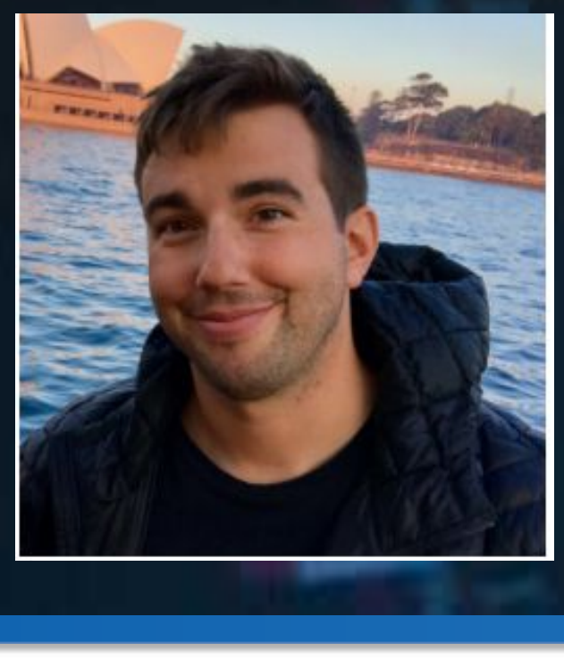
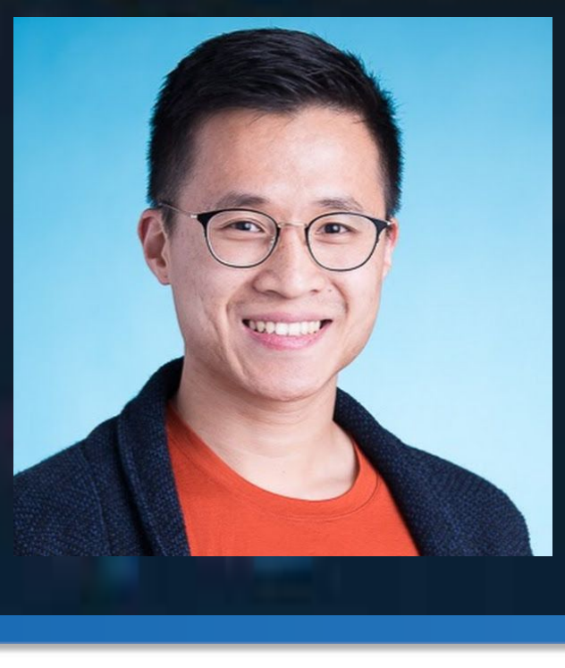
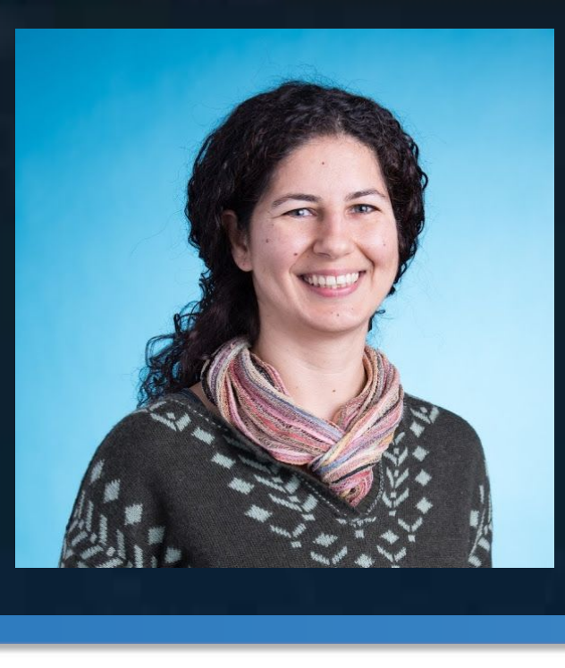
# Hybrid Models with Deep and Invertible Features

Eric Nalisnick*, Akihiro Matsukawa*, Yee Whye Teh, Dilan Gorur,  Balaji Lakshminarayanan

* equal contribution

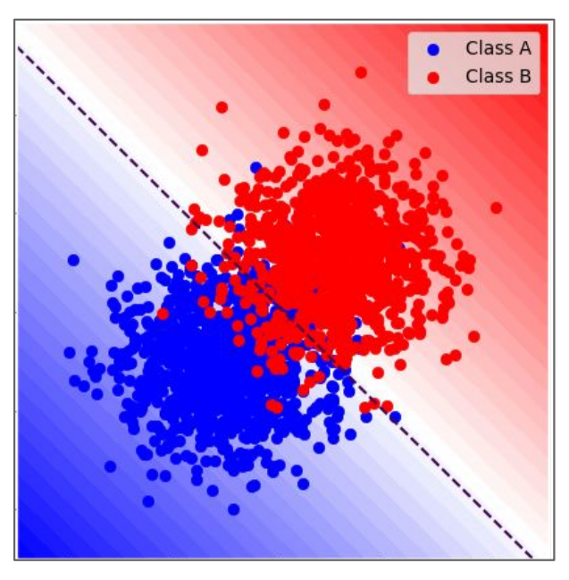






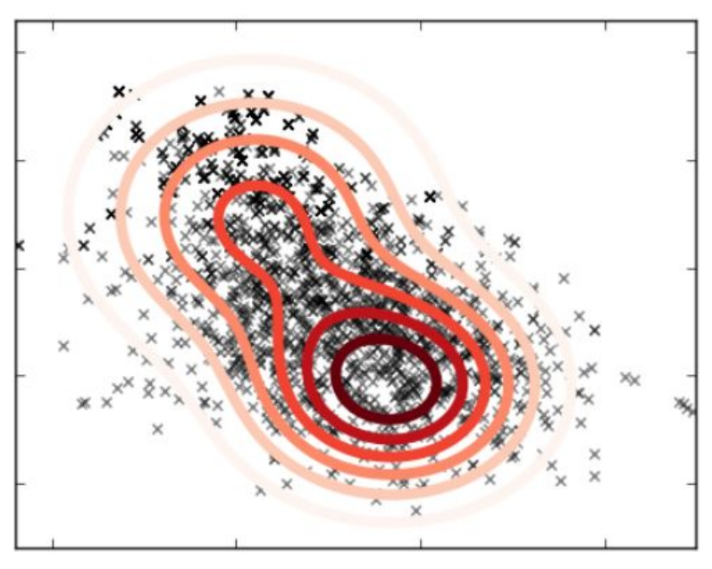
DeepMind

## 1. Hybrid Models



$p(\mathbf{y}|\mathbf{x})$

**Discriminative Model**



$p(\mathbf{x})$

**Generative Model**

Deep neural networks commonly model conditional distributions of the form *p(**y**|**x**)*, where ***y*** denotes a label and ***x*** features. However, in many applications, modeling just the conditional distribution is insufficient. For instance, if we believe that the model may be subjected to inputs unlike those of the training data, a model for *p(**x**)* can possibly detect an outlier before it is passed along for prediction. Thus, modeling the joint distribution *p(**y**, **x**)* provides a richer representation of the data than a conditional one. Models of this form are known as **hybrid models** [Ng & Jordan, 2002; Raina et al., 2004; Lasserre et al., 2006] as they are **defined by combining discriminative (i.e. *p(y|x)*) and generative (i.e. *p(x)*) components.**.

## 2. Background

Our hybrid model will be composed of two primary building blocks: **flow-based, invertible generative models** and **generalized linear models**.

### INVERTIBLE GENERATIVE MODELS

**Deep invertible generative models** are simply high-capacity, bijective transformations with a tractable Jacobian matrix and inverse. The best known models of this class are the **real non-volume preserving (RNVP) transform** [Dinh et al., 2017] and its recent extension, the **Glow transform** [Kingma & Dhariwal, 2018]. The bijective nature of these transforms is crucial as it allows us to **employ the change-of-variables (COV) formula for exact density evaluation:**

$$\log p_x(\boldsymbol{x}) = \log p_z(f(\boldsymbol{x};\boldsymbol{\phi})) + \log\left|\frac{\partial \boldsymbol{f}_\phi}{\partial \boldsymbol{x}}\right|$$

where *f* denotes the transform with parameters **φ**, |∂***f*** ⁄ ∂***x***| the determinant of the Jacobian of the transform, and *p(**z**)* a distribution on the latent variables. The modeler is free to choose *p(**z**)*, and therefore it is often set as a factorized standard Gaussian for computational simplicity. The parameters **φ** are estimated via maximizing the exact log-likelihood.

### GENERALIZED LINEAR MODELS

**Generalized linear models (GLMs)** [Nelder & Baker, 1972] model the expected response ***y*** as follows:

$$\mathbb{E}[y_n|\mathbf{z}_n] = g^{-1}\left(\boldsymbol{\beta}^T\mathbf{z}_n + \beta_0\right)$$

- *E[**y**|**z**]* denotes the expected value of ***y***
- ***β*** is a d-dimensional vector of real-valued parameters, $\beta_o$ is a scalar bias, ***z*** are covariates
- $g^{-1}(\cdot)$ a **link function** such that $g^{-1}$: R→$\mu_{y|z}$
- Bayesian GLM could be defined by specifying a prior *p(**β**)* and computing the posterior *p(**β**|**y**, **Z**)*.

## 3. Combining Deep Invertible Transforms and GLMs

We propose a **hybrid model** architecture consisting of a **deep invertible transform coupled with a GLM**. Together the two define a deep predictive model with both **the ability to compute *p(x)* and *p(y|x)* exactly, in a single feed-forward pass**. Let ***θ*** = {**φ**, **β**, $\beta_0$} denote the set of generative and discriminative parameters. The model defines the following joint distribution over a label-feature pair *(**x**, **y**)*:

$$p(y_n, \boldsymbol{x}_n; \boldsymbol{\theta}) = p(\mathbf{y}_n|\boldsymbol{x}_n; \boldsymbol{\phi}, \boldsymbol{\beta}, \beta_0)\ \ p(\boldsymbol{x}_n; \boldsymbol{\phi})$$

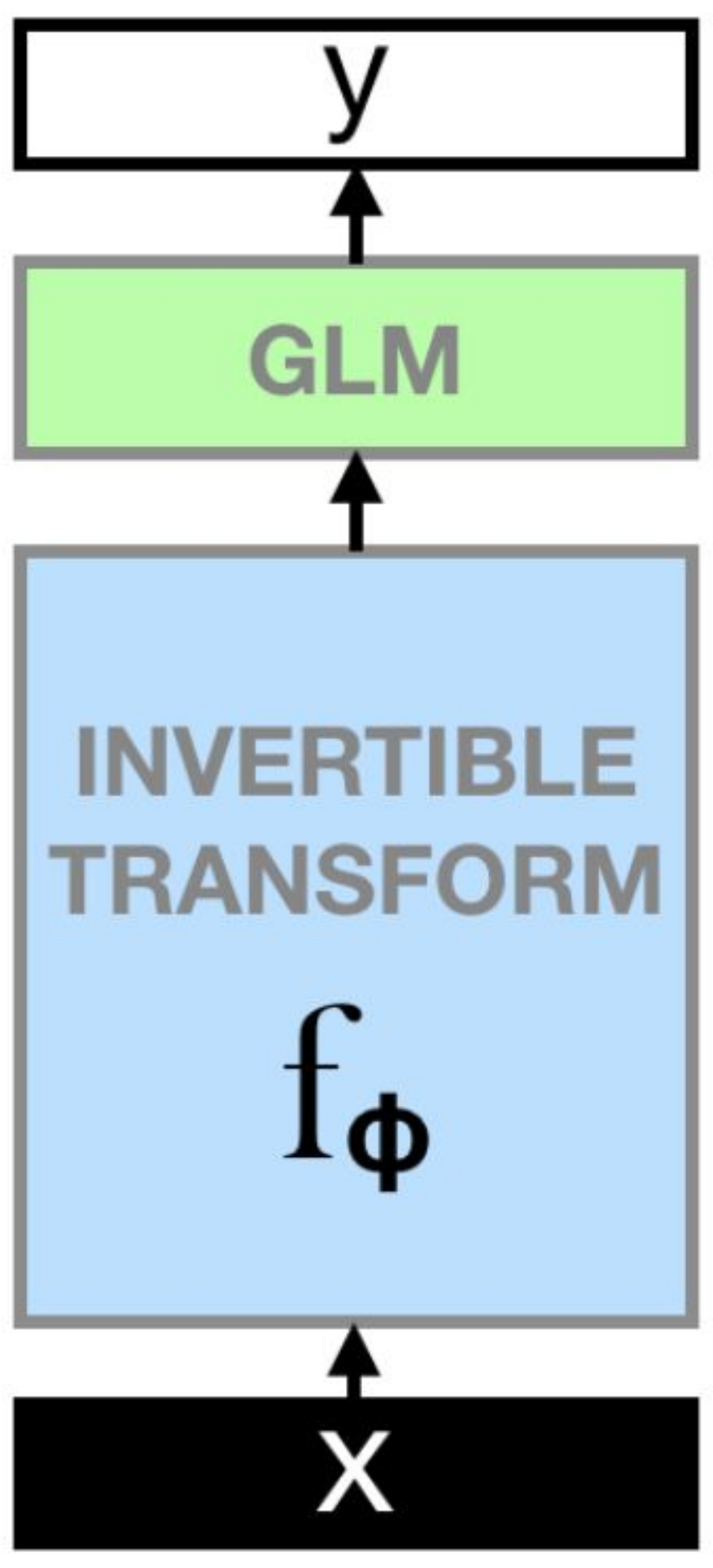
$$= p(\mathbf{y}_n|f(\boldsymbol{x}_n; \boldsymbol{\phi}); \boldsymbol{\beta}, \beta_0)\ \ p_z(f(\boldsymbol{x}_n; \boldsymbol{\phi}))\ \left|\frac{\partial \boldsymbol{f}_\phi}{\partial \boldsymbol{x}_n}\right|$$

where ***z*** = *f(**x**, **φ**)* is the output of the invertible transformation, *p(**z**)* is the latent distribution, and p(**y**|f(**x**, **φ**) ; **β**, $\beta_0$) is a GLM with the latent variables as its input features. The figure to the right shows a diagram of the computational pipeline. We term this model ***DIGLM*: Deep Invertible Generalized Linear Model**.

In practice we found better performance is obtained by scaling the contribution of *p(**x**)* to account for the drastic difference in dimensionality between **y** and **x**. We denote this modified objective as:

$$\mathcal{J}_\lambda(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(y_n|\mathbf{x}_n; \boldsymbol{\theta}) + \lambda \log p(\mathbf{x}_n; \boldsymbol{\phi})$$

where ***λ* is the scaling constant.** Weighted losses are commonly used in hybrid models [Lasserre et al., 2006; McCallum et al., 2006; Kingma et al., 2014].



**Model Architecture**

## 4. Experiments

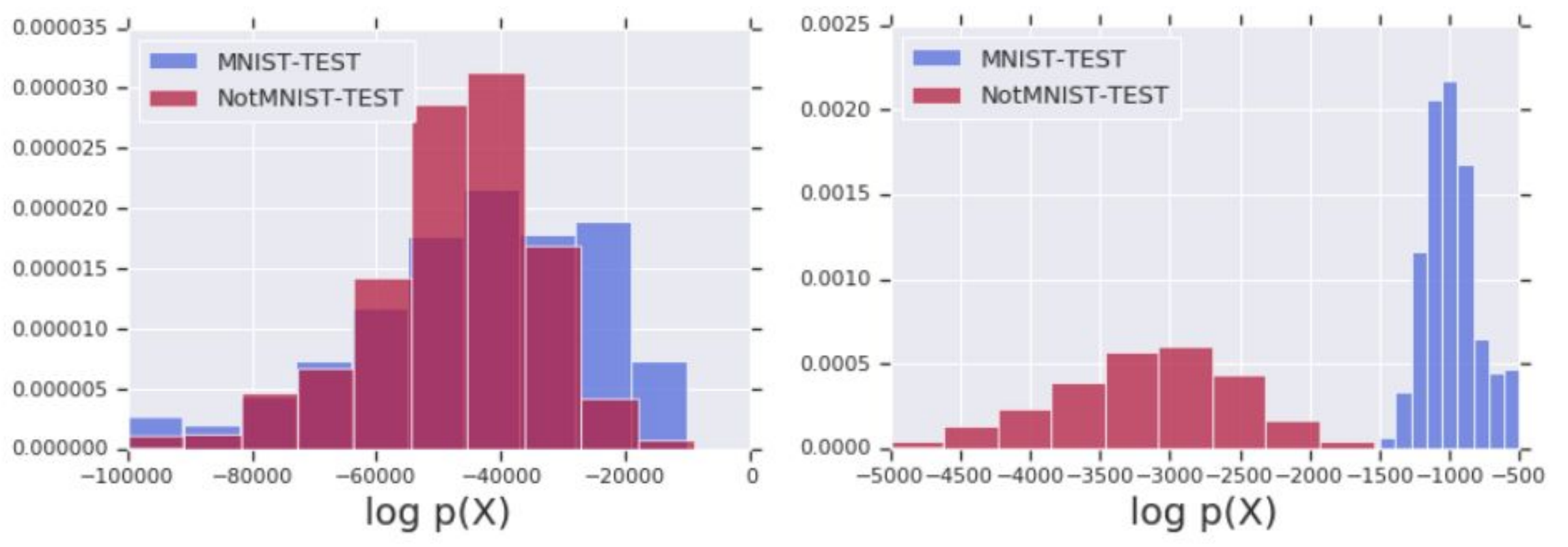
### MNIST CLASSIFICATION

We train a DIGLM on the MNIST dataset with *Glow* [Kingma & Dhariwal, 2018] as the invertible architecture. We compare the hybrid model to the discriminative model obtained by setting *λ = 0*. We compare **test classification error**, **negative log-likelihood (NLL)**, and **entropy** of the predictive distribution *p(**y**|**x**)*. Following Lakshminarayanan et al. (2017), we evaluate on both the MNIST test set and the out-of-distribution (OOD) NotMNIST test set. The OOD test is a proxy to test if the system exhibits higher uncertainty on inputs not seen during training data.

| Model | MNIST-error ↓ | MNIST-NLL ↓ | NotMNIST-NLL ↓ | NotMNIST-Entropy ↑ |
|---|---|---|---|---|
| Discriminative ($\lambda = 0$) | **0.67%** | 0.081 | 29.27 | 0.1302 |
| Hybrid ($\lambda = 0.01/D$) | 0.73% | **0.033** | **10.10** | **0.3695** |

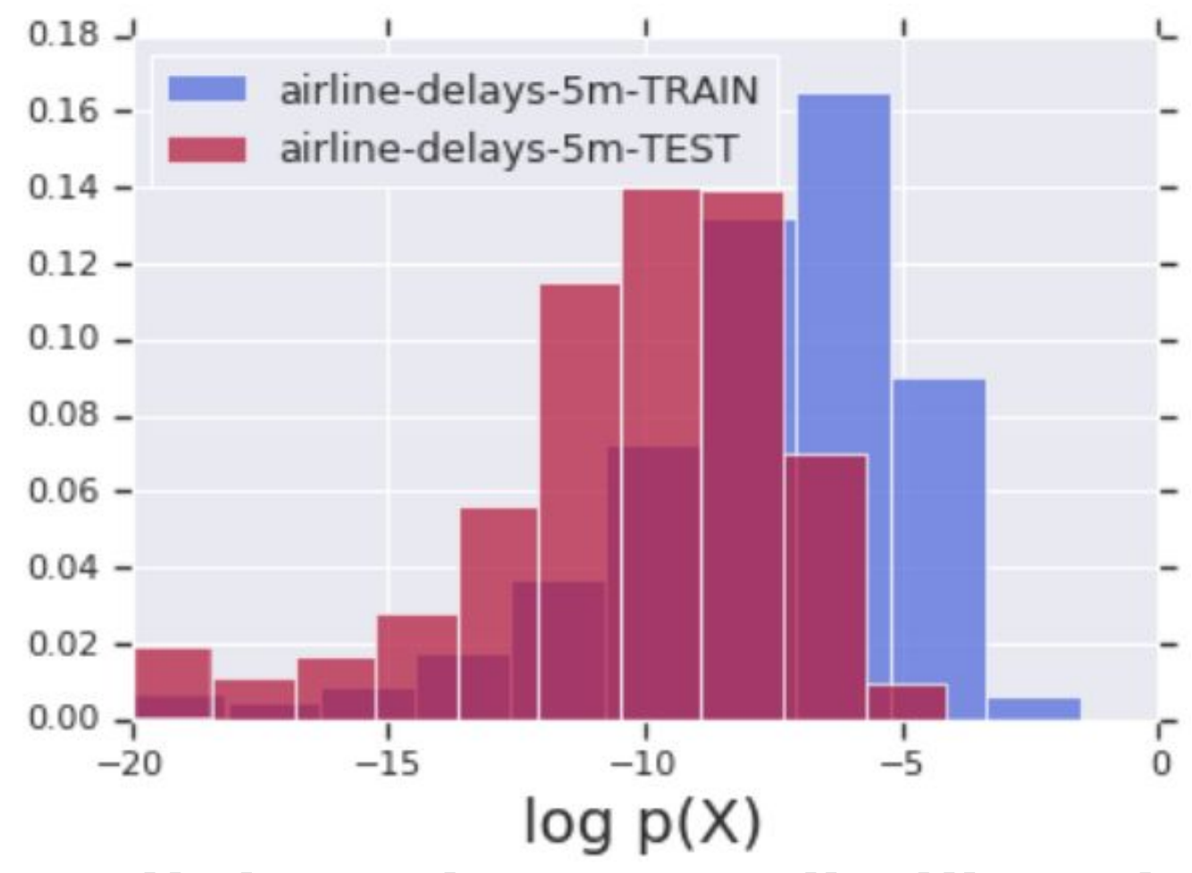
Table 1: Results comparing hybrid model to discriminative model.

The results are shown in Table 1. The discriminative model achieves slightly lower test error, however **the hybrid model achieves better NLL and entropy**. Next, we compare the generative density *p(**x**)* for the hybrid model as well as the discriminative model. In the histogram to the right, we see that **the pure discriminative model (left) assigns similar density to the OOD inputs**. On the other hand, the **hybrid model (right) evaluates to lower density values for NotMNIST (OOD)**. Hence the hybrid model allows a user to abstain from trusting the model's predictions when the density is lower.



Histogram of $\log p(\boldsymbol{x})$ for discriminative (left) and hybrid (right)

### REGRESSION ON FLIGHT DELAY DATA SET



**Flight Delay Log-Likelihood: Train vs Test**

We illustrate performance on a regression task using the flight delay dataset, processed by Hensman et al. (2013), with the goal of predicting how long flights are delayed based on eight features. We use RNVP flows as the bijector and evaluate performance by measuring the root mean squared error (RMSE) and NLL. Following Deisenroth & Ng (2015), we train using the first 5M (million) data points and use the following 100,000 as test data points. We picked this split of the dataset not only to illustrate the scalability of our method, but also due to the fact that **the test distribution is known to be slightly different from training, which poses challenges due to non-stationarity**.

To the best of our knowledge, **the state-of-the-art performance is a RMSE of 38.38 and NLL of 6.91** [Lakshminarayanan et al., 2016]. Our hybrid model, which assumes *p(**y**|**x**)* to be a heteroscedastic Gaussian, achieves a slightly worse RMSE of 40.46 but **achieves better NLL of 5.07**. The lower NLL shows the usefulness of the hybrid model on non-stationary problems: the histogram to the left confirms that the test data points indeed have lower density than the training points.

**Summary** Our hybrid model DIGLM combines deep invertible features and GLMs so that p(x) and p(y|x) can be computed exactly in a single feedforward pass. **DIGLM's predictive performance is competitive with pure discriminative models p(y|x), while the generative model p(x) can be useful for better uncertainty estimation and generating samples from the model.** Future work will explore Bayesian GLM and applications to semi-supervised learning, active learning and domain adaptation.

**Contact**: e.nalisnick@eng.cam.ac.uk,   {amatsukawa, ywteh, dilang, balajiln}@google.com