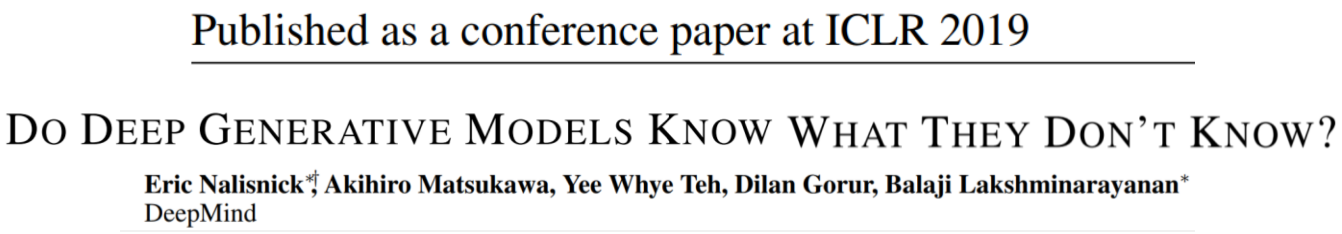
# Do Deep Generative Models Know What They Don’t Know ?



Published as a conference paper at ICLR 2019

DO DEEP GENERATIVE MODELS KNOW WHAT THEY DON’T KNOW?

Eric Nalisnick[*†], Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, Balaji Lakshminarayanan[*]
DeepMind

**内容提要：**

➢1、一般的神经网络用来做识别很容易被“Out of distribution(OoD)”攻击。（网络给出很高的置信度，但却是错误的）

➢2、生成网络通常被认为对于此类情况具有很强的鲁棒性，因为模型学习的是输入特征的概率密度。

➢3、这篇文章主要挑战了上述观点。本文在三个主流的生成模型flow，VAEs，PixelCNNs上发现，不能将门牌号和狗，卡车，马等图片区分开来。

➢4、论文基于flow模型在FashionMNIST vs MNIST, CelebA vs SVHN, ImageNet vs CIFAR-10 / CIFAR-100 / SVHN 等数据集对上进行了测试。

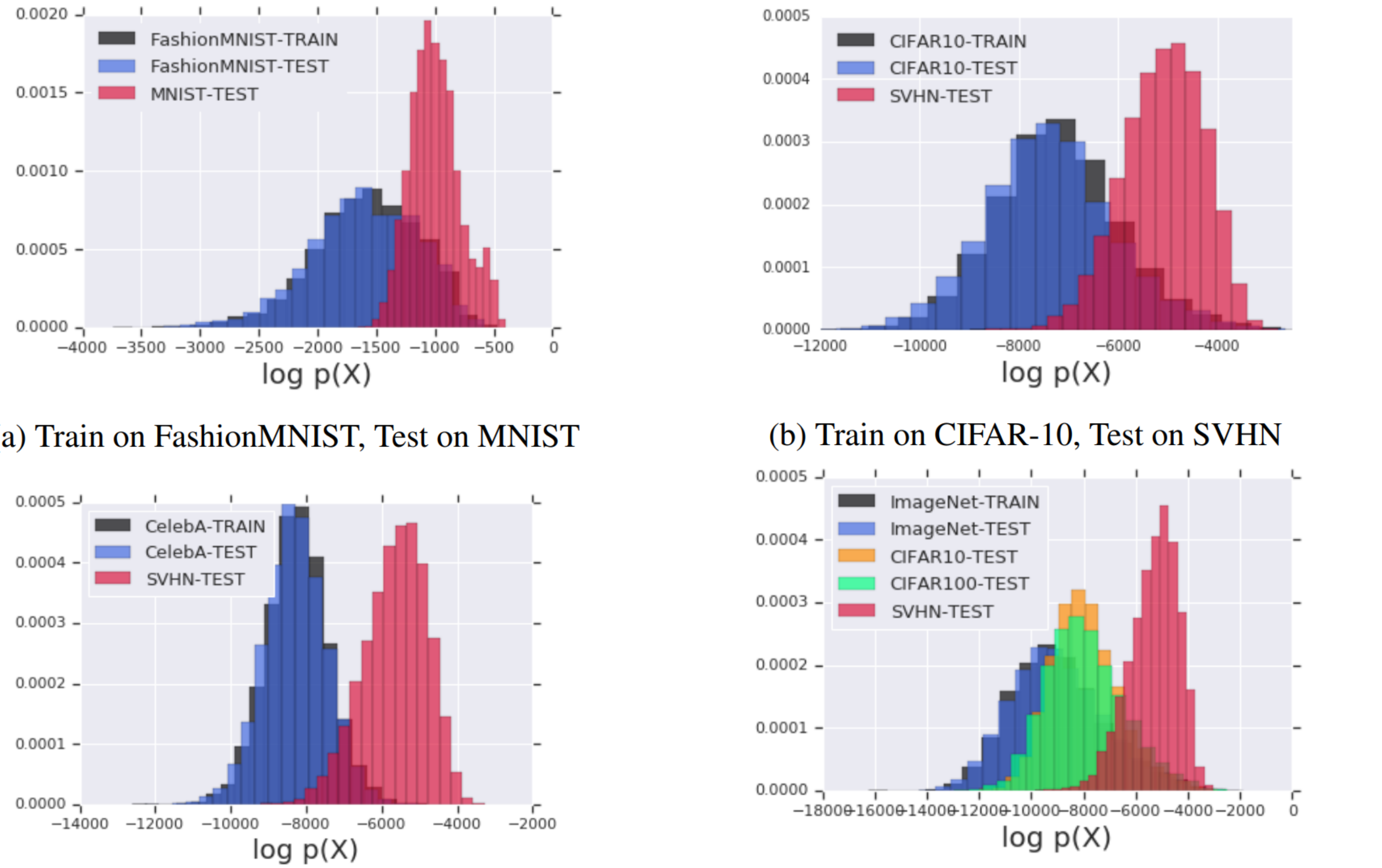
➢5、为了便于分析，论文采用了“constant-volume flow”

➢6、最后文章提醒，在“Ood”行为被更好的理解前应该慎重使用生成模型来做识别。

| Data Set | Avg. Bits Per Dimension |
|---|---|
| *Glow Trained on FashionMNIST* | |
| FashionMNIST-Train | 2.902 |
| FashionMNIST-Test | 2.958 |
| MNIST-Test | **1.833** |
| *Glow Trained on MNIST* | |
| MNIST-Test | 1.262 |

| Data Set | Avg. Bits Per Dimension |
|---|---|
| *Glow Trained on CIFAR-10* | |
| CIFAR10-Train | 3.386 |
| CIFAR10-Test | 3.464 |
| SVHN-Test | **2.389** |
| *Glow Trained on SVHN* | |
| SVHN-Test | 2.057 |

Figure 1: *Testing Out-of-Distribution.* Log-likelihood (expressed in bits per dimension) calculated from Glow (Kingma & Dhariwal, 2018) on MNIST, FashionMNIST, SVHN, CIFAR-10.



(a) Train on FashionMNIST, Test on MNIST

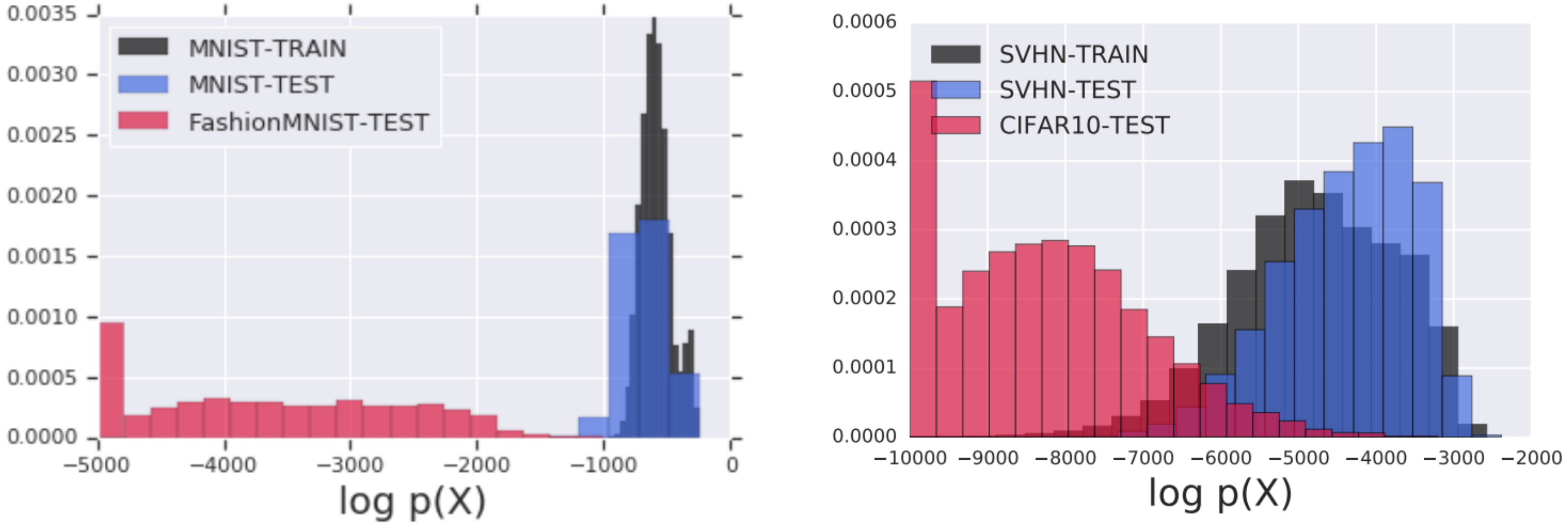
(b) Train on CIFAR-10, Test on SVHN

(c) Train on CelebA, Test on SVHN

(d) Train on ImageNet,
Test on CIFAR-10 / CIFAR-100 / SVHN

Figure 2: Histogram of Glow log-likelihoods for FashionMNIST vs MNIST (a), CIFAR-10 vs SVHN (b), CelebA vs SVHN (c), and ImageNet vs CIFAR-10 / CIFAR-100 / SVHN (d).

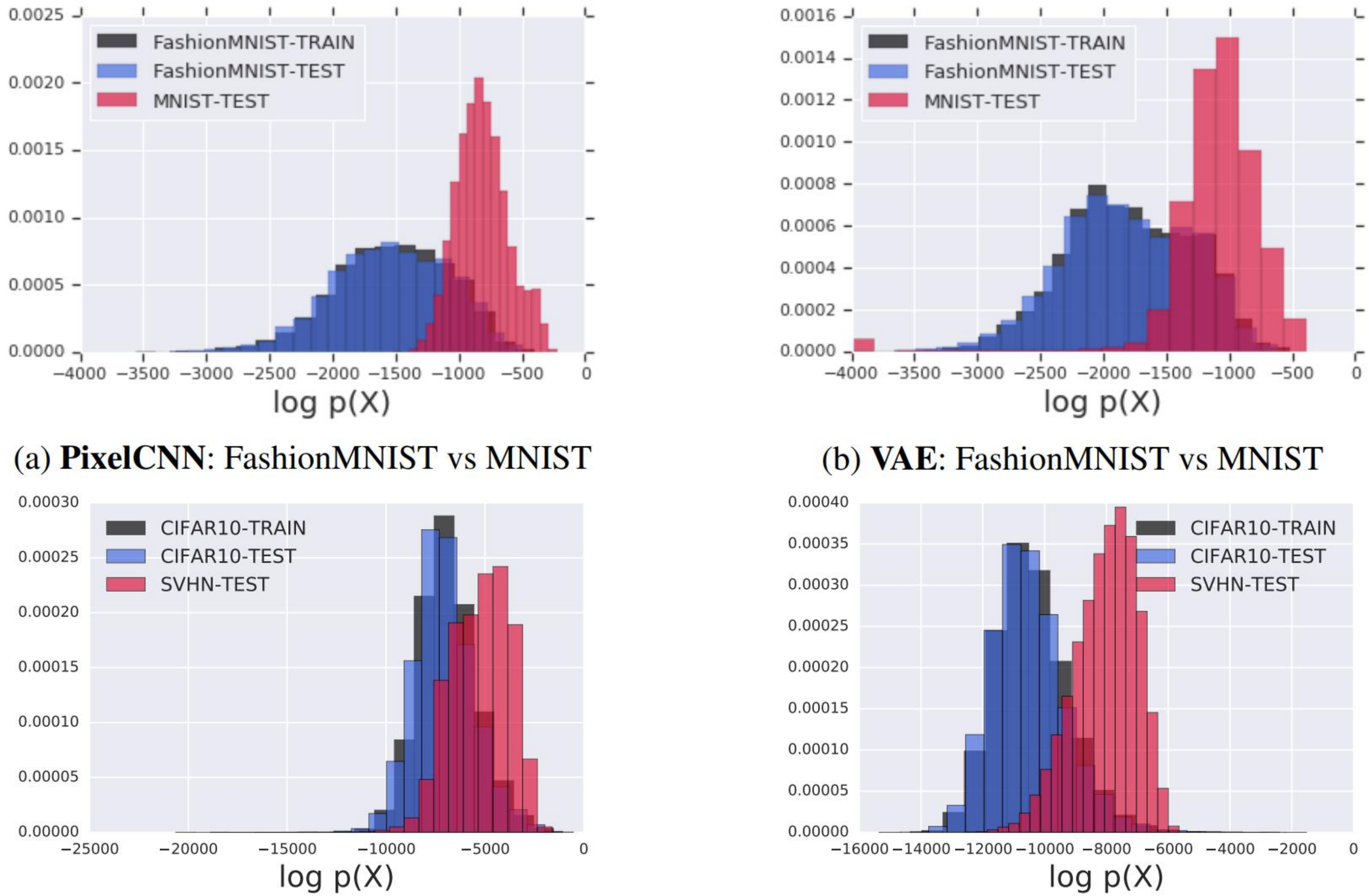
# B RESULTS ILLUSTRATING ASYMMETRIC BEHAVIOR



(a) Train on MNIST, Test on FashionMNIST

(b) Train on SVHN, Test on CIFAR-10

Figure 6: Histogram of Glow log-likelihoods for MNIST vs FashionMNIST and SVHN vs CIFAR-10. Note that the model trained on SVHN (MNIST) is able to assign lower likelihood to CIFAR-10 (FashionMNIST), which illustrates the asymmetry compared to Figure 2.



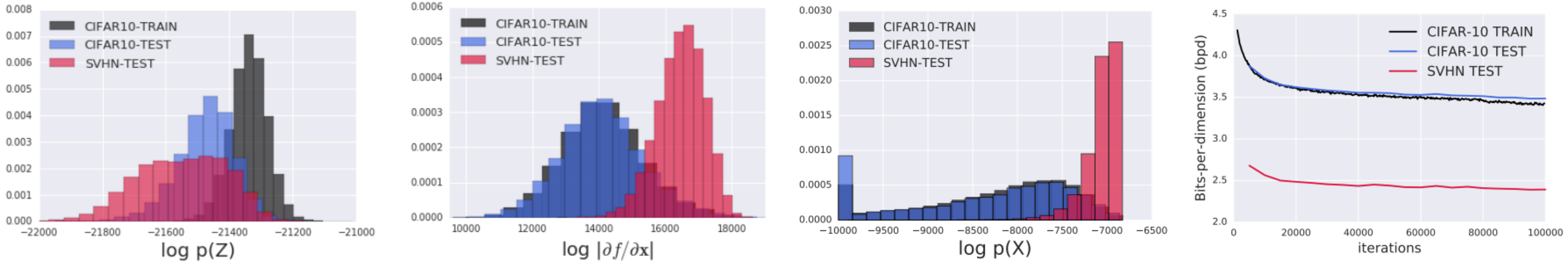
(a) **PixelCNN**: FashionMNIST vs MNIST

(b) **VAE**: FashionMNIST vs MNIST

(c) **PixelCNN**: CIFAR-10 vs SVHN

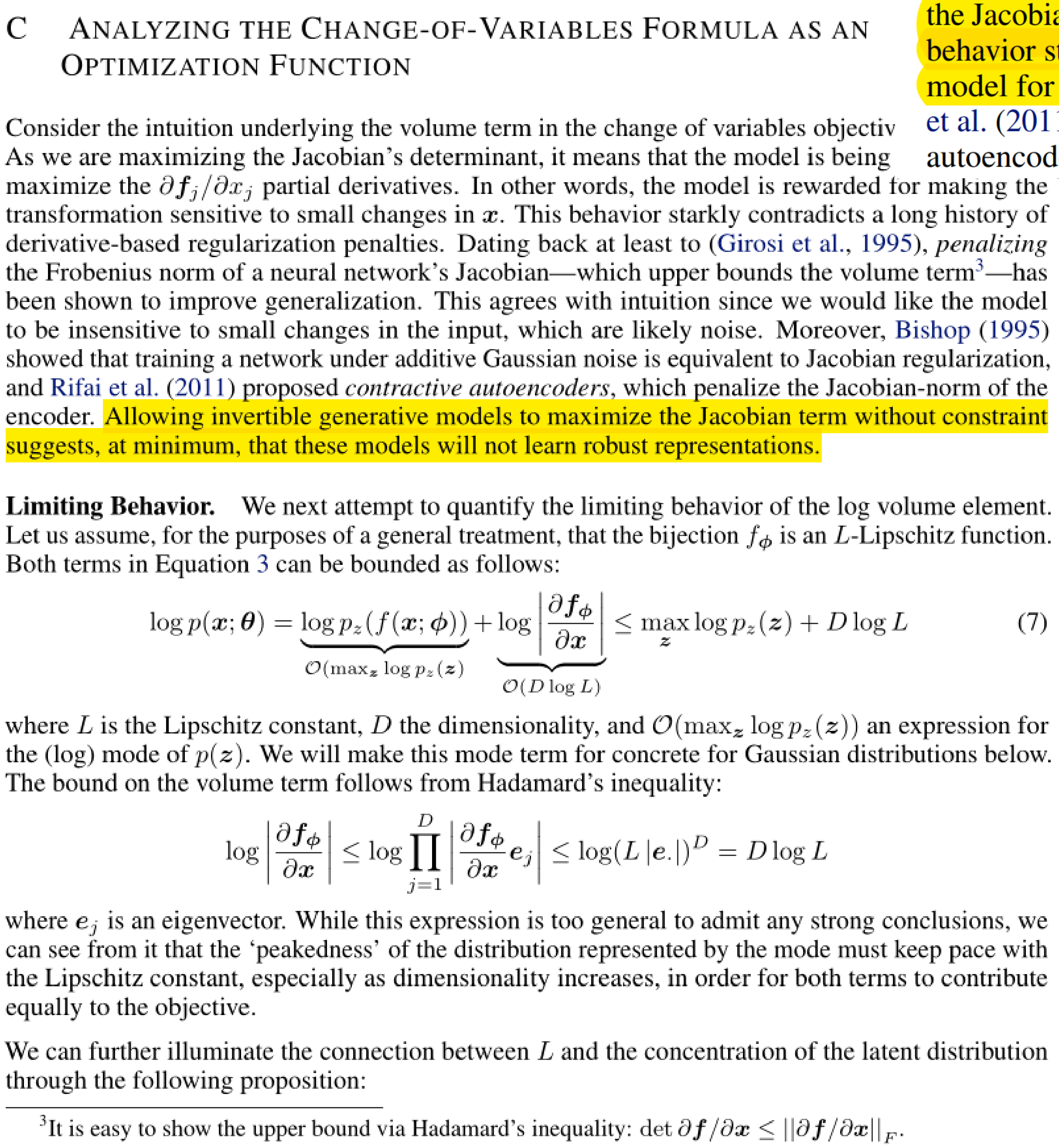
(d) **VAE**: CIFAR-10 vs SVHN

Figure 3: *PixelCNN and VAE*. Log-likelihoods calculated by PixelCNN (a, c) and VAE (b, d) on FashionMNIST vs MNIST (a, b) and CIFAR-10 vs SVHN (c, d). VAE models are the convolutional categorical variant described by Rosca et al. (2018).



(a) CIFAR-10: $\log p(\boldsymbol{z})$ (b) CIFAR-10: Volume (c) CV-Glow Likelihoods (d) Log-Likelihood vs Iter.

Figure 4: *Decomposing the Likelihood of NVP-Glow / CV-Glow Results.* The histograms in (a) and (b) show NVP-Glow's log-likelihood decomposed into contributions from the $\boldsymbol{z}$-distribution and volume element, respectively, for CIFAR-10 vs SVHN. Subfigure (c) shows log-likelihood evaluations for constant-volume (CV) Glow, again when trained on CIFAR-10 and tested on SVHN. Subfigure (d) reports NVP-Glow's BPD over the course of training, showing that the phenomenon happens throughout and could not be prevented by early stopping.



## C Analyzing the Change-of-Variables Formula as an Optimization Function

Consider the intuition underlying the volume term in the change of variables objectiv
As we are maximizing the Jacobian's determinant, it means that the model is being
maximize the $\partial \boldsymbol{f}_j / \partial x_j$ partial derivatives. In other words, the model is rewarded for making the transformation sensitive to small changes in $\boldsymbol{x}$. This behavior starkly contradicts a long history of derivative-based regularization penalties. Dating back at least to (Girosi et al., 1995), *penalizing* the Frobenius norm of a neural network's Jacobian—which upper bounds the volume term[3]—has been shown to improve generalization. This agrees with intuition since we would like the model to be insensitive to small changes in the input, which are likely noise. Moreover, Bishop (1995) showed that training a network under additive Gaussian noise is equivalent to Jacobian regularization, and Rifai et al. (2011) proposed *contractive autoencoders*, which penalize the Jacobian-norm of the encoder. Allowing invertible generative models to maximize the Jacobian term without constraint suggests, at minimum, that these models will not learn robust representations.

**Limiting Behavior.** We next attempt to quantify the limiting behavior of the log volume element. Let us assume, for the purposes of a general treatment, that the bijection $f_\phi$ is an $L$-Lipschitz function. Both terms in Equation 3 can be bounded as follows:

$$\log p(\boldsymbol{x};\boldsymbol{\theta}) = \underbrace{\log p_z(f(\boldsymbol{x};\boldsymbol{\phi}))}_{\mathcal{O}(\max_{\boldsymbol{z}} \log p_z(\boldsymbol{z}))} + \underbrace{\log \left| \frac{\partial \boldsymbol{f}_\phi}{\partial \boldsymbol{x}} \right|}_{\mathcal{O}(D \log L)} \leq \max_{\boldsymbol{z}} \log p_z(\boldsymbol{z}) + D \log L \tag{7}$$

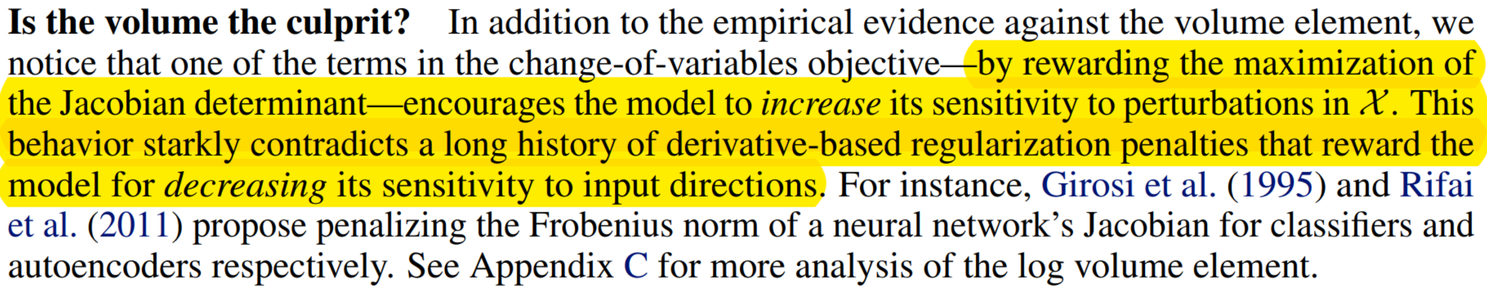
where $L$ is the Lipschitz constant, $D$ the dimensionality, and $\mathcal{O}(\max_{\boldsymbol{z}} \log p_z(\boldsymbol{z}))$ an expression for the (log) mode of $p(\boldsymbol{z})$. We will make this mode term for concrete for Gaussian distributions below. The bound on the volume term follows from Hadamard's inequality:

$$\log \left| \frac{\partial \boldsymbol{f}_\phi}{\partial \boldsymbol{x}} \right| \leq \log \prod_{j=1}^{D} \left| \frac{\partial \boldsymbol{f}_\phi}{\partial \boldsymbol{x}} \boldsymbol{e}_j \right| \leq \log(L\,|\boldsymbol{e}_\cdot|)^D = D \log L$$

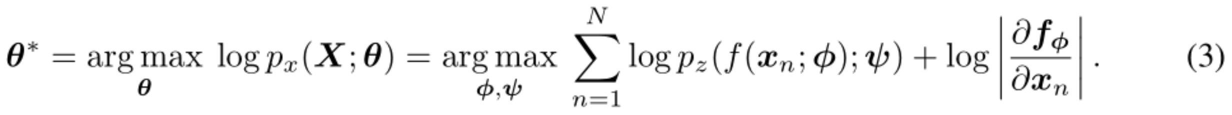
where $\boldsymbol{e}_j$ is an eigenvector. While this expression is too general to admit any strong conclusions, we can see from it that the 'peakedness' of the distribution represented by the mode must keep pace with the Lipschitz constant, especially as dimensionality increases, in order for both terms to contribute equally to the objective.

We can further illuminate the connection between $L$ and the concentration of the latent distribution through the following proposition:

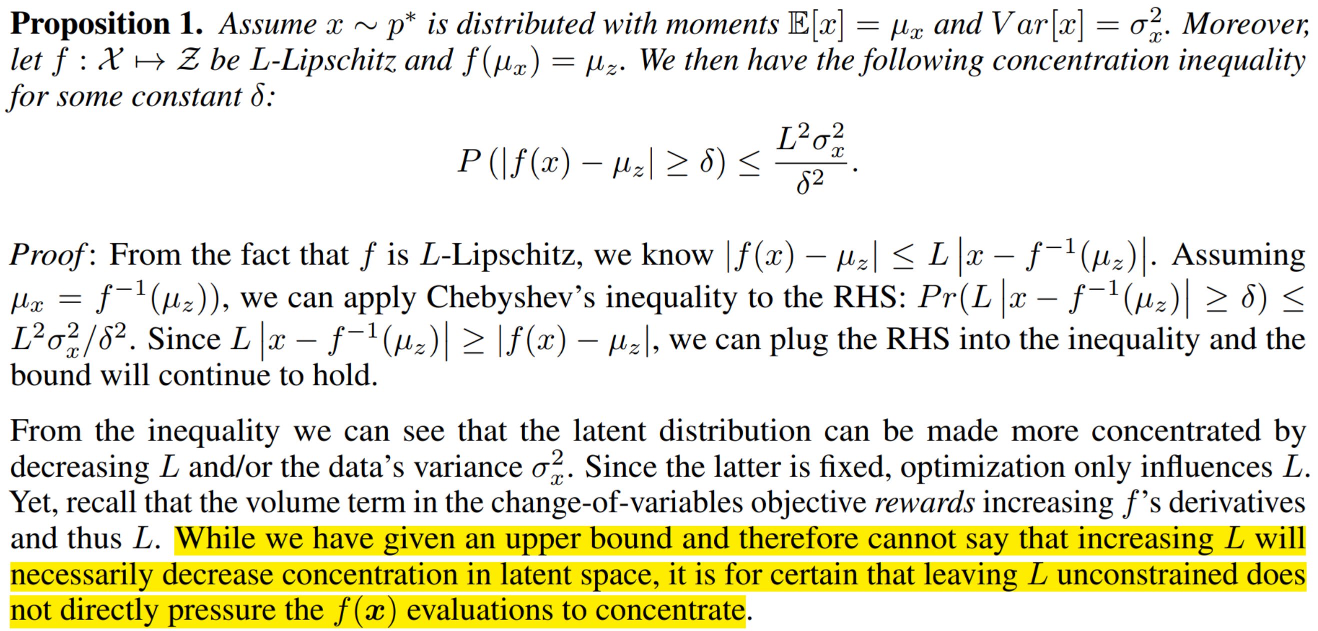
[3] It is easy to show the upper bound via Hadamard's inequality: $\det \partial \boldsymbol{f}/\partial \boldsymbol{x} \leq ||\partial \boldsymbol{f}/\partial \boldsymbol{x}||_F$.



**Is the volume the culprit?** In addition to the empirical evidence against the volume element, we notice that one of the terms in the change-of-variables objective—by rewarding the maximization of the Jacobian determinant—encourages the model to *increase* its sensitivity to perturbations in $\mathcal{X}$. This behavior starkly contradicts a long history of derivative-based regularization penalties that reward the model for *decreasing* its sensitivity to input directions. For instance, Girosi et al. (1995) and Rifai et al. (2011) propose penalizing the Frobenius norm of a neural network's Jacobian for classifiers and autoencoders respectively. See Appendix C for more analysis of the log volume element.



$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \log p_x(\boldsymbol{X};\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\phi},\boldsymbol{\psi}} \sum_{n=1}^{N} \log p_z(f(\boldsymbol{x}_n;\boldsymbol{\phi});\boldsymbol{\psi}) + \log \left| \frac{\partial \boldsymbol{f}_\phi}{\partial \boldsymbol{x}_n} \right|. \tag{3}$$

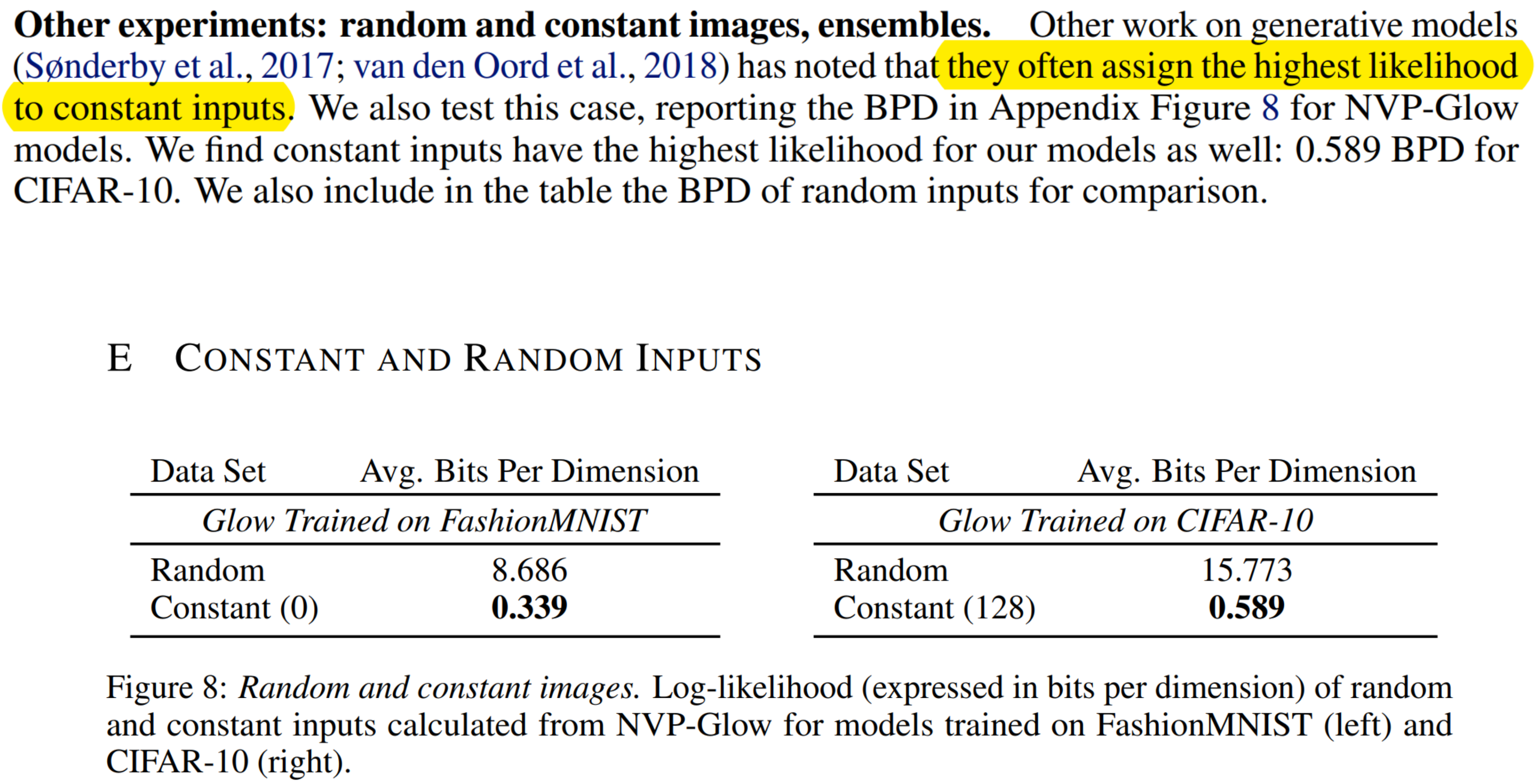


**Proposition 1.** *Assume $x \sim p^*$ is distributed with moments $\mathbb{E}[x] = \mu_x$ and $Var[x] = \sigma_x^2$. Moreover, let $f : \mathcal{X} \mapsto \mathcal{Z}$ be $L$-Lipschitz and $f(\mu_x) = \mu_z$. We then have the following concentration inequality for some constant $\delta$:*

$$P\left(|f(x) - \mu_z| \geq \delta\right) \leq \frac{L^2 \sigma_x^2}{\delta^2}.$$

*Proof*: From the fact that $f$ is $L$-Lipschitz, we know $|f(x) - \mu_z| \leq L\left|x - f^{-1}(\mu_z)\right|$. Assuming $\mu_x = f^{-1}(\mu_z))$, we can apply Chebyshev's inequality to the RHS: $Pr(L\left|x - f^{-1}(\mu_z)\right| \geq \delta) \leq L^2\sigma_x^2/\delta^2$. Since $L\left|x - f^{-1}(\mu_z)\right| \geq |f(x) - \mu_z|$, we can plug the RHS into the inequality and the bound will continue to hold.

From the inequality we can see that the latent distribution can be made more concentrated by decreasing $L$ and/or the data's variance $\sigma_x^2$. Since the latter is fixed, optimization only influences $L$. Yet, recall that the volume term in the change-of-variables objective *rewards* increasing $f$'s derivatives and thus $L$. While we have given an upper bound and therefore cannot say that increasing $L$ will necessarily decrease concentration in latent space, it is for certain that leaving $L$ unconstrained does not directly pressure the $f(\boldsymbol{x})$ evaluations to concentrate.



**Other experiments: random and constant images, ensembles.** Other work on generative models (Sønderby et al., 2017; van den Oord et al., 2018) has noted that they often assign the highest likelihood to constant inputs. We also test this case, reporting the BPD in Appendix Figure 8 for NVP-Glow models. We find constant inputs have the highest likelihood for our models as well: 0.589 BPD for CIFAR-10. We also include in the table the BPD of random inputs for comparison.

## E Constant and Random Inputs

| Data Set | Avg. Bits Per Dimension |
|---|---|
| *Glow Trained on FashionMNIST* | |
| Random | 8.686 |
| Constant (0) | **0.339** |

| Data Set | Avg. Bits Per Dimension |
|---|---|
| *Glow Trained on CIFAR-10* | |
| Random | 15.773 |
| Constant (128) | **0.589** |

Figure 8: *Random and constant images.* Log-likelihood (expressed in bits per dimension) of random and constant inputs calculated from NVP-Glow for models trained on FashionMNIST (left) and CIFAR-10 (right).

# When another distribution might have higher likelihood than the one used for training?

$$\mathbb{E}_q[\log p(\boldsymbol{x};\boldsymbol{\theta})] - \mathbb{E}_{p^*}[\log p(\boldsymbol{x};\boldsymbol{\theta})] > 0$$

$$\log p(\boldsymbol{x};\boldsymbol{\theta}) \approx \log p(\boldsymbol{x}_0;\boldsymbol{\theta}) + \nabla_{\boldsymbol{x}_0} \log p(\boldsymbol{x}_0;\boldsymbol{\theta})^T(\boldsymbol{x}-\boldsymbol{x}_0) + \frac{1}{2}\operatorname{Tr}\{\nabla^2_{\boldsymbol{x}_0} \log p(\boldsymbol{x}_0;\boldsymbol{\theta})(\boldsymbol{x}-\boldsymbol{x}_0)(\boldsymbol{x}-\boldsymbol{x}_0)^T\}$$

$$\begin{aligned} 0 &< \mathbb{E}_q[\log p(\boldsymbol{x};\boldsymbol{\theta})] - \mathbb{E}_{p^*}[\log p(\boldsymbol{x};\boldsymbol{\theta})] \\ &\approx \nabla_{\boldsymbol{x}_0} \log p(x_0;\boldsymbol{\theta})^T(\mathbb{E}_q[\boldsymbol{x}] - \mathbb{E}_{p^*}[\boldsymbol{x}]) + \frac{1}{2}\operatorname{Tr}\{\nabla^2_{\boldsymbol{x}_0} \log p(\boldsymbol{x}_0;\boldsymbol{\theta})(\boldsymbol{\Sigma}_q - \boldsymbol{\Sigma}_{p^*})\} \end{aligned} \quad (4)$$

where $\boldsymbol{\Sigma} = \mathbb{E}\left[(\boldsymbol{x}-\boldsymbol{x}_0)(\boldsymbol{x}-\boldsymbol{x}_0)^T\right]$, we next assume that $\mathbb{E}_q[\boldsymbol{x}] = \mathbb{E}_{p^*}[\boldsymbol{x}] = \boldsymbol{x}_0$.

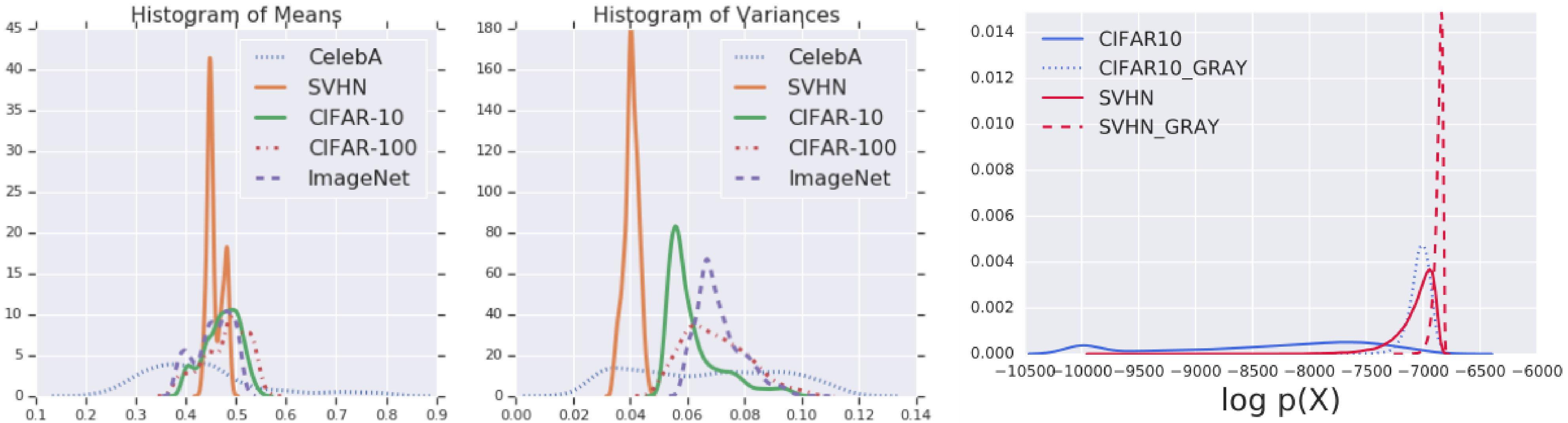
$$\begin{aligned} 0 < \mathbb{E}_q[\log p(\boldsymbol{x};\boldsymbol{\theta})] - \mathbb{E}_{p^*}[\log p(\boldsymbol{x};\boldsymbol{\theta})] &\approx \frac{1}{2}\operatorname{Tr}\{\nabla^2_{\boldsymbol{x}_0} \log p(\boldsymbol{x}_0;\boldsymbol{\theta})(\boldsymbol{\Sigma}_q - \boldsymbol{\Sigma}_{p^*})\} \\ = \frac{1}{2}\operatorname{Tr}\left\{\left[\nabla^2_{\boldsymbol{x}_0} \log p_z(f(\boldsymbol{x}_0;\boldsymbol{\phi})) + \nabla^2_{\boldsymbol{x}_0} \log\left|\frac{\partial \boldsymbol{f}_{\boldsymbol{\phi}}}{\partial \boldsymbol{x}_0}\right|\right](\boldsymbol{\Sigma}_q - \boldsymbol{\Sigma}_{p^*})\right\}, \end{aligned} \quad (5)$$

and data's second moment. The second derivative terms simplify considerably for CV-Glow with a spherical latent density. Given a $C \times C$ kernel $\boldsymbol{U}_k$, with $k$ indexing the flow and $C$ the number of input channels, the derivatives are $\partial f_{h,w,c} / \partial x_{h,w,c} = \prod_k \sum_{j=1}^{C} u_{k,c,j}$, with $h$ and $w$ indexing the spatial height and width and $j$ the columns of the $k$th flow's $1 \times 1$ convolutional kernel. The second derivative is then $\partial^2 f_{h,w,c} / \partial x_{h,w,c}^2 = 0$, which allows us to write

$$\begin{aligned} &\operatorname{Tr}\left\{\left[\nabla^2_{\boldsymbol{x}_0} \log p(\boldsymbol{x}_0; \boldsymbol{\theta})\right] (\boldsymbol{\Sigma}_q - \boldsymbol{\Sigma}_{p^*})\right\} \\ &\quad = \frac{\partial^2}{\partial z^2} \log p(\boldsymbol{z}; \boldsymbol{\psi}) \sum_{c=1}^{C} \left( \prod_{k=1}^{K} \sum_{j=1}^{C} u_{k,c,j} \right)^2 \sum_{h,w} (\sigma^2_{q,h,w,c} - \sigma^2_{p^*,h,w,c}). \end{aligned}$$

The derivation is given in Appendix G. Plugging in the second derivative of the Gaussian's log density—a common choice for the latent distribution in flow models (Dinh et al., 2017; Kingma & Dhariwal, 2018)—and the empirical variances, we have:

$$\begin{aligned} &\mathbb{E}_{\text{SVHN}}[\log p(\boldsymbol{x}; \boldsymbol{\theta})] - \mathbb{E}_{\text{CIFAR-10}}[\log p(\boldsymbol{x}; \boldsymbol{\theta})] \\ &\quad \approx \frac{-1}{2\sigma^2_{\boldsymbol{\psi}}} \left[\alpha_1^2 (49.6 - 61.9) + \alpha_2^2 (52.7 - 59.2) + \alpha_3^2 (53.6 - 68.1)\right] \\ &\quad = \frac{1}{2\sigma^2_{\boldsymbol{\psi}}} \left[\alpha_1^2 \cdot 12.3 + \alpha_2^2 \cdot 6.5 + \alpha_3^2 \cdot 14.5\right] \geq 0 \quad \text{where } \alpha_c = \prod_{k=1}^{K} \sum_{j=1}^{C} u_{k,c,j} \end{aligned} \tag{6}$$



(a) Histogram of per-dimension means and variances (empirical).

(b) Graying images increases likelihood.

Figure 5: *Empirical Distributions and Graying Effect.* Note that pixels are converted from 0-255 scale to 0-1 scale by diving by 256. See Figure 10 for results on datasets of 28 × 28 × 1 images.

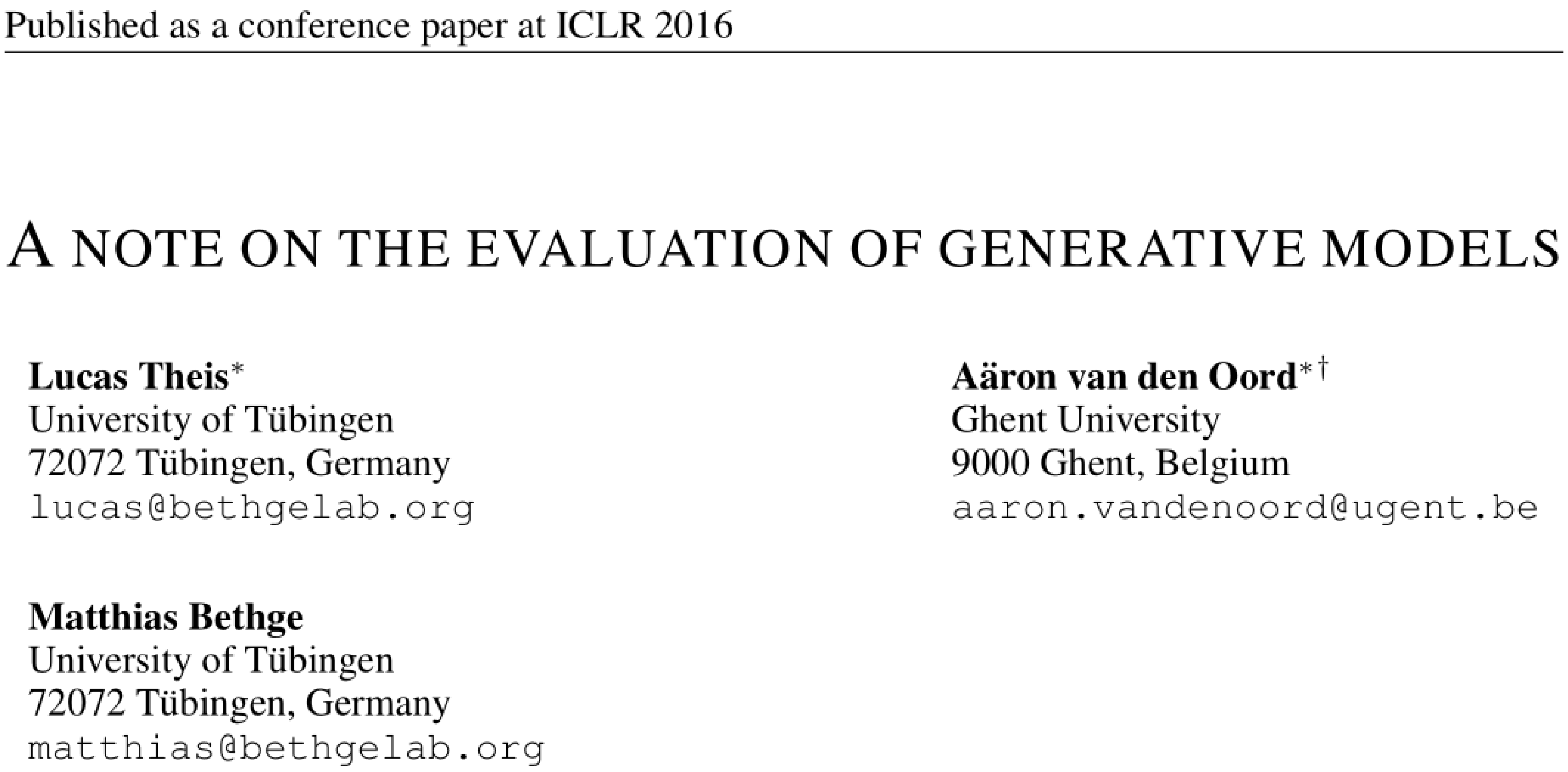
Our conclusion is that SVHN simply "sits inside of" CIFAR-10—roughly same mean, smaller variance—resulting in its higher likelihood. This insight also holds true for the additional results

This paper is inspired by and most related to recent work on evaluation of generative models. Worthy of foremost mention is the work of Theis et al. (2016), which showed that high likelihood is neither sufficient nor necessary for the model to produce visually satisfying samples. However, their paper does not consider out-of-distribution inputs. In this regard, there has been much work on *adversarial inputs* (Szegedy et al., 2014). While the term is used broadly, it commonly refers to inputs that have been imperceptibly modified so that the model can no longer provide an accurate output (a mis-classification, usually). Adversarial attacks on generative models have been studied by (at least) Tabacof et al. (2016) and Kos et al. (2018), but these methods of attack require access to the model. We, on the other hand, are interested in model calibration for any out-of-distribution set and especially for common data sets not constructed with any nefarious intentions nor for attack on a particular model. Various papers (Hendrycks & Gimpel, 2017; Lakshminarayanan et al., 2017; Liang et al., 2018) have reported that discriminative neural networks can produce overconfident predictions on out-of-distribution inputs. In a related finding, Lee et al. (2018) reported that it was much harder to recognize an input as out-of-distribution when the classifier was trained on CIFAR-10 in comparison to training on SVHN.

Testing the robustness of deep generative models to out-of-distribution inputs had not been investigated previously, to the best of our knowledge. However, there is work concurrent with ours that has tested their ability to detect anomalous inputs. Shafaei et al. (2018) and Hendrycks et al. (2019) also observe that PixelCNN++ cannot provide reliable outlier detection. Hendrycks et al. (2019) mitigate the CIFAR-10 vs SVHN issue by exposing the model to outliers during training. They do not consider flow-based models. Škvára et al. (2018) experimentally compare VAEs and GANs against k-nearest neighbors (kNNs), showing that VAEs and GANs outperform kNNs only when known outliers can be used for hyperparameter selection. In the work most similar to ours, Choi & Jang (2018) report the same CIFAR-10 vs SVHN phenomenon for Glow—independently confirming our

## 7 DISCUSSION

We have shown that comparing the likelihoods of deep generative models alone cannot identify the training set or inputs like it. Therefore we urge caution when using these models with out-of-training-distribution inputs or in unprotected user-facing systems. Moreover, our analysis in Section 5 shows that the CIFAR-10 vs SVHN phenomenon would persist for any constant-volume Glow no matter the parameter values nor the choice of latent density (as long as it is log-concave). While we cannot conclude that this is a pathology in deep generative models, it does suggest the need for further work on generative models and their evaluation. The models we tested seem to be capturing low-level statistics rather than high-level semantics, and better inductive biases, optimization procedures, or uncertainty quantification may be necessary. Yet, deep generative models can detect out-of-distribution inputs when using alternative metrics (Choi & Jang, 2018) and modified training procedures (Hendrycks et al., 2019). The problem then may be a fundamental limitation of high-dimensional likelihoods. Until these open problems are better understood, we must temper the enthusiasm with which we preach the benefits of deep generative models.



Published as a conference paper at ICLR 2016

# A NOTE ON THE EVALUATION OF GENERATIVE MODELS

**Lucas Theis**[*]
University of Tübingen
72072 Tübingen, Germany
lucas@bethgelab.org

**Aäron van den Oord**[*][†]
Ghent University
9000 Ghent, Belgium
aaron.vandenoord@ugent.be

**Matthias Bethge**
University of Tübingen
72072 Tübingen, Germany
matthias@bethgelab.org

# A NOTE ON THE EVALUATION OF GENERATIVE MODELS

**内容提要：**

- 1、概率生成模型通常被用于：压缩、去噪、图像补全、文本合成、半监督学习、非监督特征学习等任务
- 2、不同的应用往往需要不同的训练和评估方法：文章主要对比了三种评价方法， average log-likelihood, Parzen window estimates, and visual fidelity of samples
- 3、结果表明当数据维度较高的时候这些评价方法往往是高度相互独立的，也就是说在一个标准上表现良好，并不一定意味着其在其它标准上也表现良好。

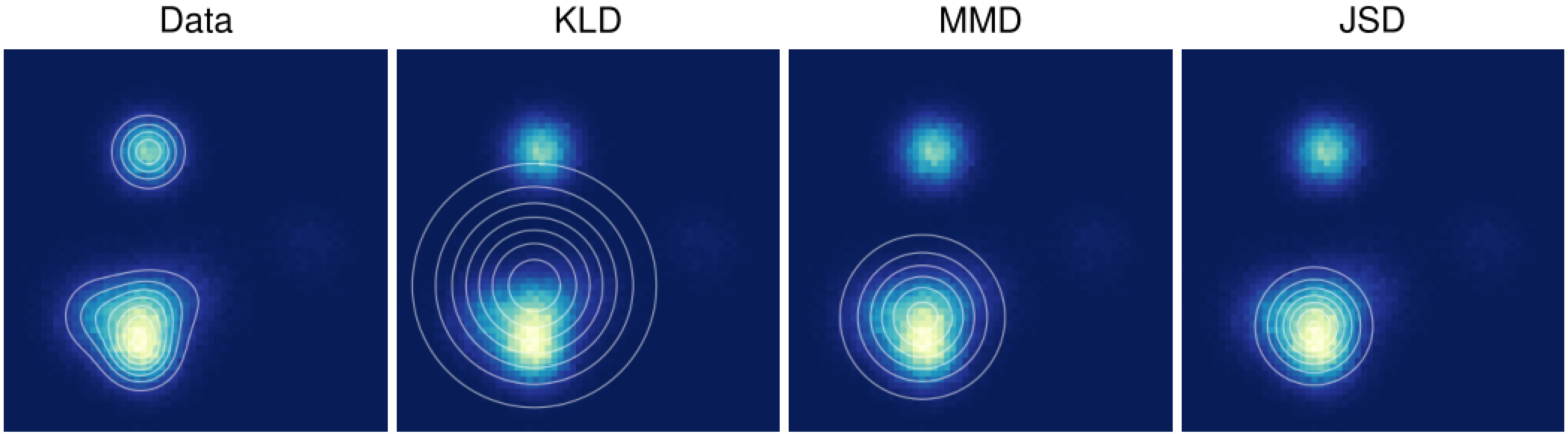


Figure 1: An isotropic Gaussian distribution was fit to data drawn from a mixture of Gaussians by either minimizing Kullback-Leibler divergence (KLD), maximum mean discrepancy (MMD), or Jensen-Shannon divergence (JSD). The different fits demonstrate different tradeoffs made by the three measures of distance between distributions.

**MMD：** has been used with generative moment matching networks
**JSD：** has connections to the objective function optimized by GAN

Minimizing MMD or JSD yields a Gaussian **which fits one mode well, but which ignores other parts of the data**. On the other hand, **maximizing average log-likelihood** or equivalently minimizing Kullback-Leibler divergence (**KLD**) **avoids assigning extremely small probability to any data point but assigns a lot of probability mass to non-data regions**.

### GREAT LOG-LIKELIHOOD AND POOR SAMPLES

Perhaps surprisingly, the ability to produce plausible samples is not only not sufficient, but also *not necessary* for high likelihood as a simple argument by van den Oord & Dambre (2015) shows: Assume $p$ is the density of a model for $d$ dimensional data $\mathbf{x}$ which performs arbitrarily well with respect to average log-likelihood and $q$ corresponds to some bad model (e.g., white noise). Then samples generated by the mixture model

$$0.01p(\mathbf{x}) + 0.99q(\mathbf{x}) \tag{11}$$

will come from the poor model 99% of the time. Yet the log-likelihood per pixel will hardly change if $d$ is large:

$$\log\left[0.01p(\mathbf{x}) + 0.99q(\mathbf{x})\right] \geq \log\left[0.01p(\mathbf{x})\right] = \log p(\mathbf{x}) - \log 100 \tag{12}$$

For high-dimensional data, $\log p(\mathbf{x})$ will be proportional to $d$ while $\log 100$ stays constant. For instance, already for the 32 by 32 images found in the CIFAR-10 dataset the difference between log-likelihoods of different models can be in the thousands, while $\log(100)$ is only about 4.61 nats (van den Oord & Dambre, 2015). This shows that a model can have large average log-likelihood but generate very poor samples.

### GOOD LOG-LIKELIHOOD AND GREAT SAMPLES

### 3.3 Samples and applications

One might conclude that something must be wrong with log-likelihood if it does not care about a model's ability to generate plausible samples. However, note that the mixture model in Equation 11 might also still work very well in applications. While $q$ is much more likely a priori, $p$ is going to be much more likely a posteriori in tasks like inpainting, denoising, or classification. Consider prediction of a quantity $y$ representing, for example, a class label or missing pixels. A model with joint distribution

$$0.01p(\mathbf{x})p(y \mid \mathbf{x}) + 0.99q(\mathbf{x})q(y \mid \mathbf{x}) \tag{13}$$

may again generate poor samples 99% of the time. For a given fixed $\mathbf{x}$, the posterior over $y$ will be a mixture

$$\alpha p(y \mid \mathbf{x}) + (1 - \alpha)q(y \mid \mathbf{x}), \tag{14}$$

where a few simple calculations show that

$$\alpha = \sigma\left(\ln p(\mathbf{x}) - \ln q(\mathbf{x}) - \ln 99\right) \tag{15}$$

and $\sigma$ is the sigmoidal logistic function. Since we assume that $p$ is a good model, $q$ is a poor model, and $\mathbf{x}$ is high-dimensional, we have

$$\ln p(\mathbf{x}) \gg \ln q(\mathbf{x}) + \ln 99 \tag{16}$$

and therefore $\alpha \approx 1$. That is, mixing with $q$ has hardly changed the posterior over $y$. While the samples are dominated by $q$, the classification performance is dominated by $p$. This shows that high visual fidelity of samples is generally not necessary for achieving good performance in applications.

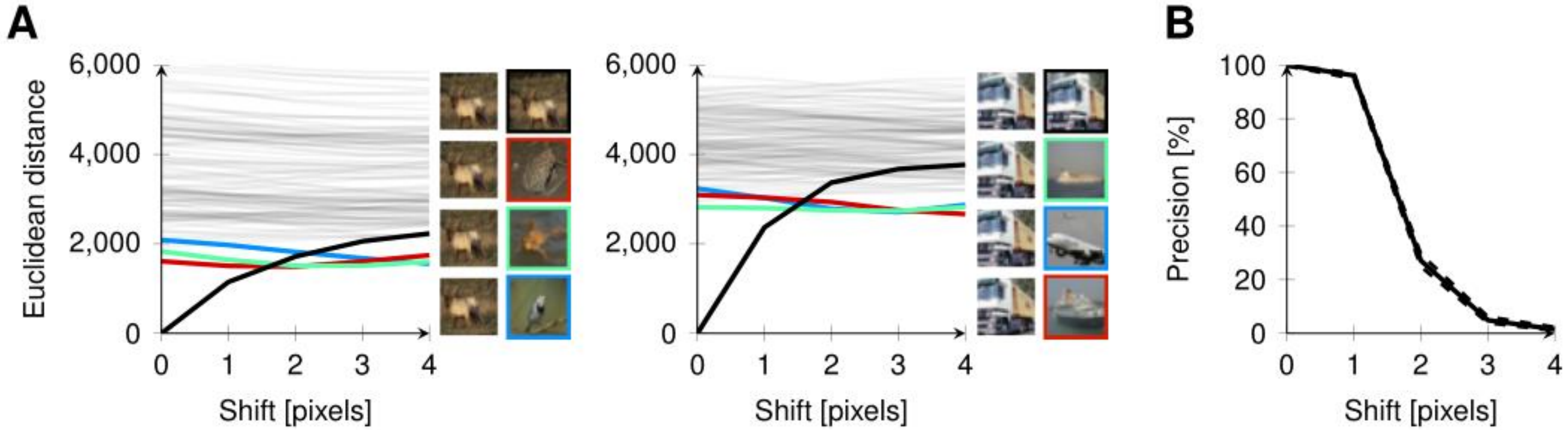


Figure 2: **A:** Two examples demonstrating that small changes of an image can lead to large changes in Euclidean distance affecting the choice of nearest neighbor. The images shown represent the query image shifted by between 1 and 4 pixels (left column, top to bottom), and the corresponding nearest neighbor from the training set (right column). The gray lines indicate Euclidean distance of the query image to 100 randomly picked images from the training set. **B:** Fraction of query images assigned to the correct training image. The average was estimated from 1,000 images. Dashed lines indicate a 90% confidence interval.

总结：

- 1、不同的模型需要不同的评估手段。在一个应用下表现良好，不一定在其它应用上也有好的表现。
- 2、一个基于“samples”的模型更倾向于过拟合，模型更倾向于大熵？
- 3、高likelihood并不能保证样本视觉上的真实。

Published as a conference paper at ICLR 2019

# DEEP ANOMALY DETECTION WITH OUTLIER EXPOSURE

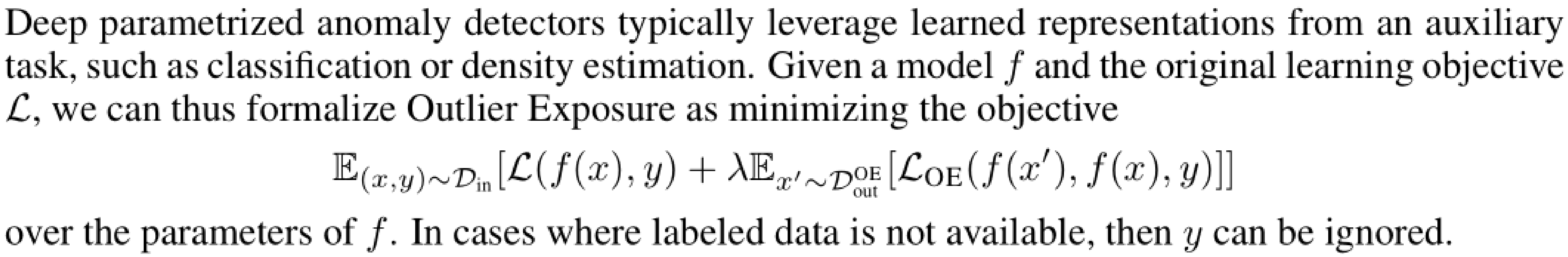
**Dan Hendrycks**
University of California, Berkeley
hendrycks@berkeley.edu

**Mantas Mazeika**
University of Chicago
mantas@ttic.edu

**Thomas Dietterich**
Oregon State University
tgd@oregonstate.edu

**内容提要：**

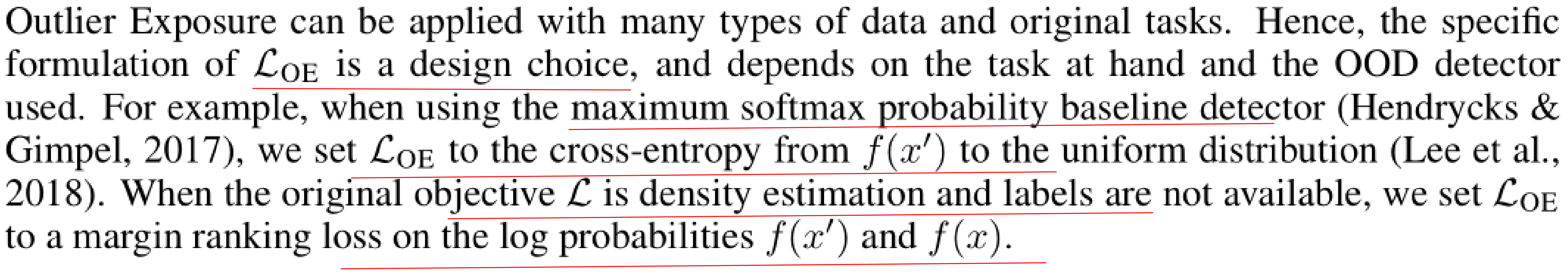
- 1、在部署机器学习系统时，异常输入的检测非常重要。在深度学习中越大和与复杂的输入数据将会放大异常数据和正常数据的识别难度。
- 与此同时，存在大量训练集外的数据。文章提出利用这些数据来提升深度异常检测，方法是通过这些异常值作为辅助数据集来训练检测器。称为“Outlier Exposure(OE)”



Deep parametrized anomaly detectors typically leverage learned representations from an auxiliary task, such as classification or density estimation. Given a model $f$ and the original learning objective $\mathcal{L}$, we can thus formalize Outlier Exposure as minimizing the objective

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{in}}}[\mathcal{L}(f(x),y)+\lambda\mathbb{E}_{x'\sim\mathcal{D}_{\text{out}}^{\text{OE}}}[\mathcal{L}_{\text{OE}}(f(x'),f(x),y)]]$$

over the parameters of $f$. In cases where labeled data is not available, then $y$ can be ignored.



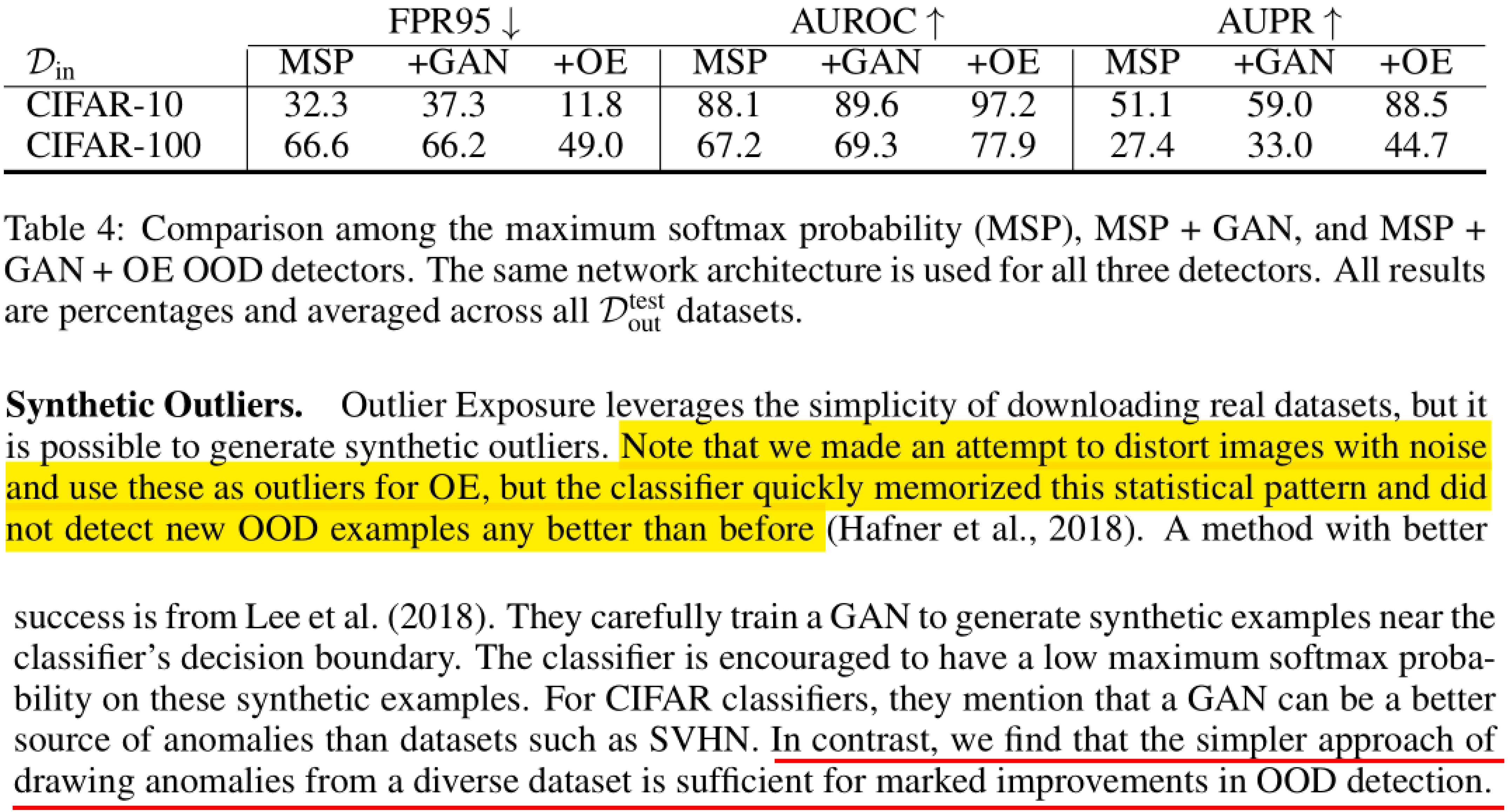
Outlier Exposure can be applied with many types of data and original tasks. Hence, the specific formulation of $\mathcal{L}_{\text{OE}}$ is a design choice, and depends on the task at hand and the OOD detector used. For example, when using the maximum softmax probability baseline detector (Hendrycks & Gimpel, 2017), we set $\mathcal{L}_{\text{OE}}$ to the cross-entropy from $f(x')$ to the uniform distribution (Lee et al., 2018). When the original objective $\mathcal{L}$ is density estimation and labels are not available, we set $\mathcal{L}_{\text{OE}}$ to a margin ranking loss on the log probabilities $f(x')$ and $f(x)$.

|  | FPR95 ↓ |  | AUROC ↑ |  | AUPR ↑ |  |
|---|---|---|---|---|---|---|
| $\mathcal{D}_{\text{in}}$ | MSP | +OE | MSP | +OE | MSP | +OE |
| SVHN | 6.3 | 0.1 | 98.0 | 100.0 | 91.1 | 99.9 |
| CIFAR-10 | 34.9 | 9.5 | 89.3 | 97.8 | 59.2 | 90.5 |
| CIFAR-100 | 62.7 | 38.5 | 73.1 | 87.9 | 30.1 | 58.2 |
| Tiny ImageNet | 66.3 | 14.0 | 64.9 | 92.2 | 27.2 | 79.3 |
| Places365 | 63.5 | 28.2 | 66.5 | 90.6 | 33.1 | 71.0 |

Table 1: Out-of-distribution image detection for the maximum softmax probability (MSP) baseline detector and the MSP detector after fine-tuning with Outlier Exposure (OE). Results are percentages and also an average of 10 runs. Expanded results are in Appendix A.

|  | FPR90 ↓ |  | AUROC ↑ |  | AUPR ↑ |  |
|---|---|---|---|---|---|---|
| $\mathcal{D}_{\text{in}}$ | MSP | +OE | MSP | +OE | MSP | +OE |
| 20 Newsgroups | 42.4 | 4.9 | 82.7 | 97.7 | 49.9 | 91.9 |
| TREC | 43.5 | 0.8 | 82.1 | 99.3 | 52.2 | 97.6 |
| SST | 74.9 | 27.3 | 61.6 | 89.3 | 22.9 | 59.4 |

Table 2: Comparisons between the MSP baseline and the MSP of the natural language classifier fine-tuned with OE. Results are percentages and averaged over 10 runs.



| | FPR95 ↓ | | | AUROC ↑ | | | AUPR ↑ | | |
|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{D}_{\text{in}}$ | MSP | +GAN | +OE | MSP | +GAN | +OE | MSP | +GAN | +OE |
| CIFAR-10 | 32.3 | 37.3 | 11.8 | 88.1 | 89.6 | 97.2 | 51.1 | 59.0 | 88.5 |
| CIFAR-100 | 66.6 | 66.2 | 49.0 | 67.2 | 69.3 | 77.9 | 27.4 | 33.0 | 44.7 |

Table 4: Comparison among the maximum softmax probability (MSP), MSP + GAN, and MSP + GAN + OE OOD detectors. The same network architecture is used for all three detectors. All results are percentages and averaged across all $\mathcal{D}_{\text{out}}^{\text{test}}$ datasets.

**Synthetic Outliers.** Outlier Exposure leverages the simplicity of downloading real datasets, but it is possible to generate synthetic outliers. Note that we made an attempt to distort images with noise and use these as outliers for OE, but the classifier quickly memorized this statistical pattern and did not detect new OOD examples any better than before (Hafner et al., 2018). A method with better success is from Lee et al. (2018). They carefully train a GAN to generate synthetic examples near the classifier's decision boundary. The classifier is encouraged to have a low maximum softmax probability on these synthetic examples. For CIFAR classifiers, they mention that a GAN can be a better source of anomalies than datasets such as SVHN. In contrast, we find that the simpler approach of drawing anomalies from a diverse dataset is sufficient for marked improvements in OOD detection.

| | | FPR95 ↓ | | AUROC ↑ | | AUPR ↑ | |
|---|---|---|---|---|---|---|---|
| $\mathcal{D}_{\text{in}}$ | $\mathcal{D}_{\text{out}}^{\text{test}}$ | BPP | +OE | BPP | +OE | BPP | +OE |
| CIFAR-10 | Gaussian | 0.0 | 0.0 | 100.0 | 100.0 | 100.0 | 99.6 |
| | Rademacher | 61.4 | 50.3 | 44.2 | 56.5 | 14.2 | 17.3 |
| | Blobs | 17.2 | 1.3 | 93.2 | 99.5 | 60.0 | 96.2 |
| | Textures | 96.8 | 48.9 | 69.4 | 88.8 | 40.9 | 70.0 |
| | SVHN | 98.8 | 86.9 | 15.8 | 75.8 | 9.7 | 60.0 |
| | Places365 | 86.1 | 50.3 | 74.8 | 89.3 | 38.6 | 70.4 |
| | LSUN | 76.9 | 43.2 | 76.4 | 90.9 | 36.5 | 72.4 |
| | CIFAR-100 | 96.1 | 89.8 | 52.4 | 68.5 | 19.0 | 41.9 |
| Mean | | 66.6 | **46.4** | 65.8 | **83.7** | 39.9 | **66.0** |

Table 5: OOD detection results with a PixelCNN++ density estimator, and the same estimator after applying OE. The model's bits per pixel (BPP) scores each sample. All results are percentages. Test distributions $\mathcal{D}_{\text{out}}^{\text{test}}$ are described in Appendix A.

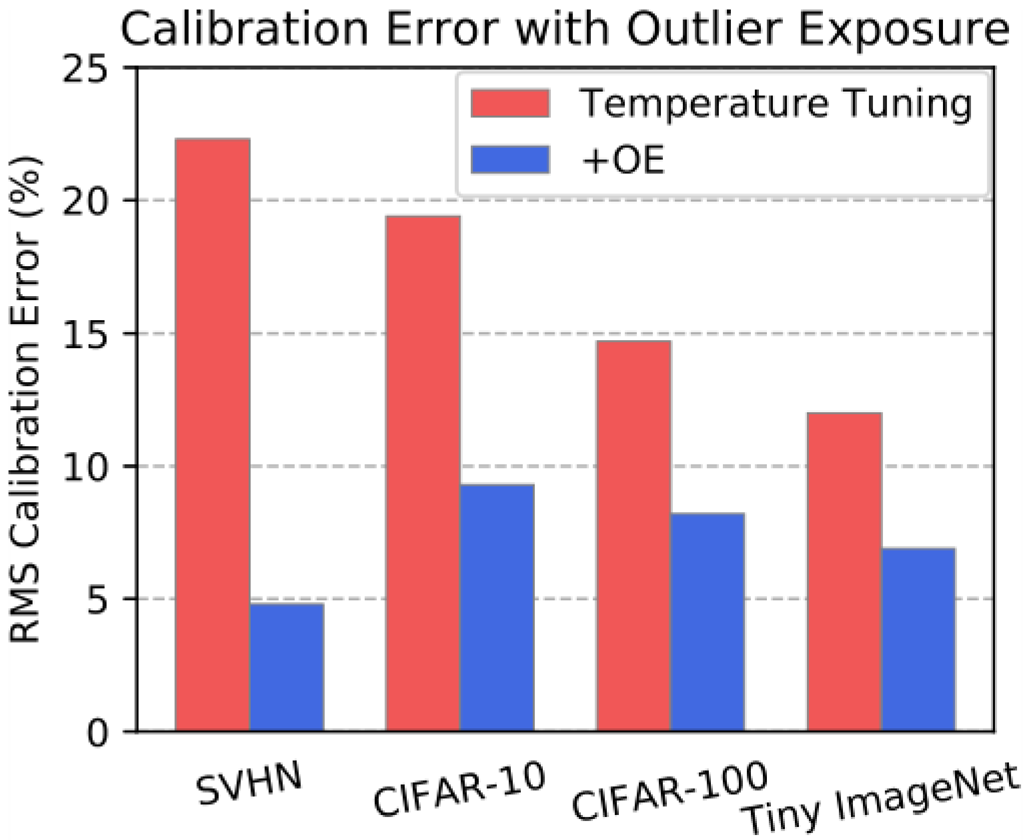
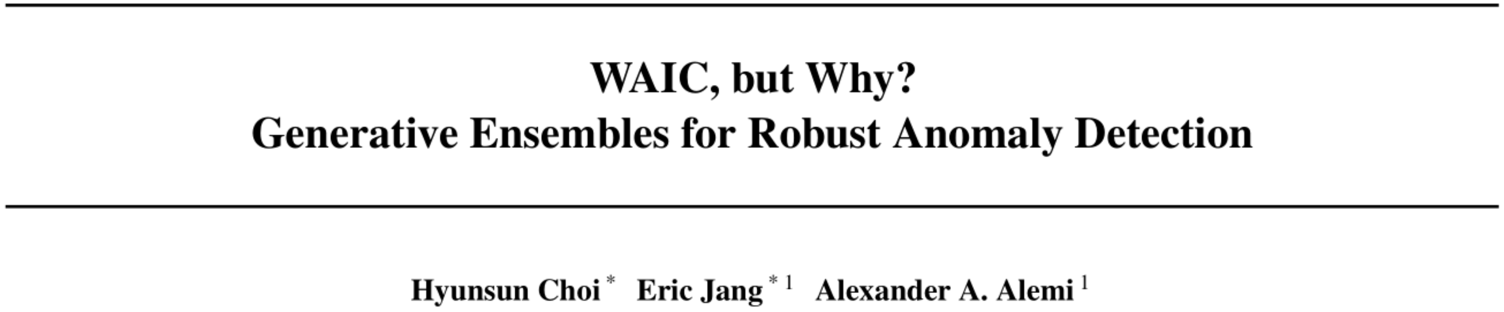


Figure 3: Root Mean Square Calibration Error values with temperature tuning and temperature tuning + OE across various datasets.



**WAIC, but Why?**
**Generative Ensembles for Robust Anomaly Detection**

**Hyunsun Choi**[*] **Eric Jang**[*1] **Alexander A. Alemi**[1]

## 内容提要：

- 1、对于OoD数据，通常是利用似然估计来训练一个数据分布，然后用这个模型来排除不可能的输入。
- 2、但是基于自然数据的似然模型本身就容易出现OoD误差，经常会给OoD数据分配很大的似然值。
- 3、文章提出了一个方式“Generative Ensembles”,这种方式是通过“estimating epistemic uncertainty of the likelihood model”

### 3.1. Watanabe Akaike Information Criterion (WAIC)

First introduced in Watanabe (2010), the Watanabe-Akaike Information Criterion (WAIC) gives an asymptotically correct estimate of the gap between the training set and test set expectations. If we had access to samples from the true Bayesian posterior of a model, we could compute a corrected version of the expected log (Watanabe, 2009)[1]:

$$\mathrm{WAIC}(x) = \mathbb{E}_\theta[\log p_\theta(\mathrm{x})] - \mathrm{Var}_\theta \left[\log p_\theta(\mathrm{x})\right] \tag{1}$$

The correction term subtracts the variance in likelihoods across independent samples from the posterior. This acts to robustify our estimate, ensuring that points that are sensitive to the particular choice of posterior parameters are penalized.

In this work we do not have exact posterior samples, so we instead utilize independently trained model instances as a proxy for posterior samples, following (Lakshminarayanan et al., 2017). Being trained with Stochastic Gradient Descent (SGD), the independent models in the ensemble act as approximate posterior samples (Mandt et al., 2017).

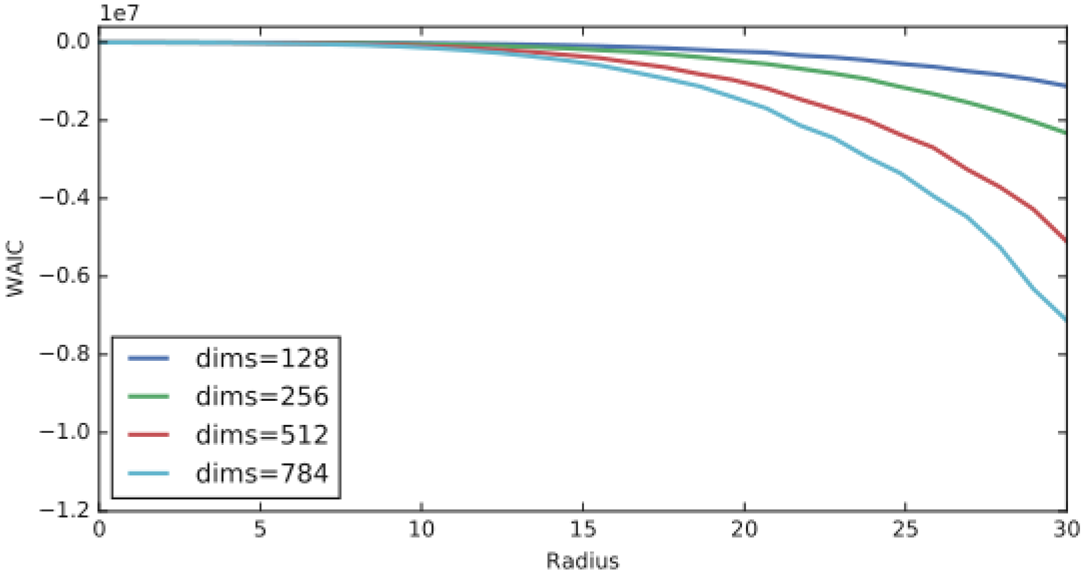


*Figure 2.* WAIC estimated using Jackknife resampling of data points drawn from an isotropic Gaussian, for an ensemble of size $N = 10$. Lines correspond to a dimensionality of the data.

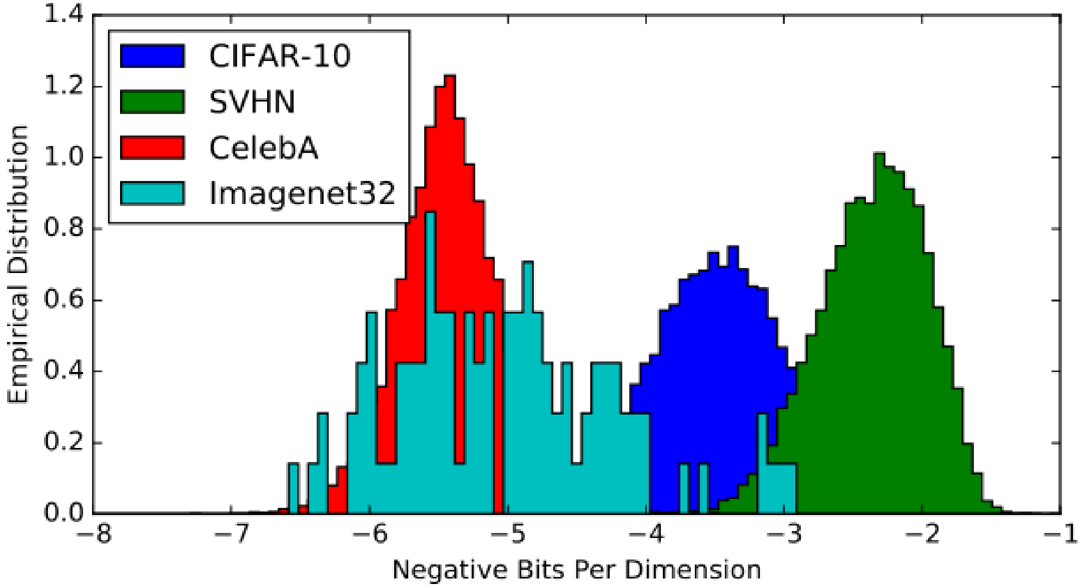


*Figure 1.* Density estimation models are not robust to OoD inputs. A GLOW model (Kingma & Dhariwal, 2018) trained on CIFAR-10 assigns much higher likelihoods to samples from SVHN than samples from CIFAR-10. .

# WAIC

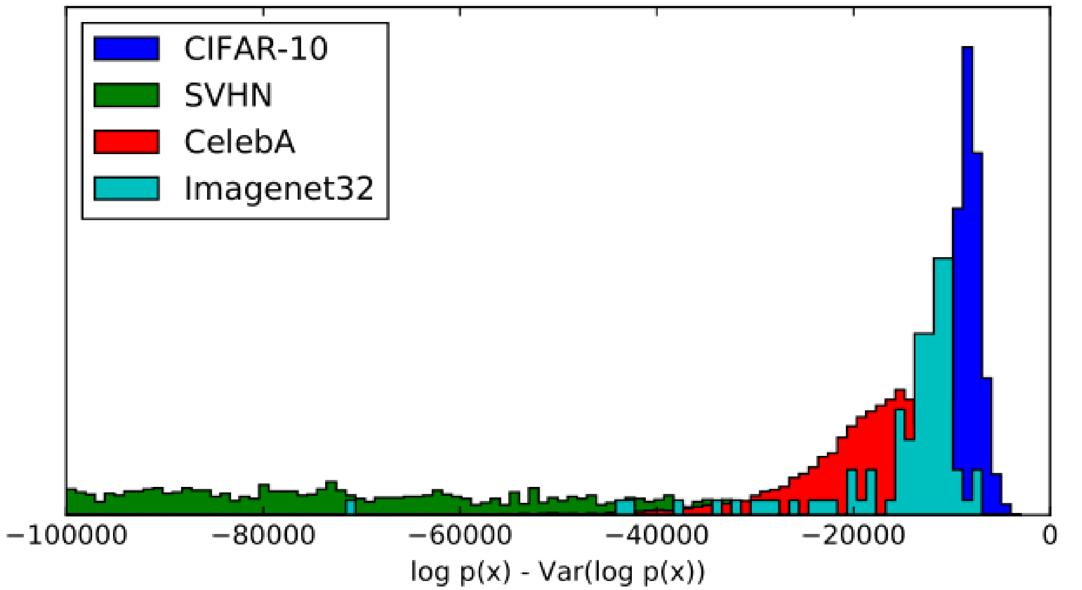


*Figure 3.* We use ensembles of generative models to implement the Watanabe-Akaike Information Criterion (WAIC), which combines density estimation with uncertainty estimation. Histograms correspond to predictions over test sets from each dataset.

*Table 1.* We train models on MNIST, Fashion MNIST, and CIFAR-10 and compare OoD classification ability to baseline methods using the threshold-independent Area Under ROC curve metric (AUROC). $p_\theta(\mathrm{x})$ is a single-model likelihood approximation (IWAE for VAE, log-density for Glow). WAIC is the Watanabe-Akaike Information Criterion as estimated by the Generative Ensemble. ODIN results reproduced from (Liang et al., 2018). Best results for each task shown in bold.

| OoD | ODIN | VIB | Rate | $p_\theta(\mathrm{x})$ | WAIC |
|---|---|---|---|---|---|
| MNIST | VAE | | | | |
| Omniglot | **100** | 97.1 | 99.1 | 97.9 | 98.5 |
| notMNIST | 98.2 | 98.6 | 99.9 | **100** | **100** |
| FashionMNIST | N/A | 85.0 | **100** | **100** | **100** |
| Uniform | **100** | 76.6 | **100** | **100** | **100** |
| Gaussian | **100** | 99.2 | **100** | **100** | **100** |
| HFlip | N/A | 63.7 | 60.0 | 84.1 | **85.0** |
| VFlip | N/A | 75.1 | 61.8 | 80.0 | **81.3** |
| Adv | N/A | N/A | **100** | 0 | **100** |
| FashionMNIST | VAE | | | | |
| Omniglot | N/A | **94.3** | 83.2 | 56.8 | 79.6 |
| notMNIST | N/A | 89.6 | 92.8 | 92.0 | **98.7** |
| MNIST | N/A | **94.1** | 87.1 | 42.3 | 76.6 |
| Uniform | N/A | 79.6 | 99.0 | **100** | **100** |
| Gaussian | N/A | 89.3 | **100** | **100** | **100** |
| HFlip | N/A | **66.7** | 53.4 | 59.4 | 62.4 |
| VFlip | N/A | **90.2** | 58.6 | 66.8 | 74.0 |
| Adv | N/A | N/A | **100** | 0.1 | **100** |
| OoD | ODIN | VIB | $\lVert d \rVert$ | $p_\theta(\mathrm{x})$ | WAIC |
| CIFAR-10 | Glow | | | | |
| CelebA | 85.7 | 73.5 | 22.9 | 75.6 | **99.7** |
| SVHN | 89.9 | 52.8 | 74.4 | 7.5 | **100** |
| ImageNet32 | 98.5 | 70.1 | 12.3 | 93.8 | **95.6** |
| Uniform | 99.9 | 54.0 | **100** | **100** | **100** |
| Gaussian | **100** | 45.8 | **100** | **100** | **100** |
| HFlip | 50.1 | **50.6** | 46.2 | 50.1 | 50.0 |
| VFlip | 84.2 | **51.2** | 44.0 | 50.6 | 50.4 |

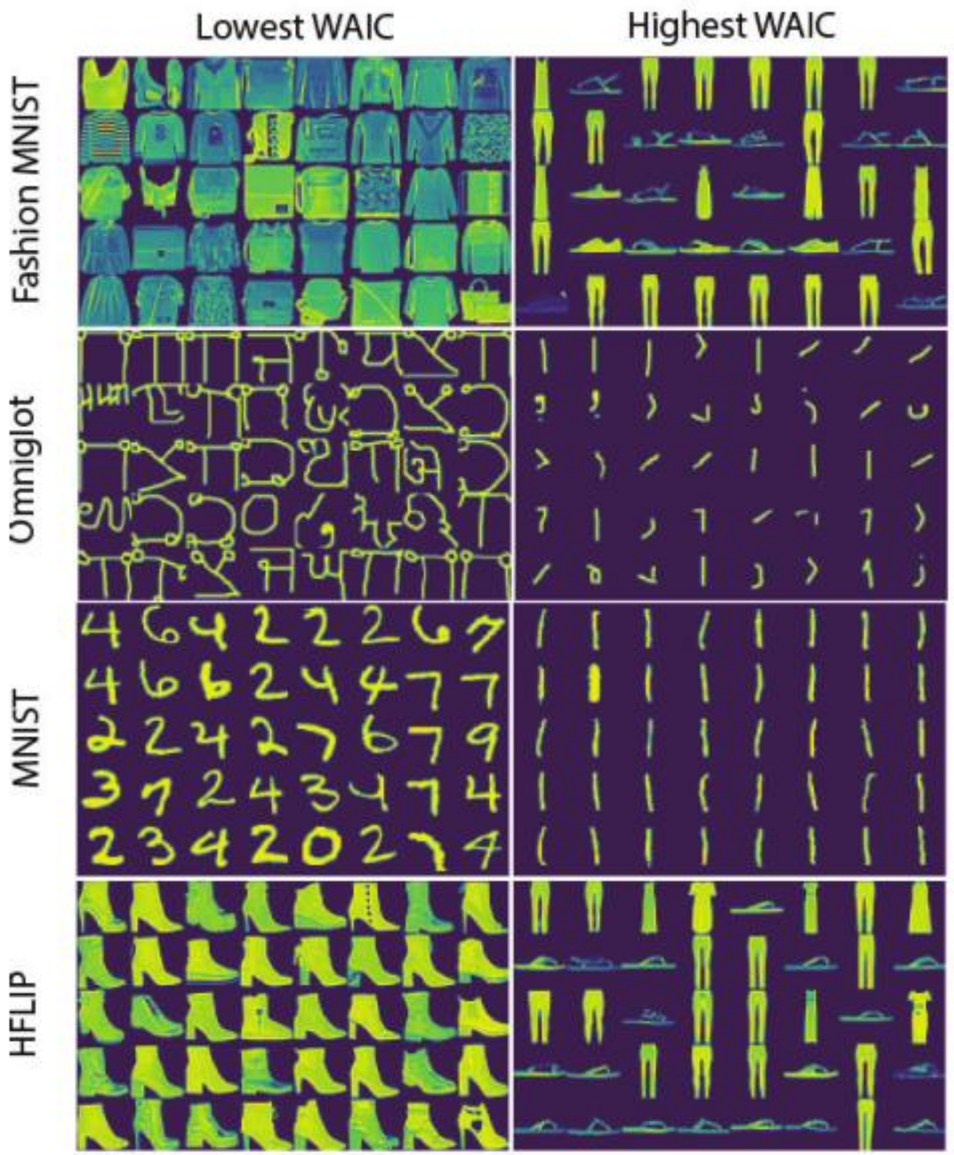


*Figure 6.* Left: lowest WAIC for each evaluation dataset. Right: highest WAIC for each evaluation dataset.

The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

# Flow-GAN: Combining Maximum Likelihood and Adversarial Learning in Generative Models

**Aditya Grover, Manik Dhar, Stefano Ermon**
Department of Computer Science
Stanford University
{adityag, dmanik, ermon}@cs.stanford.edu

## 4 Hybrid learning of Flow-GANs

In the previous section, we observed that adversarially learning Flow-GANs models attain poor held-out log-likelihoods. This makes it challenging to use such models for applications requiring density estimation. On the other hand, Flow-GANs learned using MLE are "mode covering" but do not generate high quality samples. With a Flow-GAN, it is possible to trade-off the two goals by combining the learning objectives corresponding to both these inductive principles. Without loss of generality, let $V(G_\theta, D_\phi)$ denote the minimax objective of any GAN model (such as WGAN). The hybrid objective of a Flow-GAN can be expressed as:

$$\min_{\theta} \max_{\phi} V(G_\theta, D_\phi) - \lambda \mathbb{E}_{\mathbf{x} \sim P_{\text{data}}} [\log p_\theta(\mathbf{x})] \tag{7}$$

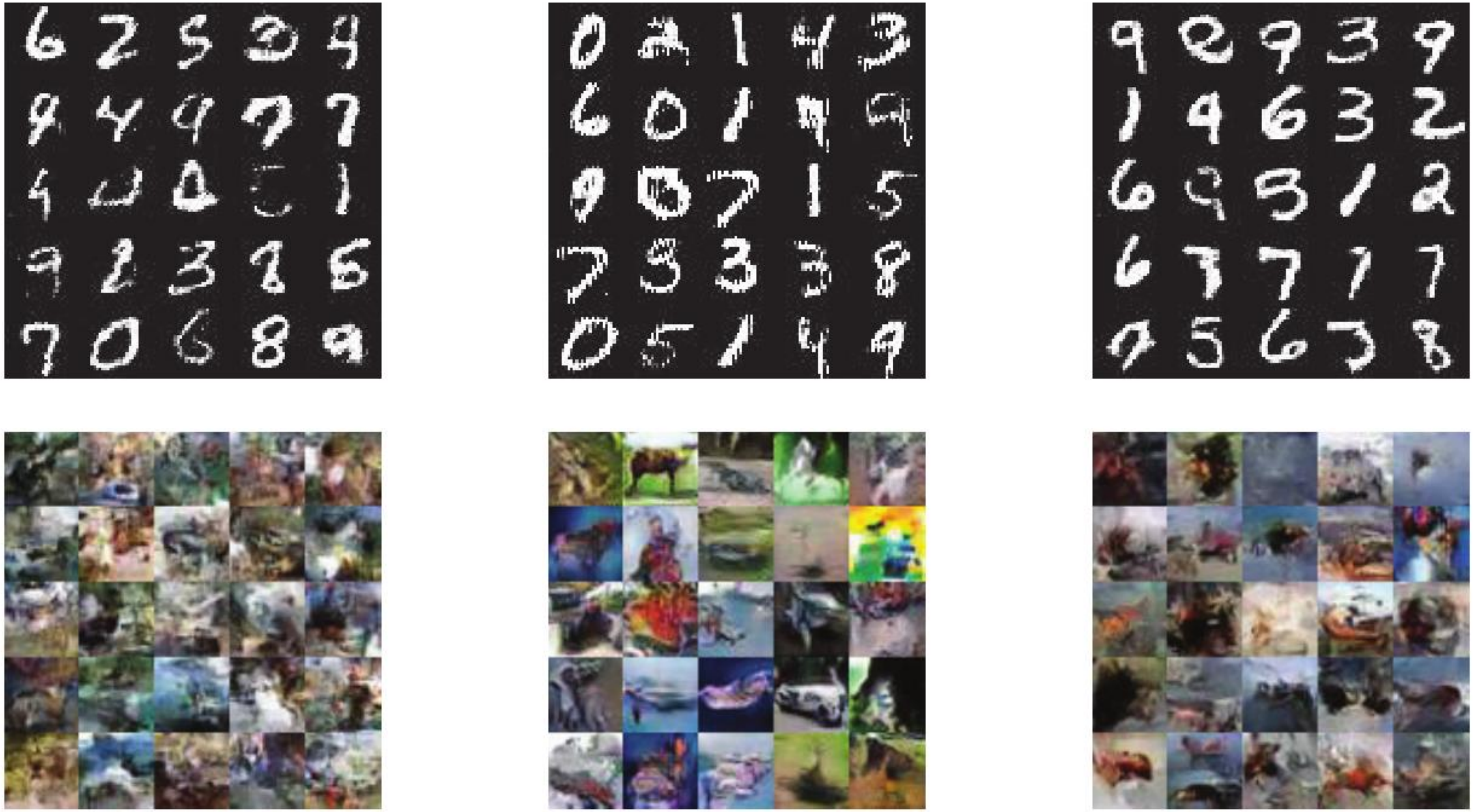
where $\lambda \geq 0$ is a hyperparameter for the algorithm. By varying $\lambda$, we can interpolate between plain adversarial training ($\lambda = 0$) and MLE (very high $\lambda$).

Table 1: Best MODE scores and test negative log-likelihood estimates for Flow-GAN models on MNIST.

| Objective | MODE Score | Test NLL (in nats) |
|---|---|---|
| MLE | $7.42$ | $-3334.56$ |
| ADV | $9.24$ | $-1604.09$ |
| Hybrid ($\lambda = 0.1$) | $\mathbf{9.37}$ | $\mathbf{-3342.95}$ |

Table 2: Best Inception scores and test negative log-likelihood estimates for Flow-GAN models on CIFAR-10.

| Objective | Inception Score | Test NLL (in bits/dim) |
|---|---|---|
| MLE | $2.92$ | $\mathbf{3.54}$ |
| ADV | $\mathbf{5.76}$ | $8.53$ |
| Hybrid ($\lambda = 1$) | $3.90$ | $4.21$ |

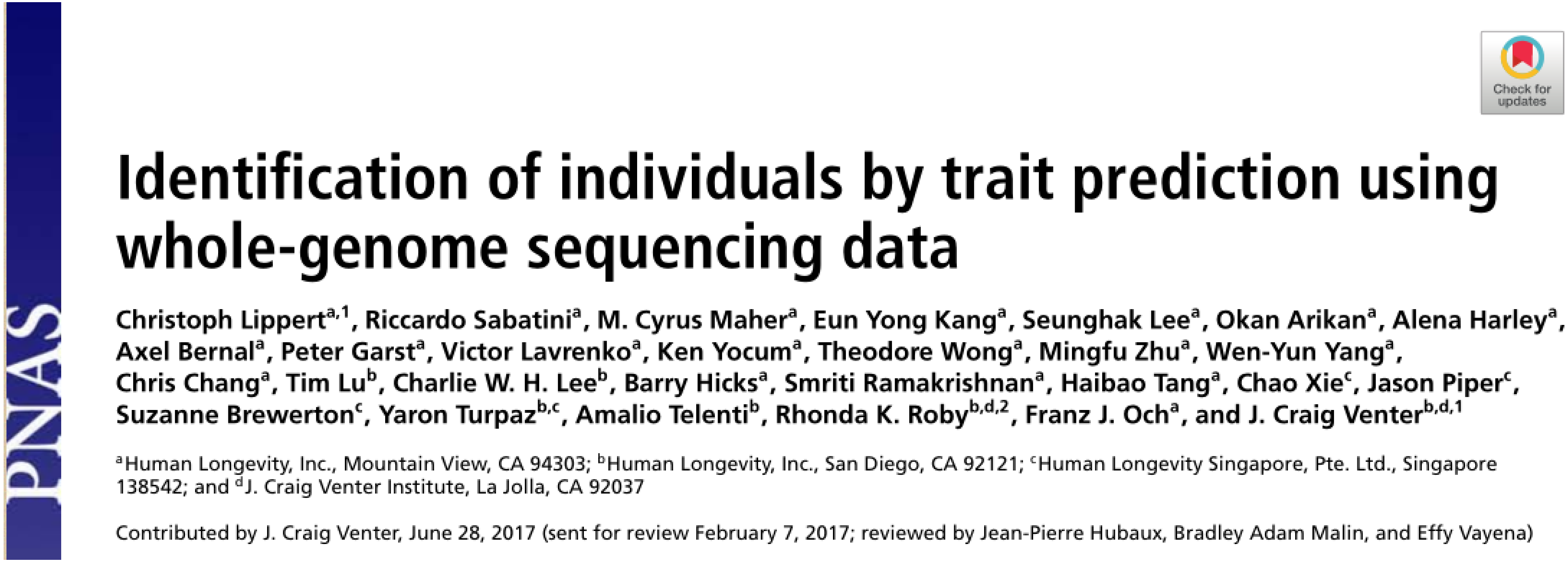


(a) MLE

(b) ADV

(c) Hybrid

Figure 1: Samples generated by Flow-GAN models with different objectives for MNIST (**top**) and CIFAR-10 (**bottom**).



# Identification of individuals by trait prediction using whole-genome sequencing data

Christoph Lippert[a,1], Riccardo Sabatini[a], M. Cyrus Maher[a], Eun Yong Kang[a], Seunghak Lee[a], Okan Arikan[a], Alena Harley[a], Axel Bernal[a], Peter Garst[a], Victor Lavrenko[a], Ken Yocum[a], Theodore Wong[a], Mingfu Zhu[a], Wen-Yun Yang[a], Chris Chang[a], Tim Lu[b], Charlie W. H. Lee[b], Barry Hicks[a], Smriti Ramakrishnan[a], Haibao Tang[a], Chao Xie[c], Jason Piper[c], Suzanne Brewerton[c], Yaron Turpaz[b,c], Amalio Telenti[b], Rhonda K. Roby[b,d,2], Franz J. Och[a], and J. Craig Venter[b,d,1]

[a]Human Longevity, Inc., Mountain View, CA 94303; [b]Human Longevity, Inc., San Diego, CA 92121; [c]Human Longevity Singapore, Pte. Ltd., Singapore 138542; and [d]J. Craig Venter Institute, La Jolla, CA 92037

Contributed by J. Craig Venter, June 28, 2017 (sent for review February 7, 2017; reviewed by Jean-Pierre Hubaux, Bradley Adam Malin, and Effy Vayena)

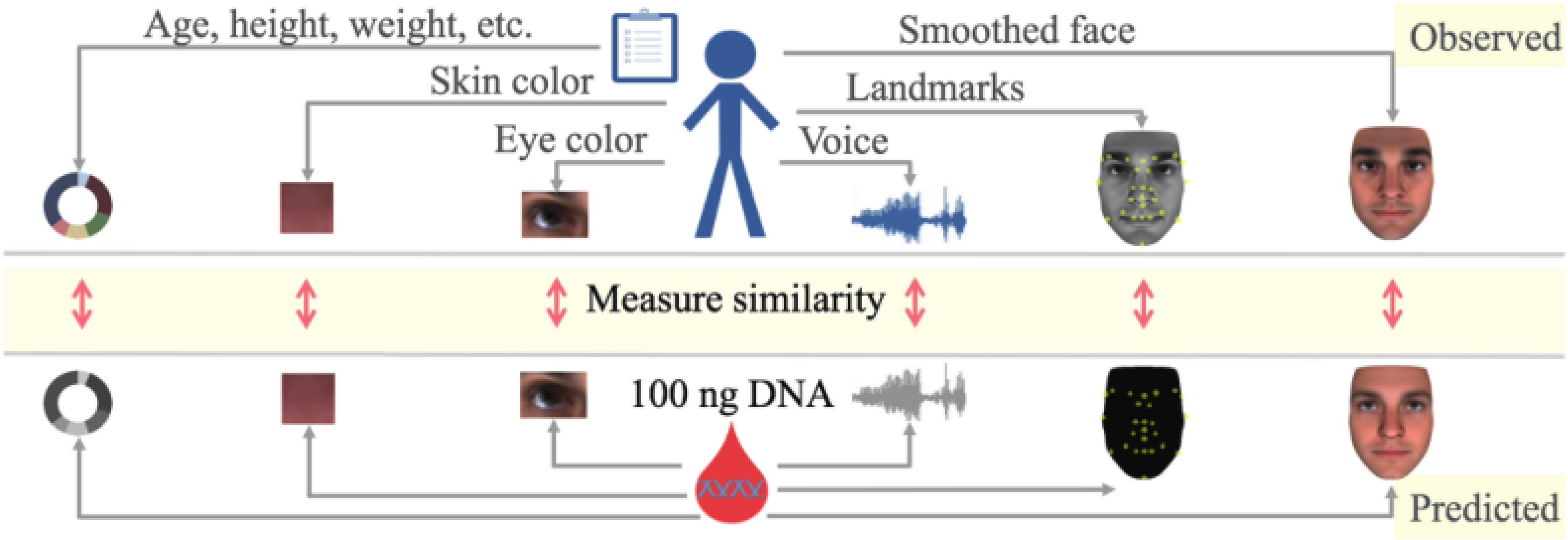
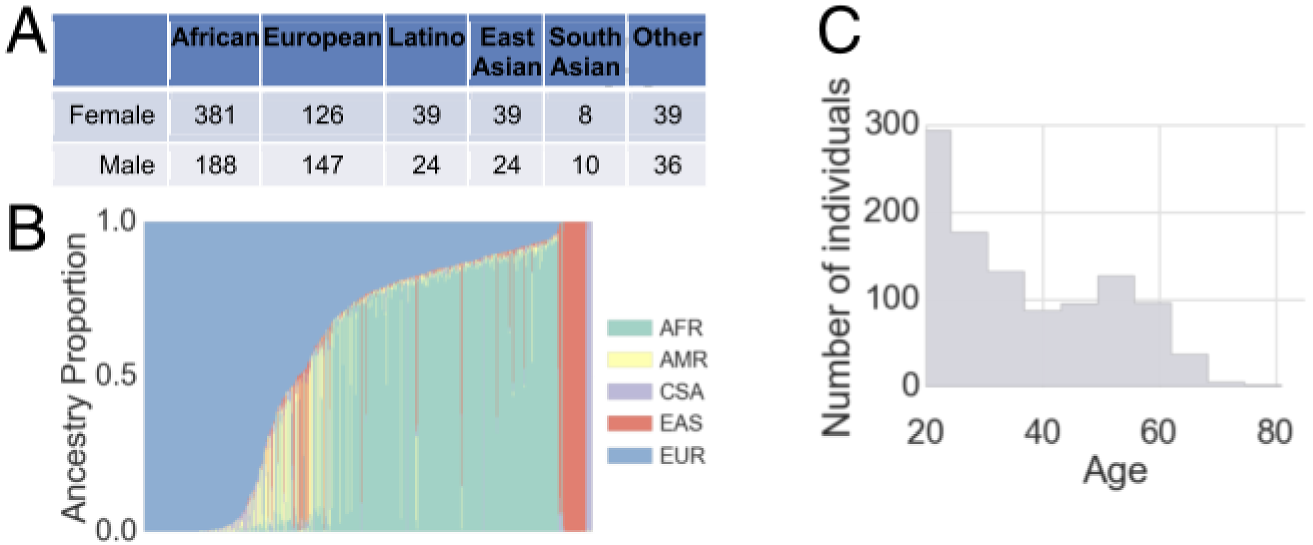


**Fig. 6.** Overview of the experimental approach. A DNA sample and a variety of phenotypes are collected for each individual. We used predictive modeling to derive a common embedding for phenotypes and the genomic sample as detailed in *SI Appendix*, Table S14. The concordance between genomic and phenotypic embeddings are used to match an individual's phenotypic profile to the DNA sample.

$$\delta_{\mathcal{P}}\left(\psi_{\mathcal{P}}\left(p\right), \phi_{\mathcal{P}}\left(g\right)\right) = \sum_{d=1}^{D} w_d \left|\psi_{\mathcal{P}}(p)_d - \phi_{\mathcal{P}}(g)_d\right|, \qquad [1]$$

where the weights $w_d$, which reflect the importance of $d$-th dimension of $\mathcal{E}_{\mathcal{P}}$, have been trained using a maximum entropy model (53).



| | African | European | Latino | East Asian | South Asian | Other |
|---|---|---|---|---|---|---|
| Female | 381 | 126 | 39 | 39 | 8 | 39 |
| Male | 188 | 147 | 24 | 24 | 10 | 36 |

**Fig. 1.** Study overview. (*A*) Distribution of self-reported ethnicity in the study. (*B*) Inferred genomic ancestry proportions for each study participant. Ancestry components are African (AFR), Native American (AMR), Central South Asian (CSA), East Asian (EAS), and European (EUR). (*C*) Distribution of ages in the study.

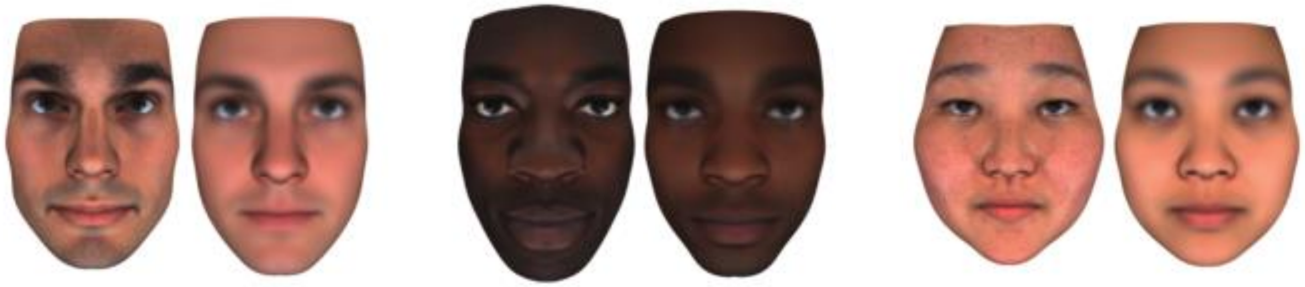


**Fig. 2.** Examples of real (*Left*) and predicted (*Right*) faces.