe that VAE likelihoods are strongly biased by contrast (Fig. 3, blue line).

3 Debiasing VAE likelihoods

To improve outlier detection with VAE likelihoods, we develop remedies for correcting for the two sources of bias discussed in the previous section.

3.1 Bias correction for pixel intensity

We develop an analytically-derived correction for VAE likelihoods. For VAEs, the marginal likelihood can be written as:

$$\log p_\theta(\mathbf{x}) = \log \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} p_\theta(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) / q_\phi(\mathbf{z}|\mathbf{x})$$

We examine the negative reconstruction error term $p_\theta(\mathbf{x}|\mathbf{z})$, assuming perfect reconstruction of the input samples by the VAE. We denote this as $p_{\mathrm{cB}}(\mathbf{x};\boldsymbol{\lambda}^*)$ where p_{cB} denotes the continuous Bernoulli pdf, and $\boldsymbol{\lambda}^*$ are optimal parameters that correspond to perfect reconstruction ($\hat{\mathbf{x}} = \mathbf{x}$). We plot $\log p_{\mathrm{cB}}(\mathbf{x};\boldsymbol{\lambda}^*)$ for simulated images in Fig. 2 (dashed green). $\log p_{\mathrm{cB}}(\mathbf{x};\boldsymbol{\lambda}^*)$ exhibits a bias that is nearly identical with the marginal likelihood (Fig. 2, blue). Thus, even if two input samples are perfectly reconstructed by the VAE, these will be assigned different likelihoods, depending on the average pixel intensity in each sample; a bias that is largely driven by the reconstruction error term.

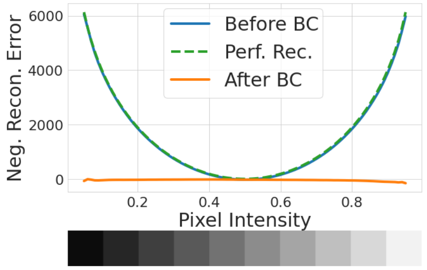


Figure 2: Bias arising from intensity.

We eliminate this bias in the reconstruction error by dividing by the error for perfect reconstruction. The "bias-corrected" marginal likelihood (BC) evaluates to:

$$\log p_\theta^c(\mathbf{x}) = \log \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\frac{p_\theta(\mathbf{x}|\mathbf{z})}{p_{\mathrm{cB}}(\mathbf{x};\boldsymbol{\lambda}^*)} \frac{p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] = \log p_\theta(\mathbf{x}) - \log p_{\mathrm{cB}}(\mathbf{x};\boldsymbol{\lambda}^*) \qquad (1)$$

The procedure for computing the correction term for the Bernoulli and continuous Bernoulli visible distributions is shown in Appendix B. For other visible distributions, an equivalent empirical correction is presented in Appendix E. Following bias correction, the bias in the negative reconstruction error is eliminated (Fig. 2, orange). We note that this correction can be computed during evaluation time and does not require retraining the VAE.

3.2 Normalization of image contrasts

For eliminating the second source of bias arising from image contrasts, we propose a standard image preprocessing step: "contrast stretching". Each image sample, for both training and testing data, is contrast normalized with the following transformation: $x_i = \min(\max(0, [x_i - a]/r), 1)$, where x_i refers to the i^{th} pixel of image $\mathbf{x}$, $r = P_{95}(\text{vec}(\mathbf{x})) - P_5(\text{vec}(\mathbf{x}))$, $a = P_5(\text{vec}(\mathbf{x}))$, P_j refers to the j^{th} percentile and vec() represents the vectorization of the input sample tensor.

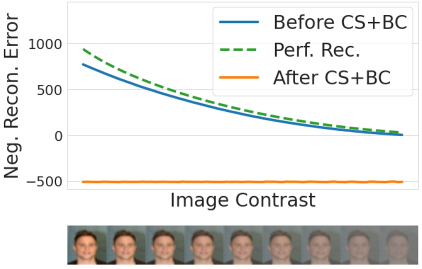


Figure 3: Bias arising from contrast.

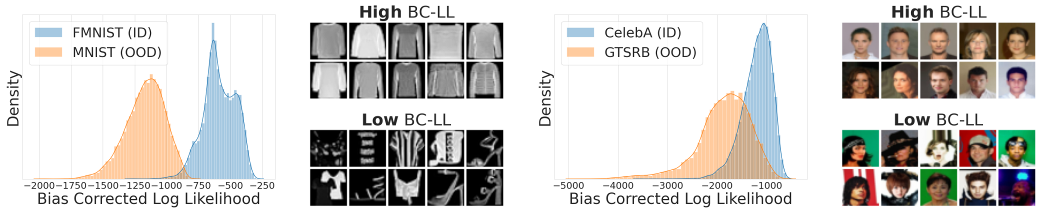
Variation in image contrasts produces systematic biases in the negative reconstruction error (Fig. 3, blue). Contrast stretching and bias correction ameliorate this bias (Fig. 3, orange).

4 Experiments

We trained and tested VAEs with 9 grayscale and natural image datasets (see Appendix A for architecture and training details). For all approaches, we report average metrics across 6 runs (3 seeds × 2 train-validation splits).

4.1 Bias-corrected likelihoods improve outlier detection

Bias correction improved OOD detection performance significantly: for the FMNIST VAE AUROC% improved from 23 to 100 and for the CelebA VAE AUROC% improved from 47 to 88. Moreover, samples that were assigned the highest and lowest bias-corrected likelihoods were visually more typical and atypical, respectively (Fig. 4b, 4d), as compared to those based on uncorrected likelihoods (Fig. 1b, 1d). Similar improvements occurred with all 9 datasets (Table 1, original LL first column, vs BC second column). We report similar improvements for additional visible distributions in Appendix E (Figs. 8, 9 and 10).



(a) (b) (c) (d)

Figure 4: Bias correction improves outlier detection.

4.2 Comparison with state-of-the-art outlier detection methods

We compare our method against four competing approaches: i) input complexity (IC) [15]), ii) likelihood ratio (LRat) [13], iii) likelihood regret (LReg) [19] and iv) WAIC (Watanabe-Akaike Information Criterion) [1]. On average, BC performed on par with, arguably, more complex state-of-the-art approaches (Table 1). We report exhaustive comparisons in a grid in Appendix C (Figs. 5 and 6). We report significant (4-100x) improvements in computation times in Appendix D (Fig. 7).

Table 1: Comparison with four competing outlier detection approaches using AUROC% for grayscale (left) and natural image datasets (right). For each ID dataset, AUROC values reflect average performance across all other (OOD) datasets in the respective table.

ID Data	LL	BC	IC	LRat	LReg	WAIC
MNIST	98	98	97	**100**	**100**	99
FMNIST	57	**100**	73	99	98	88
EMNIST	80	98	93	**100**	97	86
KMNIST	65	74	**90**	75	48	75

ID Data	LL	BC	IC	LRat	LReg	WAIC
SVHN	50	**93**	53	**93**	78	64
CelebA	73	**90**	75	63	78	73
CompCars	78	**100**	**100**	**100**	**100**	91
GTSRB	80	91	95	88	**96**	78
CIFAR10	51	63	66	47	**76**	51

5 Related work

Our work is inspired by previous studies that seek to correct for biases in generative model likelihoods. Our bias correction is, perhaps, most closely related to the work of Serra et al. (2019) [15] who proposed a correction for Glow and PixelCNN++ model likelihoods based on "input complexity" (IC). Their out-of-distribution score is computed by subtracting a sample-specific complexity estimate $L(\mathbf{x})$ from the negative log-likelihood (compare with our Equation (1)). Nonetheless, IC depends on the particular choice of a compression algorithm (e.g. PNG, JPEG2000, FLIF), which is unrelated to the VAE, whereas our correction is theoretically-grounded and contingent on the VAE decoder visible distribution. Interestingly, IC's outlier detection performance is sub-par for the Fashion-MNIST VAE (Table 1), and it also fails to distinguish in-distribution data from uniform noise for several natural image datasets (Appendix C, Fig. 6, sixth row).

Ren et al. (2019) [13] originally highlighted the problem of bias in deep generative model likelihoods, for samples with many zero-valued pixels. They proposed correcting for this bias by training a second generative model with noise-corrupted samples to capture background statistics; the "likelihood ratio" between the original and noisy VAEs provided a sensitive readout of foreground object statistics. With VAEs, our bias correction matches or outperforms the likelihood ratio score, on average (Table 1). Our bias correction also obviates the need for training multiple, duplicate models.

Similarly, Nalisnick et al. (2019) [11] originally identified the problem of bias in likelihoods arising from sample variance, and proposed a typicality test for robust outlier detection [10]. However, the typicality test works best with batches of samples, and performs relatively poorly with single samples. Interestingly, with the continuous Bernoulli visible distribution, our likelihood trends are opposite to those reported by Nalisnick et al. [11]: Samples with the lowest contrasts yield the lowest likelihoods, and vice versa, (Fig. 3), indicating that biases with VAE likelihoods depend on the choice of visible distribution.

Previous studies have also demonstrated the effectiveness of deep ensembles for outlier detection (e.g. [1, 5]). Nonetheless, many of these approaches (e.g. [5, 7]) do not work with unlabeled data. A notable exception is the WAIC score proposed by Choi et al. (2018) [1]. However, the WAIC score lacks clear theoretical arguments for its efficacy [10].

Recent work by Yong et al. (2020) [20] proposed employing bias-free Gaussian likelihoods and their variances for outlier detection. Yet, Gaussian visible distributions are not theoretically appropriate to model the finite range of pixel values (0-255) encountered in images. One way of overcoming this challenge is to discretize the Gaussian (or logistic) visible distribution with an underlying categorical representation, a solution adopted by other studies (e.g. PixelCNN++ [14]). However, in practice, such a categorical distribution also concentrates probability mass at the edges (0 and 255 values), yielding systematically biased likelihoods. Moreover, Yong et al.'s approach does not work well with natural image datasets (their Appendix C). Our contrast normalization and bias correction enable robust outlier detection, even with natural image datasets.

Xiao et al. (2020) [19] proposed a "likelihood regret" metric that involves quantifying the improvement in marginal likelihood by retraining the encoder network to obtain optimal likelihood for each sample. Such sample-specific optimizations are computationally expensive, for example, when millions of samples need to be evaluated on the fly. In contrast, our proposed metrics are readily computed with a single forward pass through pre-trained VAEs, leading to a 50-100x speedup in evaluation times over likelihood regret (Fig. 7). Interestingly, Xiao et al also showed that other state-of-the-art outlier detection metrics (e.g. IC, likelihood ratios) perform comparatively poorly with VAEs.

References

[1] Choi, H., Jang, E., and Alemi, A. A. (2018). WAIC, but Why? Generative Ensembles for Robust Anomaly Detection. *arXiv e-prints*, page arXiv:1810.01392.

[2] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[3] Kingma, D. P. and Dhariwal, P. (2018). 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.

[4] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

[5] Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[6] LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.

[7] Liang, S., Li, Y., and Srikant, R. (2018). Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

[8] Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738.

[9] Loaiza-Ganem, G. and Cunningham, J. P. (2019). The continuous bernoulli: fixing a pervasive error in variational autoencoders. 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.

[10] Nalisnick, E., Matsukawa, A., Whye Teh, Y., and Lakshminarayanan, B. (2019). Detecting Out-of-Distribution Inputs to Deep Generative Models Using Typicality. *arXiv e-prints*, page arXiv:1906.02994.

[11] Nalisnick, E. T., Matsukawa, A., Teh, Y. W., Görür, D., and Lakshminarayanan, B. (2019). Do deep generative models know what they don't know? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[12] Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In Y. Bengio and Y. LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

[13] Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., Depristo, M., Dillon, J., and Lakshminarayanan, B. (2019). 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.

[14] Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. (2017). Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[15] Serrà, J., Álvarez, D., Gómez, V., Slizovskaia, O., Núñez, J. F., and Luque, J. (2020). Input complexity and out-of-distribution detection with likelihood-based generative models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[16] Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

[17] Stallkamp, J., Schlipsing, M., Salmen, J., and Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, **32**, 323–332. Selected Papers from IJCNN 2011.

[18] Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms.

[19] Xiao, Z., Yan, Q., and Amit, Y. (2020). Likelihood regret: An out-of-distribution detection score for variational auto-encoder. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20685–20696. Curran Associates, Inc.

[20] Yong, B. X., Pearce, T., and Brintrup, A. M. (2020). Bayesian autoencoders : Analysing and fixing the bernoulli likelihood for out-of-distribution detection. *ICML Workshop on Uncertainty & Robustness in Deep Learning*.

Supplementary Material: Appendices

Appendix A: VAE architecture and training

All experiments were performed using Tensorflow 2 and Tensorflow Probability libraries. We employed a convolutional VAE architecture that follows the DCGAN [12] structure (Table 2), nearly identical with that of [19]. We used the Adam optimizer [2] with a learning rate of 5e-4 for training all of our models. Each model was trained for 1000 epochs with a batch size of 64, and the checkpoint with the best validation performance based on negative log-likelihood was used for reporting results. We used the Xavier uniform initializer (default in Tensorflow 2) for initializing network weights.

For reporting results based on the bias-corrected log-likelihood (BC score), we used a VAE with a latent dimension (nz) of size 20. The same architecture was used for training both grayscale and natural image VAEs, with two differences (grayscale: nf = 32, nc = 1; natural: nf = 64, nc = 3). Log likelihoods were estimated using the importance weighted lower bound (n=100 samples) [19].

Table 2: VAE architecture. nc: number of channels; nf: number of filters; nz: number of latent dimensions; BN: batch normalization; Conv: convolution layer; DeConv: deconvolution layer; ReLU: rectified linear unit

Encoder	**Decoder**
Input image of shape 32 × 32 × nc	Input latent code, reshape to 1 × 1 × nz
4 × 4 Conv$_{\text{nf}}$ Stride=2, BN, ReLU	4 × 4 DeConv$_{4\,\times\,\text{nf}}$ Stride=1, BN, ReLU
4 × 4 Conv$_{2\,\times\,\text{nf}}$ Stride=2, BN, ReLU	4 × 4 DeConv$_{2\,\times\,\text{nf}}$ Stride=2, BN, ReLU
4 × 4 Conv$_{4\,\times\,\text{nf}}$ Stride=2, BN, ReLU	4 × 4 DeConv $_{\text{nf}}$ Stride=2, BN, ReLU
4 × 4 Conv$_{2\,\times\,\text{nz}}$ Stride=1	4 × 4 DeConv$_{\text{nc}}$ Stride=2

Appendix B: Bias correction for Bernoulli and continuous Bernoulli decoders

Bias correction for the Bernoulli decoder. For a VAE decoder with a Bernoulli visible distribution, the negative reconstruction error is given by:

$$\begin{aligned}\log p_\theta(\mathbf{x}|\mathbf{z}) &= \log p_{\text{B}}(\mathbf{x}; \hat{\mathbf{x}}_\theta(\mathbf{z})) \\ &= \sum_{i=1}^{D} x_i \log \hat{x}_i + (1 - x_i)\log(1 - \hat{x}_i)\end{aligned}$$

where x_i is the pixel value of the i^{th} pixel in the input sample and $\hat{x}_i$ (or $\hat{x}_i(\mathbf{z})$) is the corresponding pixel value in the image reconstructed by the decoder, and $\mathbf{z}$ is the latent representation corresponding to the input image (see [4] their Appendix C.1).

The negative reconstruction error for perfect reconstruction is simply calculated by setting $\hat{x}_i = x_i$, as:

$$\log p_{\text{B}}(\mathbf{x}; \mathbf{x}) = \sum_{i=1}^{D} x_i \log x_i + (1 - x_i)\log(1 - x_i)$$

Bias correction for the continuous Bernoulli decoder. For a VAE decoder with a continuous Bernoulli visible distribution, the negative reconstruction error is given by:

$$\begin{aligned}\log p_\theta(\mathbf{x}|\mathbf{z}) &= \log p_{\text{cB}}(\mathbf{x}; \boldsymbol{\lambda}_\theta(\mathbf{z})) \\ &= \sum_{i=1}^{D} \log C(\lambda_i) + x_i \log \lambda_i + (1 - x_i)\log(1 - \lambda_i)\end{aligned} \tag{2}$$

Note that the continuous Bernoulli decoder outputs the shape parameter (λ_i) for the i^{th} pixel. The decoded pixel value itself is given by:

$$\begin{aligned} \hat{x}_i &= \frac{\lambda_i}{2\lambda_i - 1} + \frac{1}{2\tanh^{-1}(1-2\lambda_i)} && \text{if} \quad \lambda_i \neq \frac{1}{2} \\ &= \frac{1}{2} && \text{if} \quad \lambda_i = \frac{1}{2} \end{aligned}$$

As before, for perfect reconstruction we set $\hat{x}_i = x_i$. To find the optimal λ_i^* corresponding to perfect reconstruction, we used SciPy's implementation of Nelder-Mead simplex algorithm to iteratively maximize $\log p_{\text{CB}}(x_i; \lambda_i)$; the correction was then calculated by setting $\lambda_i = \lambda_i^*$ in equation (2) above.

Appendix C: Bias correction tested on 9 grayscale and natural image datasets

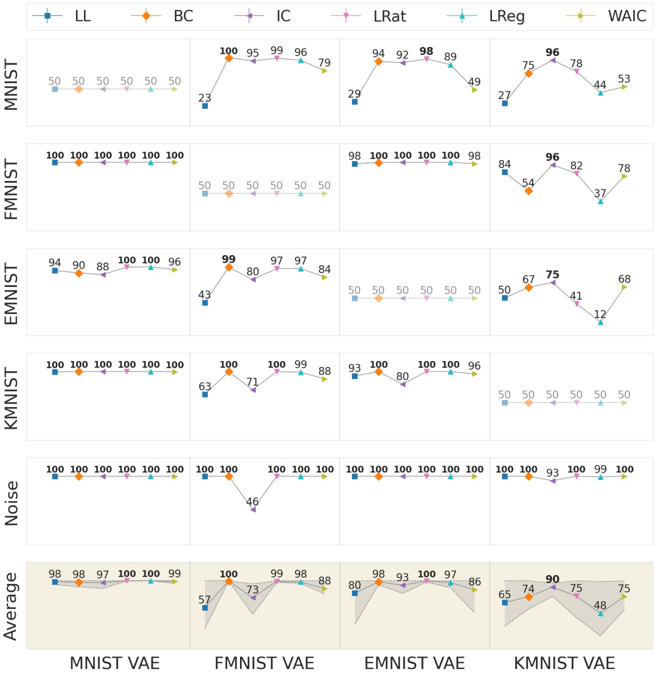


Figure 5: Outlier detection AUROC for 4 different grayscale image VAEs (columns), each tested with the other 3 grayscale image datasets and noise. The last row shows the average AUROC across all outlier datasets (also reported in Table 1).

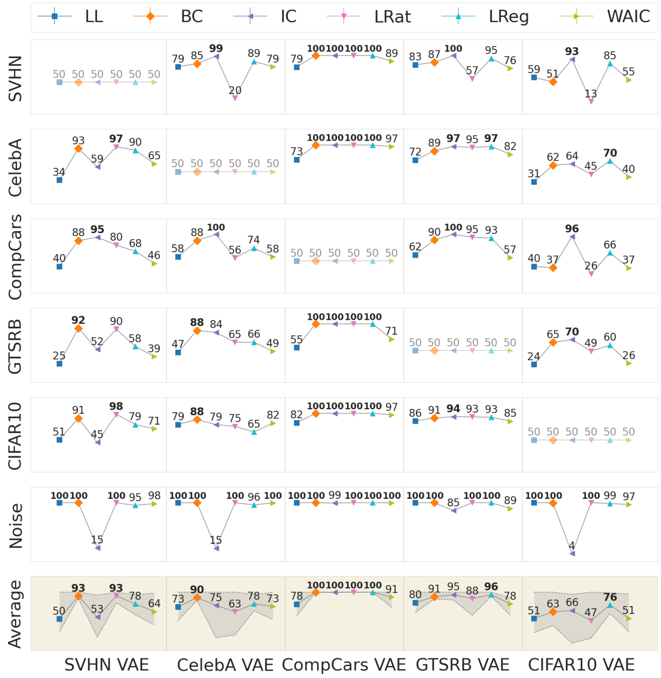


Figure 6: Same as Figure 5 but for natural image datasets.

Appendix D: Compute times and speedups

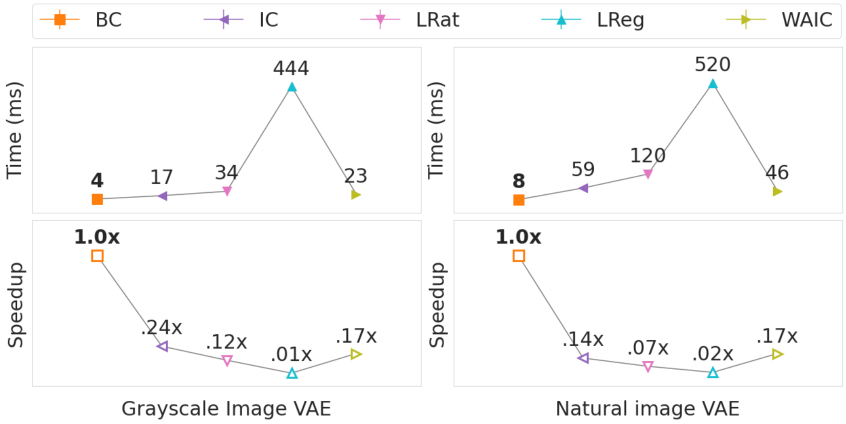


Figure 7: Compute times (top row) and speedups (bottom row) for BC using a grayscale (left) and natural (right) image VAEs. Compute times are averaged across 10,000 test examples.

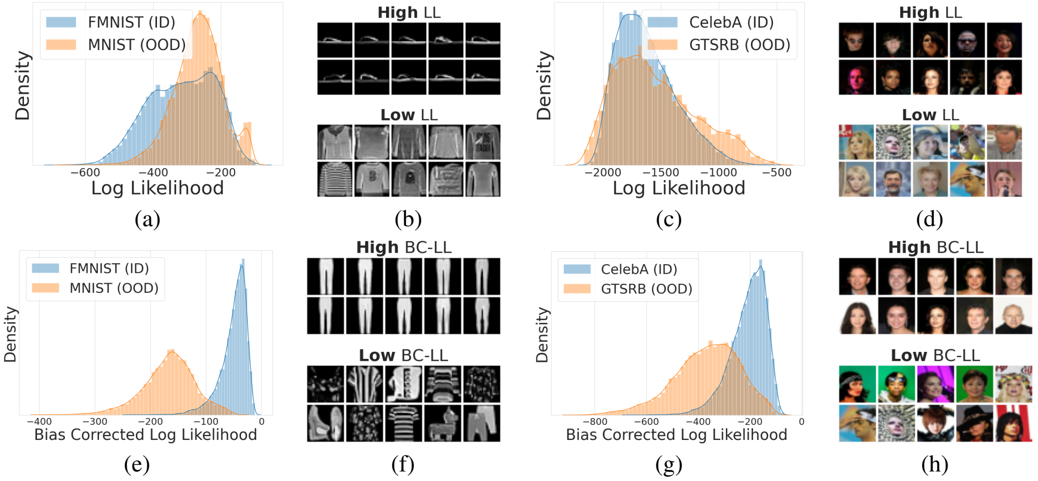
.

Appendix E: Bias correction with alternative visible distributions

We report outlier detection results for Bernoulli (Fig. 8), Categorial (Fig. 9) and truncated Gaussian (Fig. 10) VAEs. To correct biases in the Bernoulli VAE, we compute the correction factor as discussed in Appendix B. For Categorical and truncated Gaussian VAEs, we corrected for the bias with the following approach: For each VAE, we computed the average probability assigned for every target pixel value (0-255) employing all of the training (inlier) samples for that VAE. Then, for each test sample, we computed the correction term for the negative reconstruction error as its average value across pixels, under the Categorical/truncated Gaussian visible distribution. This procedure is described in detail in Algorithm 1. For all VAEs, the images were contrast stretched during both training and testing phases.

Algorithm 1: Bias correction for the Categorical/truncated Gaussian visible distribution

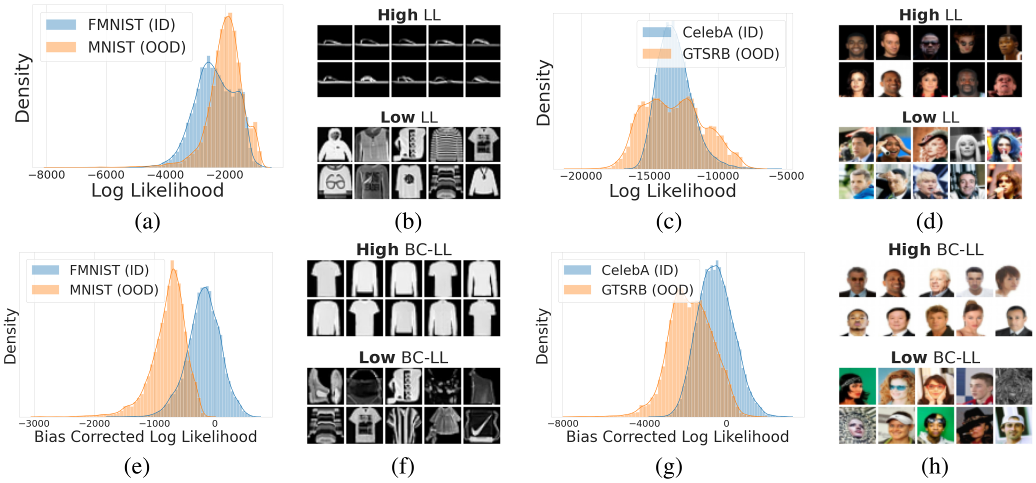
```
Data: Training Set $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, ...\mathbf{x_n}\}$ with $\mathbf{x_p}$ of shape $32 \times 32 \times nc$ (no. of channels),
      Encoder Parameters $\phi$, and Decoder Parameters $\theta$
Result: Log Correction Factor $\mathbf{C} : (v, k) \rightarrow Float$ for $v = 0, 1, \ldots 255$ and $k = 1, 2, \ldots nc$
Init: Map $\mathbf{A} : (v, k) \rightarrow EmptyList$ for $v = 0, 1, \ldots 255$ and $k = 1, 2, \ldots nc$
for $\mathbf{x_p} \in \mathbf{X}$ do
    Init: Map $\mathbf{B} : (v, k) \rightarrow EmptyList$ for $v = 0, 1, \ldots 255$ and $k = 1, 2, \ldots nc$
    $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x_p})$
    for $i \leftarrow 1$ to $32$, $j \leftarrow 1$ to $32$, $k \leftarrow 1$ to $nc$ do
        Append $p_\theta^{ijk}(x_p^{ijk}|\bar{\mathbf{z}})$ to $\mathbf{B}(x_p^{ijk}, k)$
    end
    for $v \leftarrow 0$ to $255$, $k \leftarrow 1$ to $nc$ do
        Append $\texttt{Mean}(\mathbf{B}(v, k))$ to $\mathbf{A}(v, k)$
    end
end
Init: Map $\mathbf{C} : (v, k) \rightarrow 0$ for $v = 0, 1, \ldots 255$ and $k = 1, 2, \ldots nc$
for $v \leftarrow 0$ to $255$, $k \leftarrow 1$ to $nc$ do
    $\mathbf{C}(v, k) \leftarrow \texttt{Log}(\texttt{Mean}(\mathbf{A}(v, k)))$
end
```



(a) (b) (c) (d)

(e) (f) (g) (h)

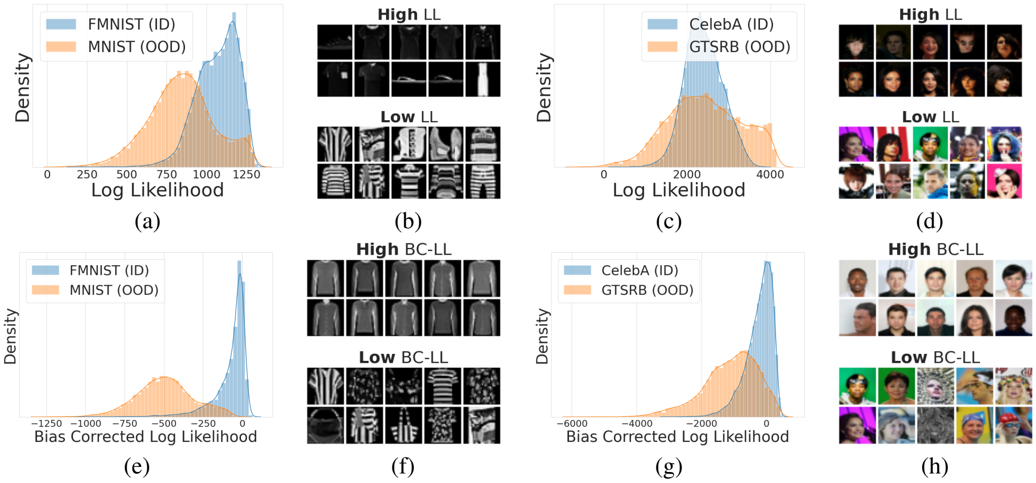
Figure 8: Bias correction improves outlier detection for the Bernoulli VAE



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 9: Bias correction improves outlier detection for the Categorical VAE



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 10: Bias correction improves outlier detection for the truncated Gaussian VAE