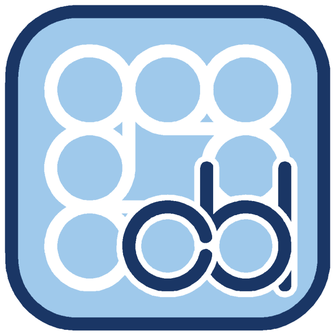
Do Deep Generative Models Know What They Don't Know?

Eric Nalisnick

CamAIML

26.3.19



Computational and Biological Learning
University of Cambridge

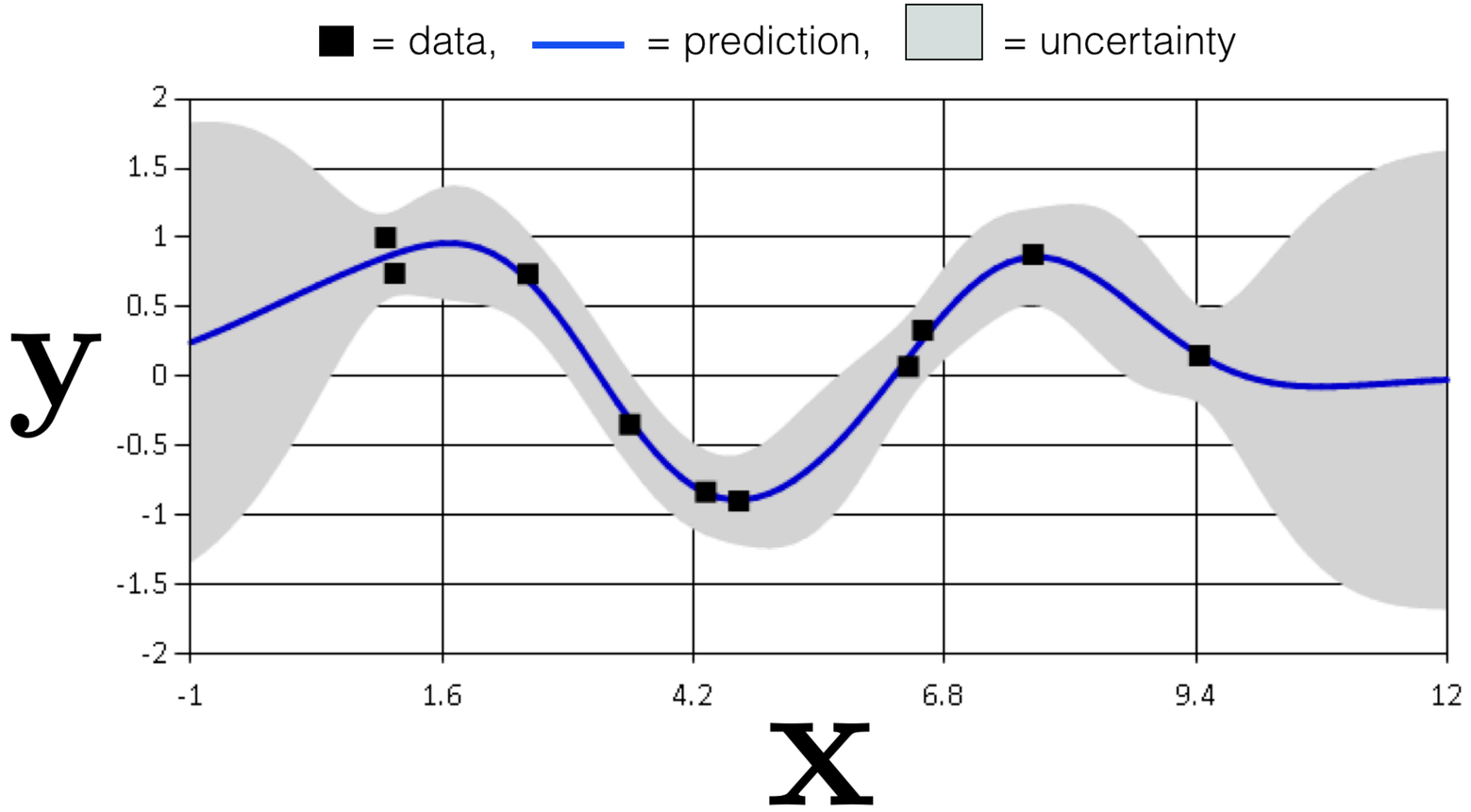
DeepMind

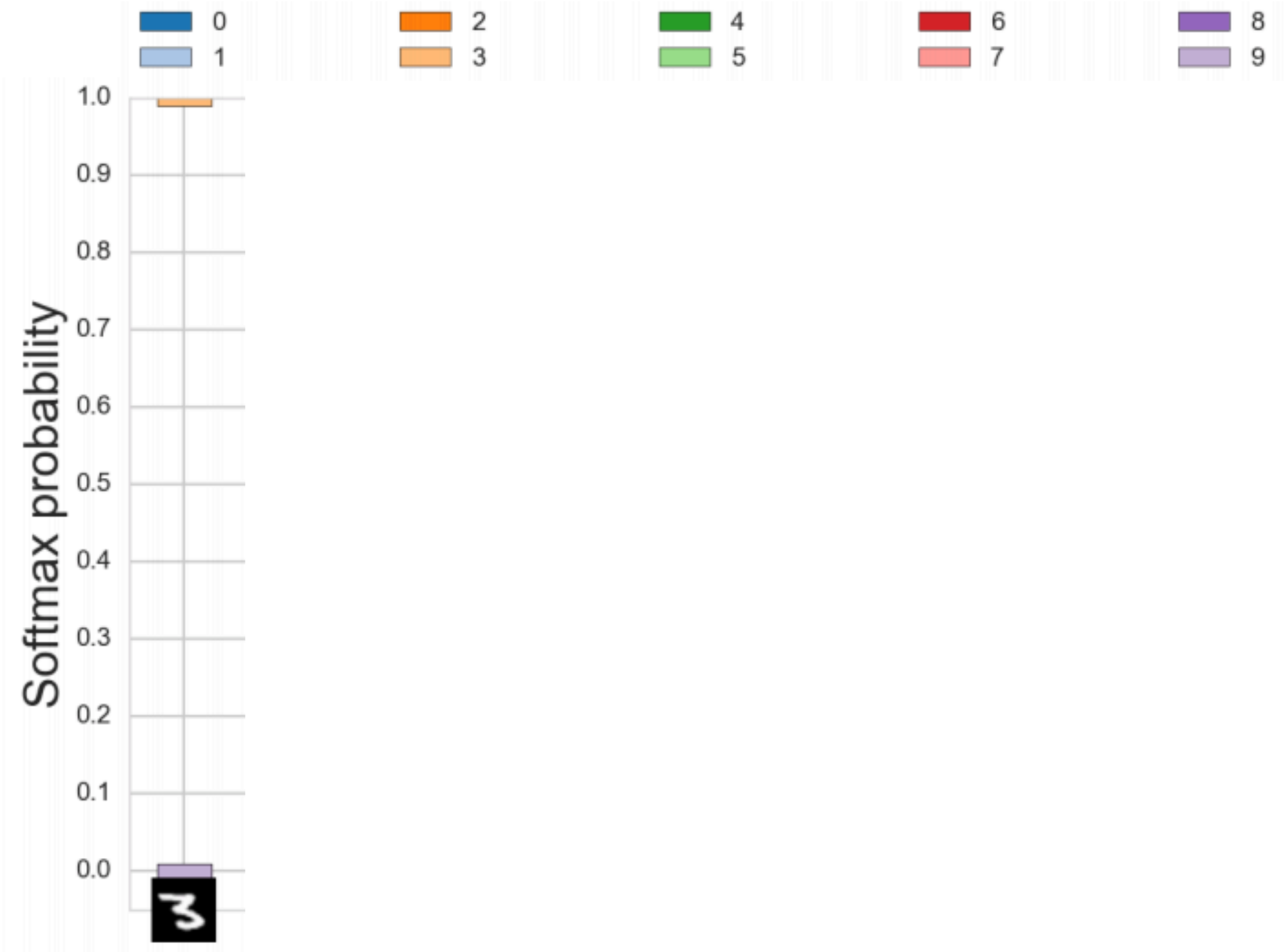
MOTIVATION

Protecting Predictive Models

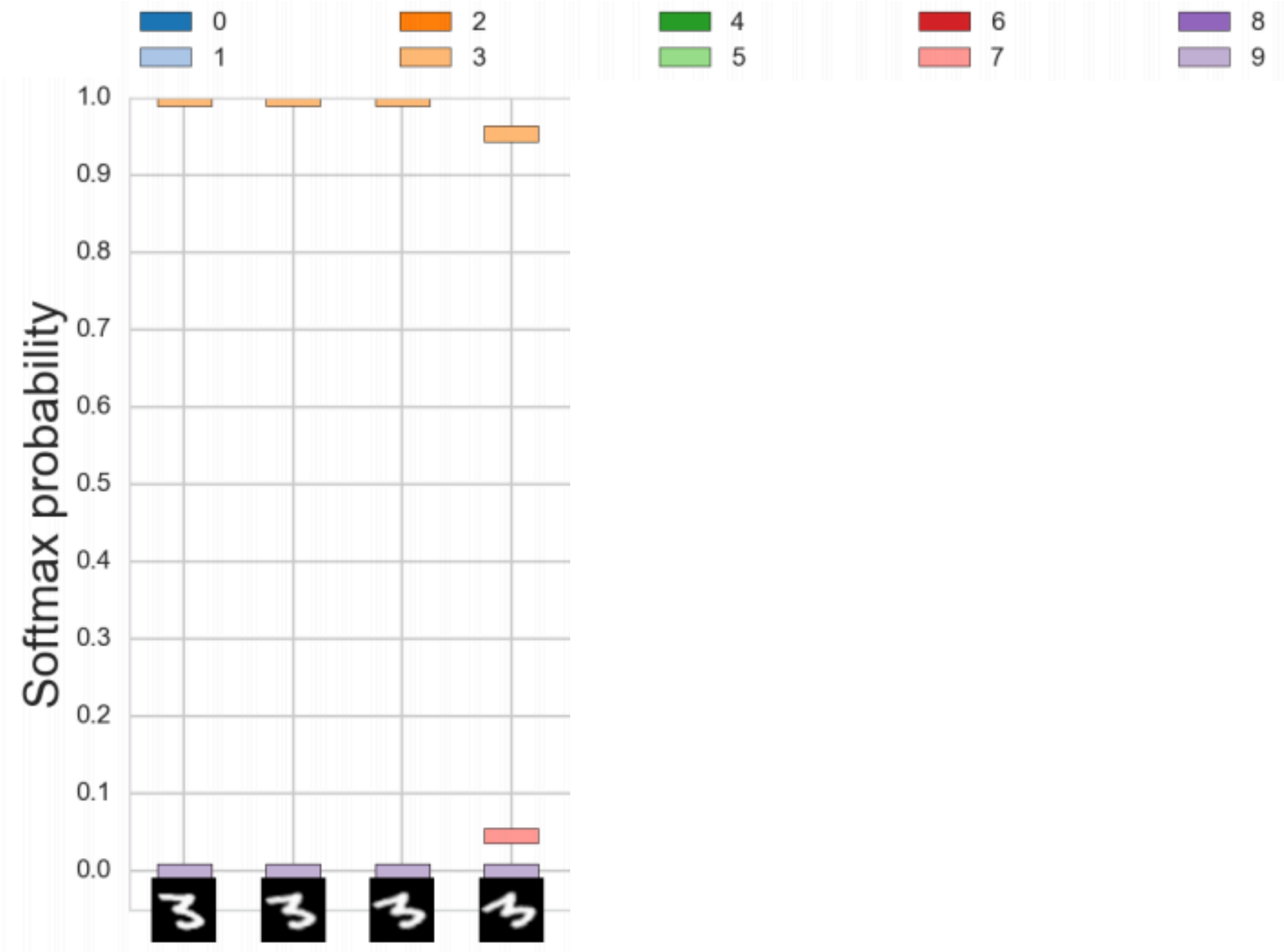
$$p(\mathbf{y}|\mathbf{X};\boldsymbol{\theta})$$

Labels **Features** **Parameters**

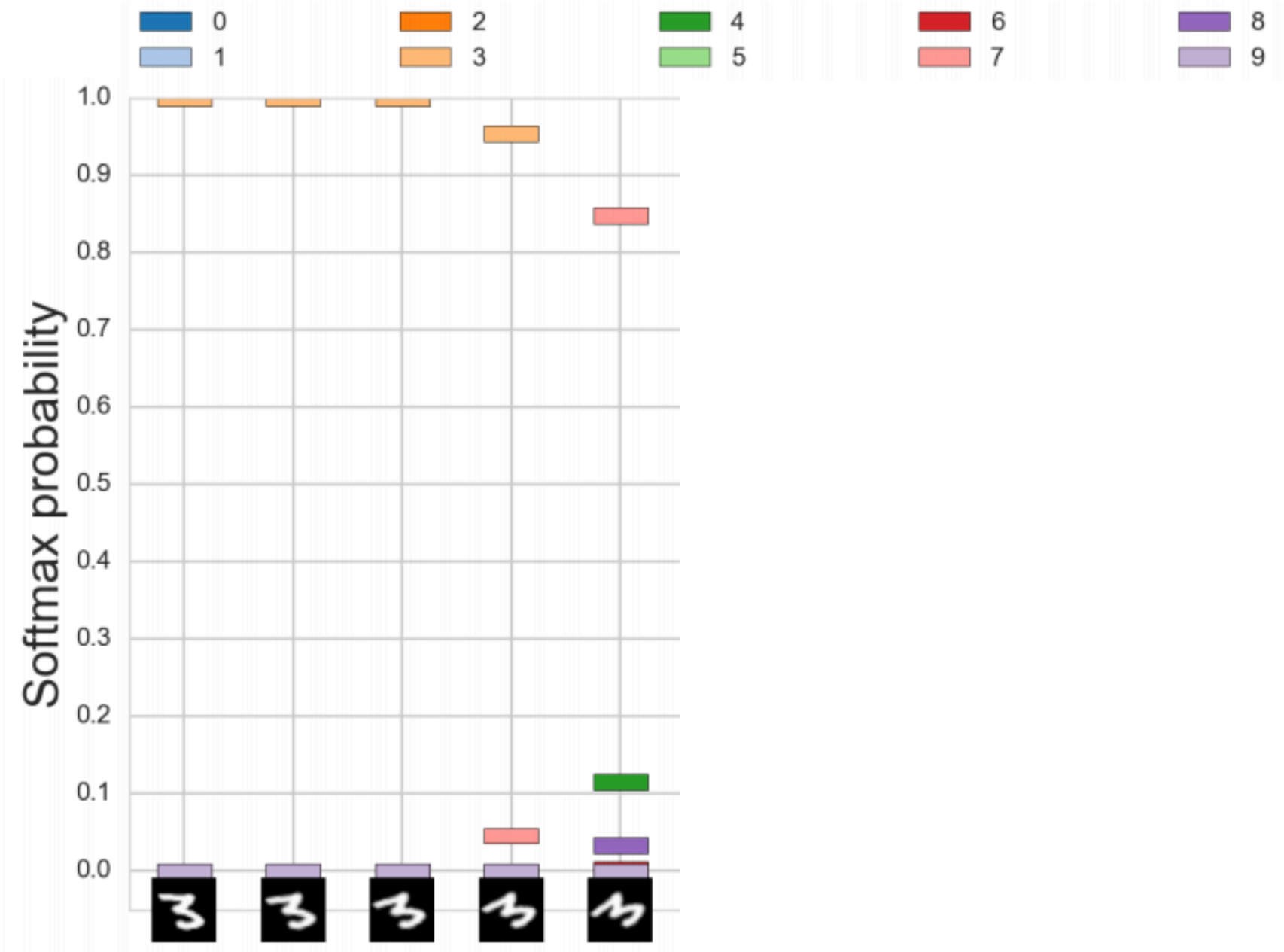




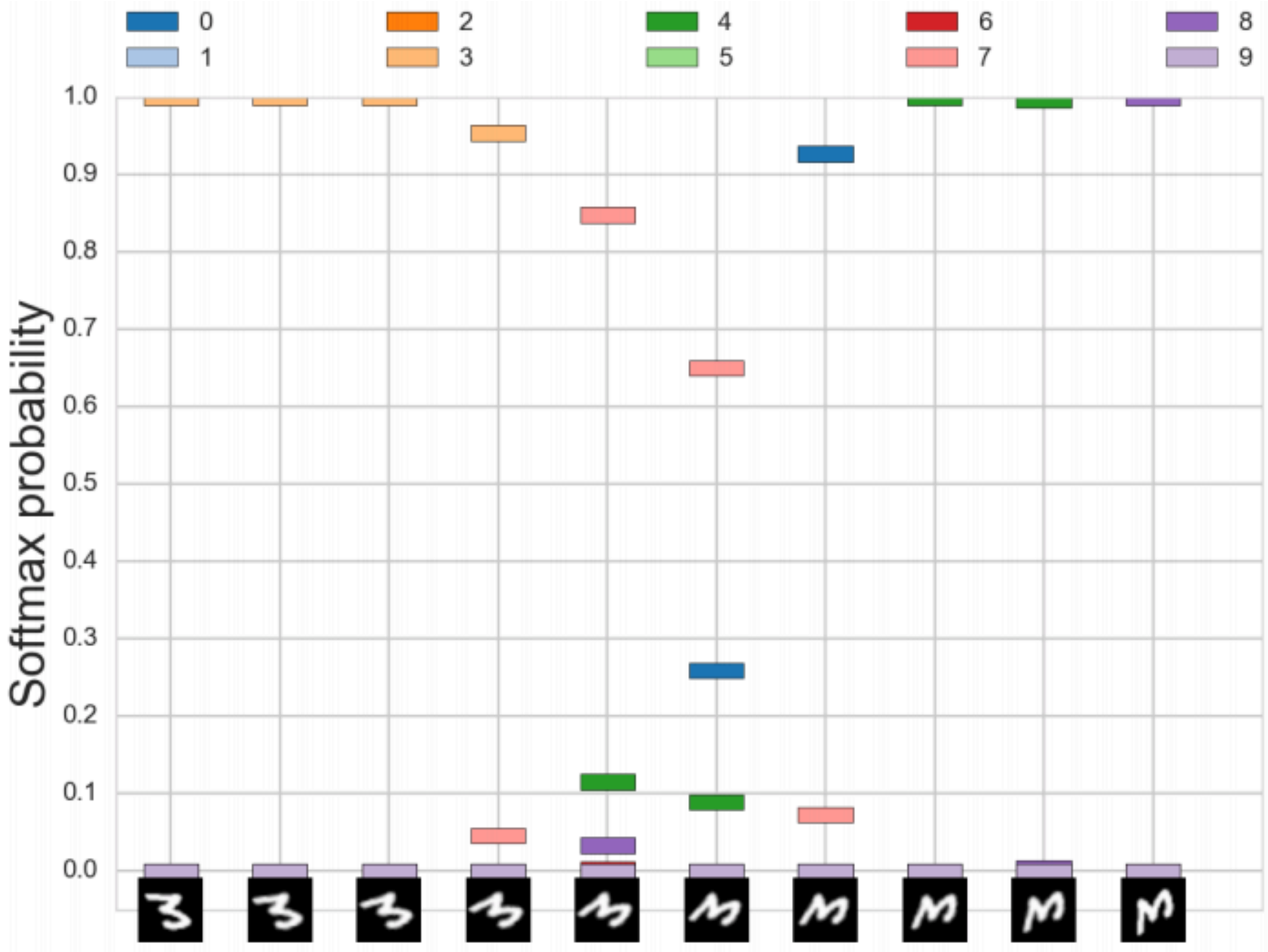
[Louizos & Welling, ICML 2017]



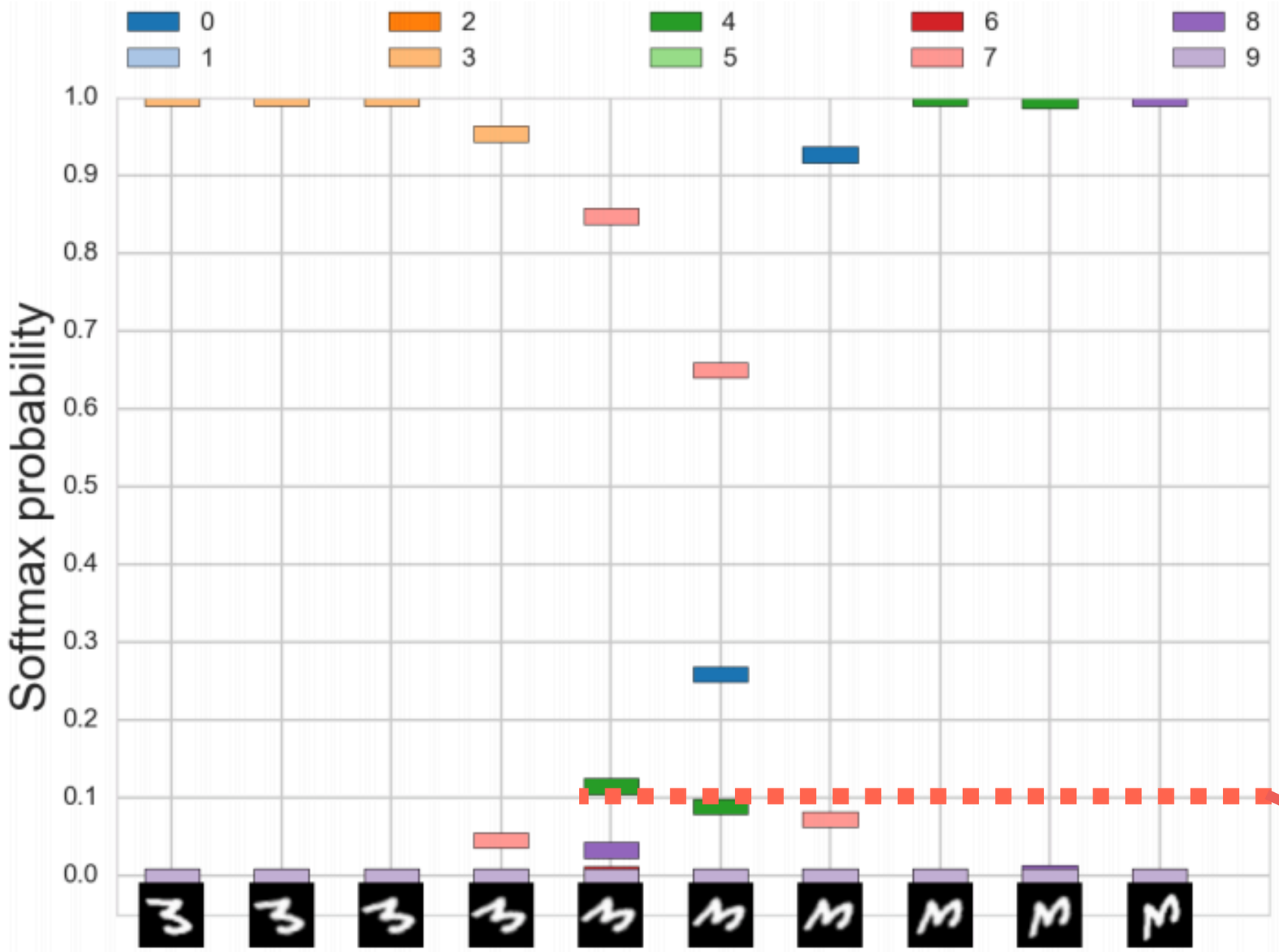
[Louizos & Welling, ICML 2017]



[Louizos & Welling, ICML 2017]



[Louizos & Welling, ICML 2017]

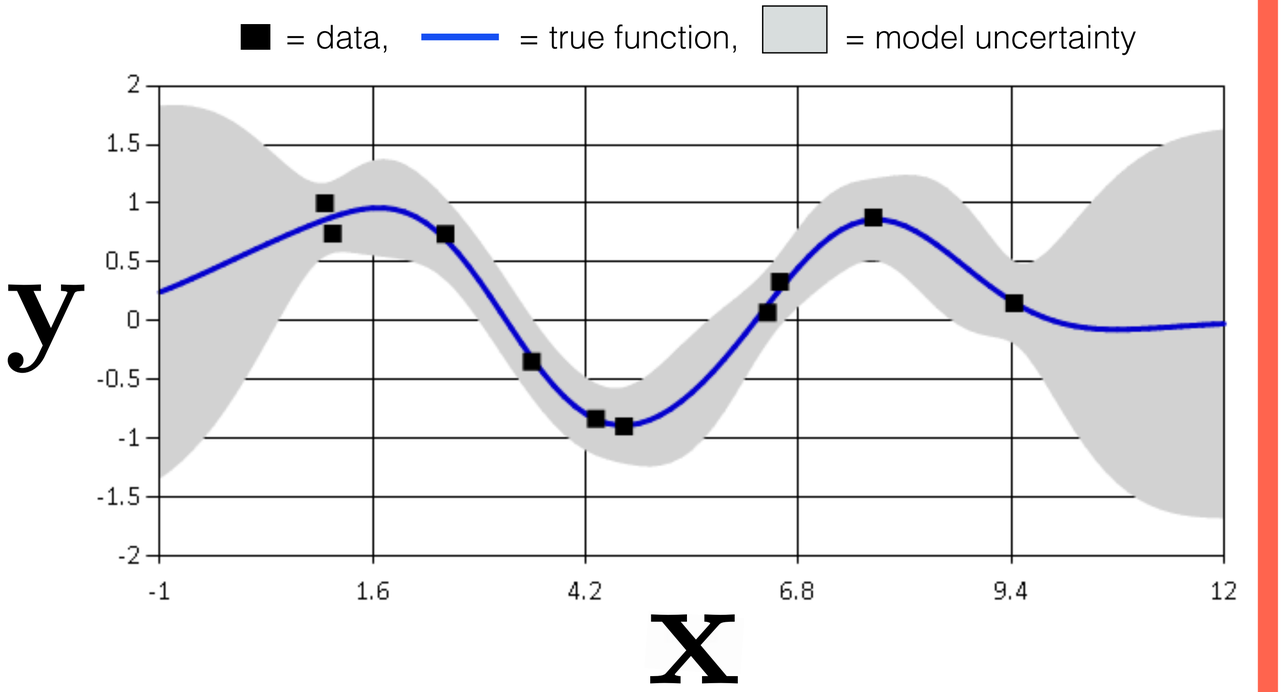


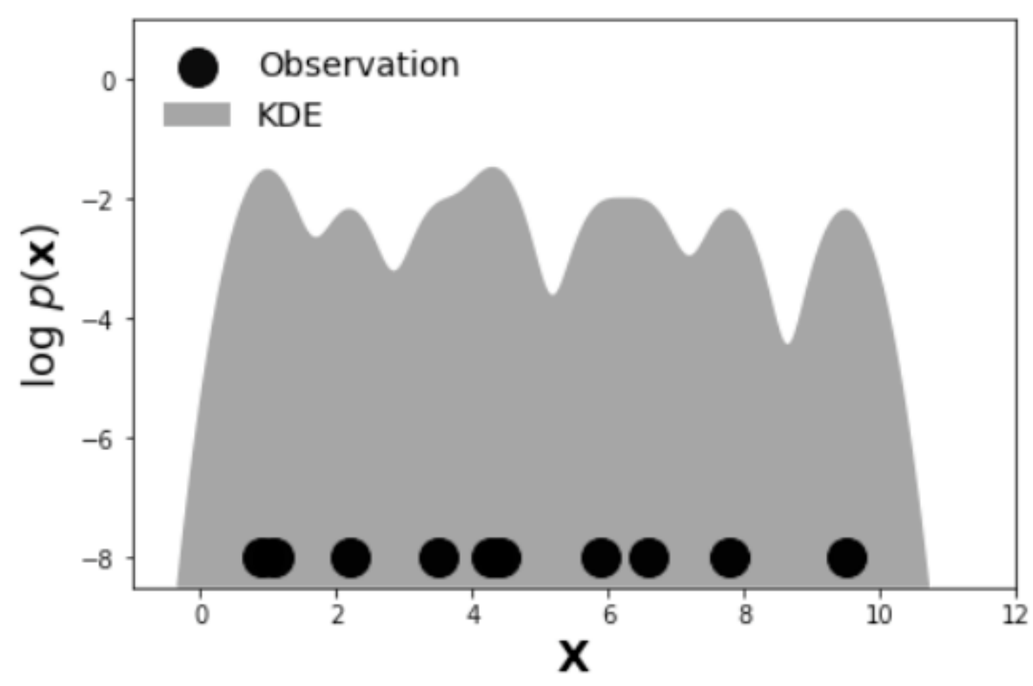
Maximum Uncertainty

[Louizos & Welling, ICML 2017]

$$p(\mathbf{y}, \mathbf{X}) = p(\mathbf{y}|\mathbf{X}; \boldsymbol{\theta})\ p(\mathbf{X}; \boldsymbol{\phi})$$

Predictive Model **Generative Model**





Deep Generative Models

- **Autoregressive Models** (e.g. WaveNet)

  [Larochelle & Murray, AISTATS 2011; van den Oord, ICLM 2016; van den Oord, NeurIPS 2016]

- **Variational Autoencoders** (VAEs)

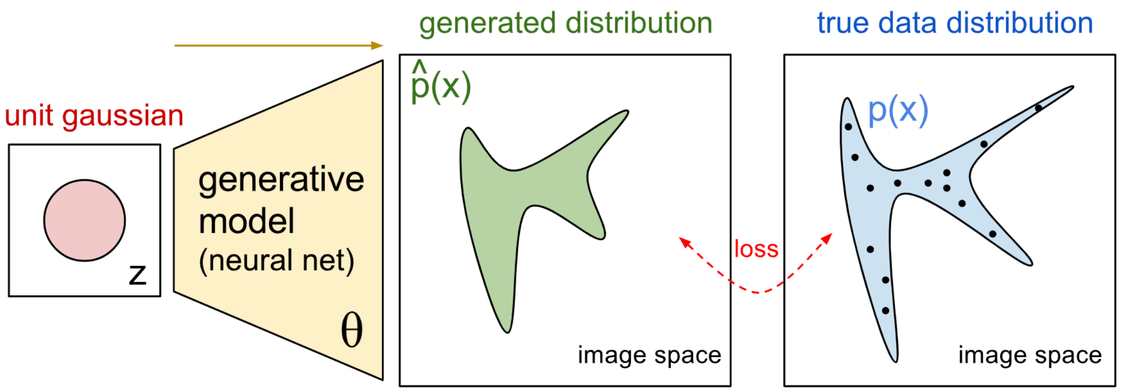
  [Kingma & Welling, ICLR 2014; Rezende et al., ICML 2014]

- **Generative Adversarial Networks** (GANs)

  [Goodfellow et al., NeurIPS 2014]

- **Normalizing Flows**

  [Tabak & Turner, 2013; Rezende & Mohamed, ICML 2015; Dinh et al., ICLR 2017]



https://blog.openai.com/generative-models/

[Kingma & Dhariwal, NeurIPS 2018]

Inputs Unlike Training Data

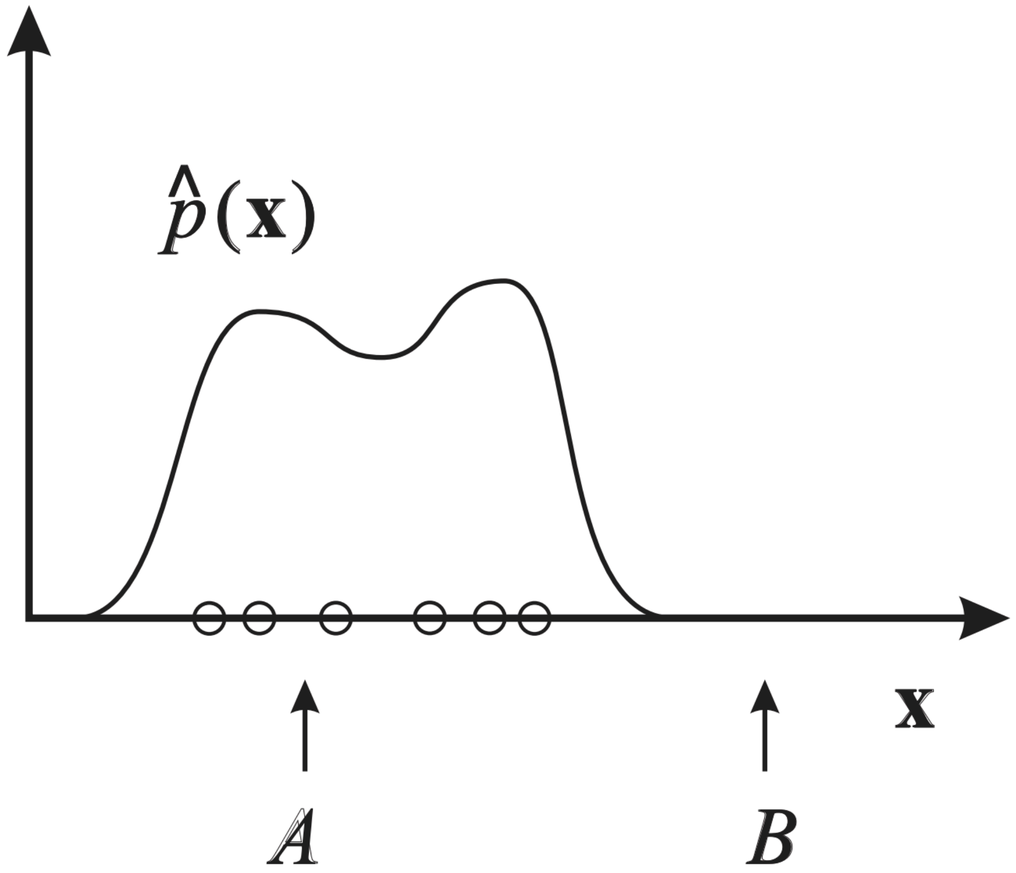
$$\text{if } p(\boldsymbol{x}^*; \boldsymbol{\phi}) < \tau,$$

$$\text{then } \texttt{reject } \boldsymbol{x}^*$$

Use p(**X**) model to reject inputs with density below some threshold [Bishop, 1994].

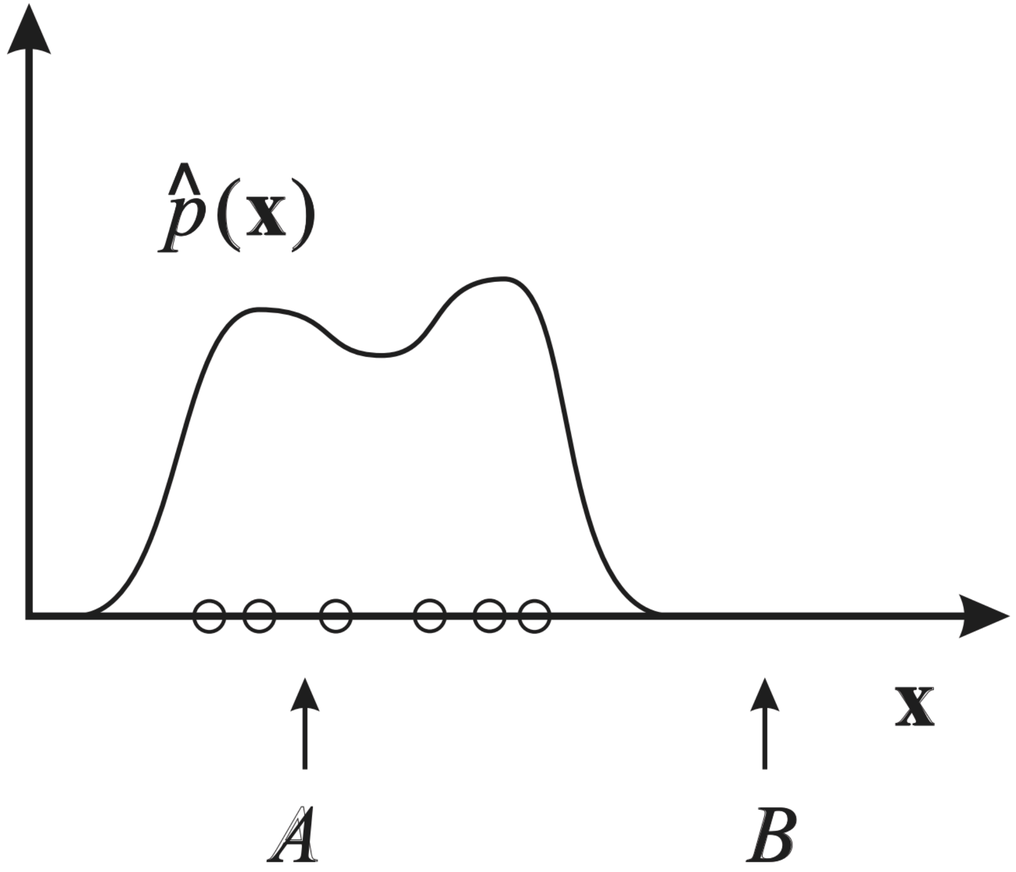
Novelty Detection and Neural Network Validation

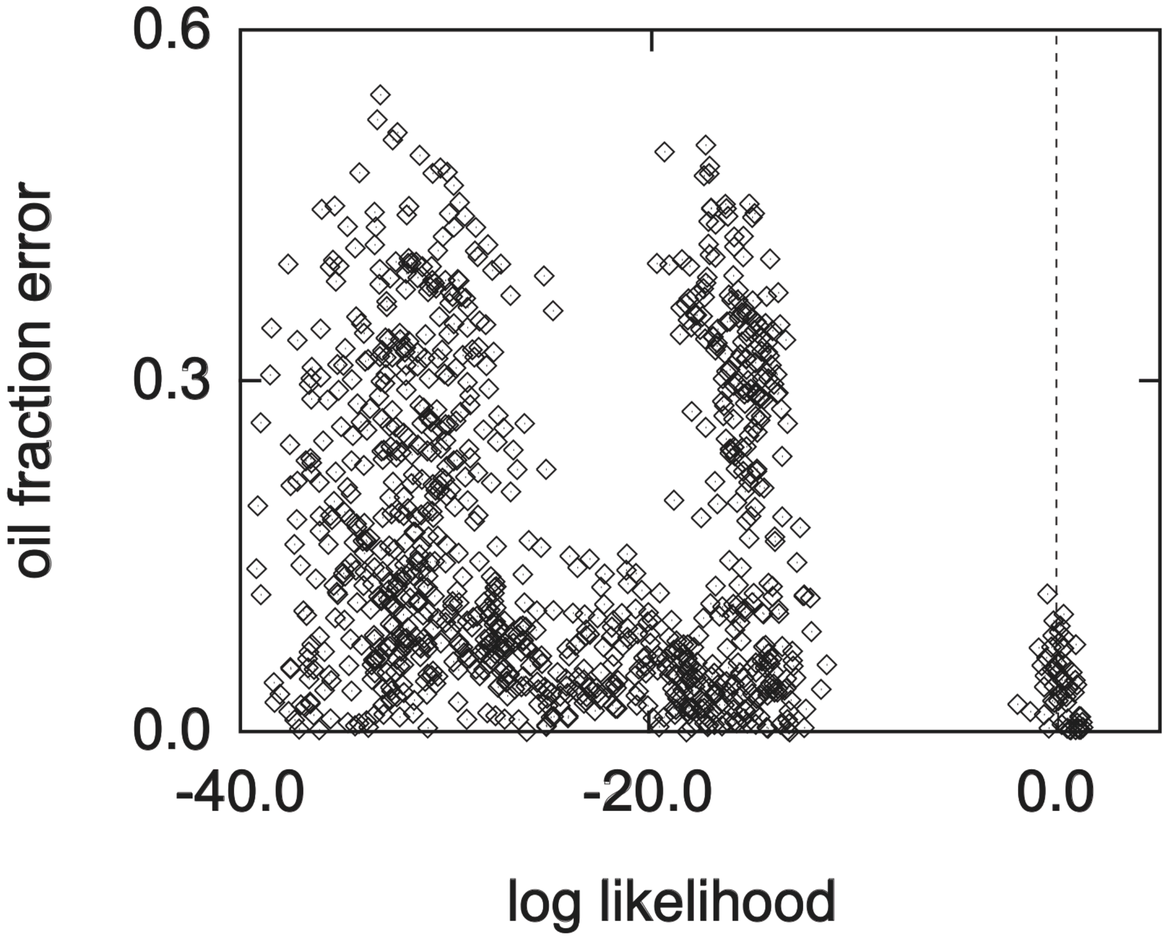
Chris M. Bishop (May 1994)



Novelty Detection and Neural Network Validation

Chris M. Bishop (May 1994)





Panel Discussion

Advances in Approximate Bayesian Inference, Dec 2017

ZOUBIN: [The Bishop (1994) procedure] should be built into the software.

Panel Discussion

Advances in Approximate Bayesian Inference, Dec 2017

ZOUBIN: [The Bishop (1994) procedure] should be built into the software.

MODERATOR: Isn't that hard?

Panel Discussion

Advances in Approximate Bayesian Inference, Dec 2017

ZOUBIN: [The Bishop (1994) procedure] should be built into the software.

MODERATOR: Isn't that hard?

ZOUBIN: If you stick a picture of a chicken into an MNIST classifier, it should tell you it's neither a seven nor a one.

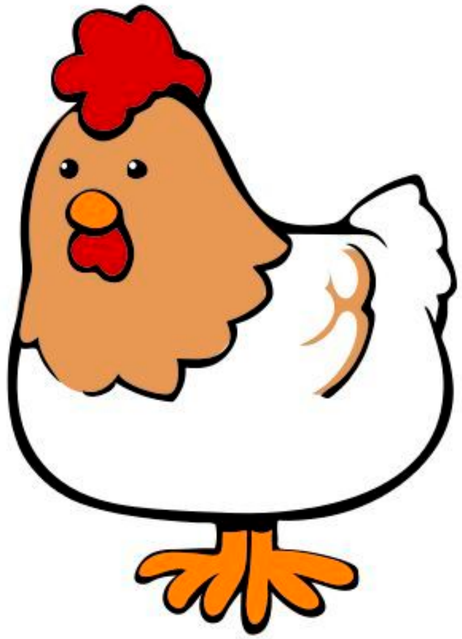
[*AUDIENCE LAUGHS*]

Panel Discussion

Advances in Approximate Bayesian Inference, Dec 2017

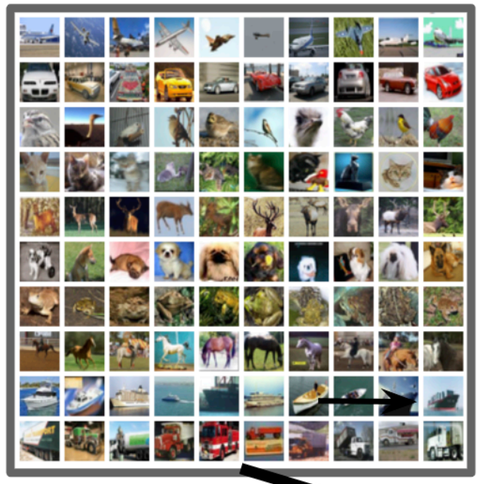
EXPERIMENT

Chicken or Seven?



Chicken or Seven?: Simulating the Scenario

Training: *CIFAR-10*



Testing: *SVHN*

GENERATIVE MODEL

?

$$p(\mathbf{x}_{\text{CIFAR-10}}) > p(\mathbf{x}_{\text{SVHN}})$$

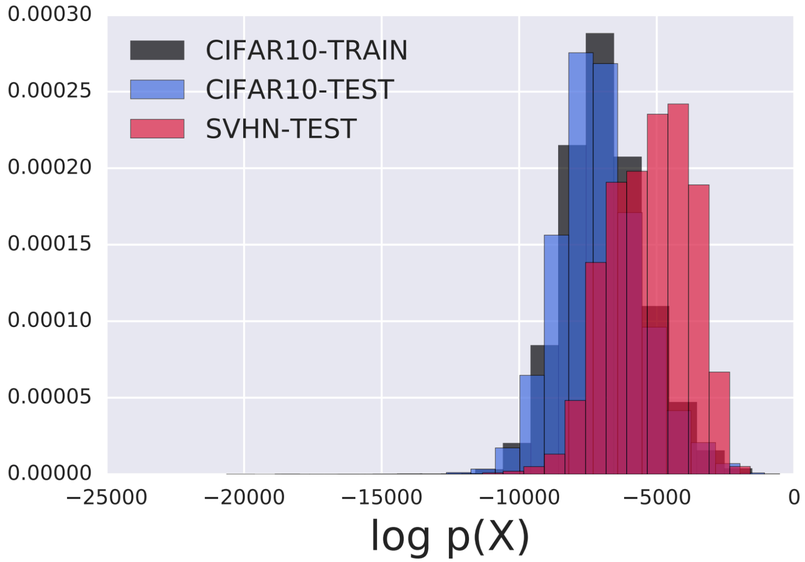
Experiment: CIFAR-10 vs SVHN

Variational Autoencoder

PixelCNN

Glow
(Normalizing Flow)

Experiment: CIFAR-10 vs SVHN

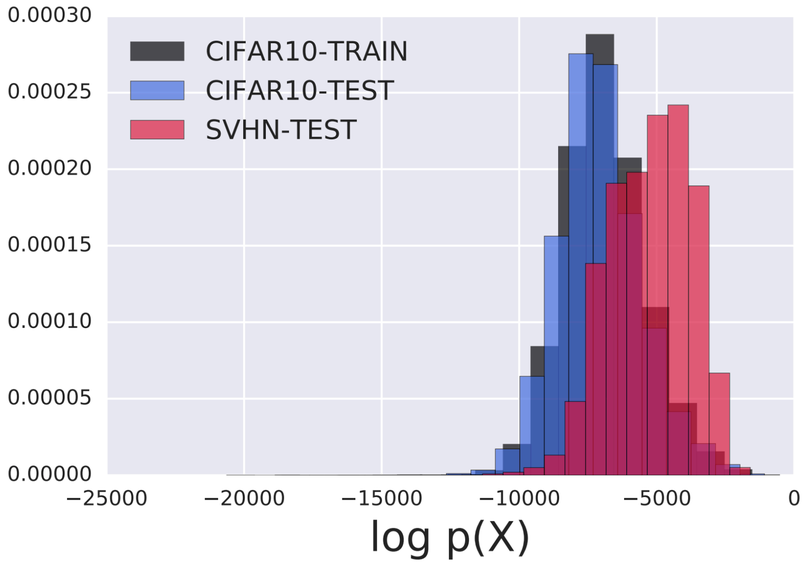


Variational Autoencoder

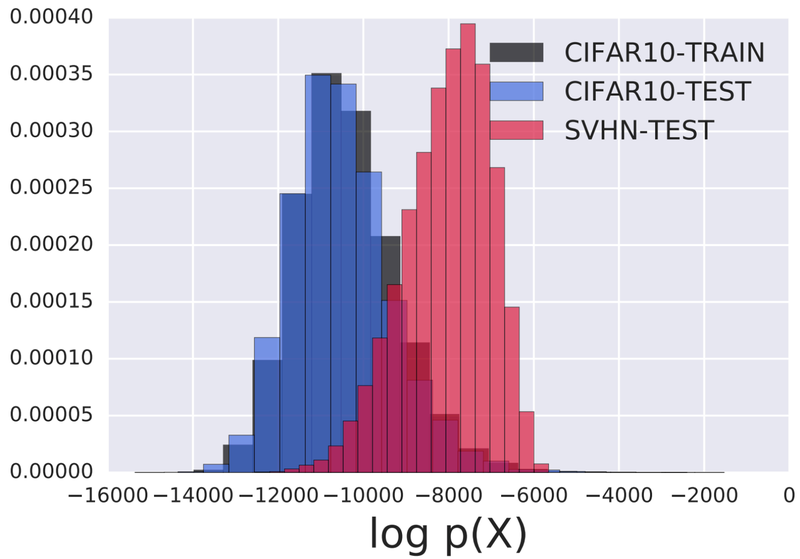
PixelCNN

Glow
(Normalizing Flow)

Experiment: CIFAR-10 vs SVHN



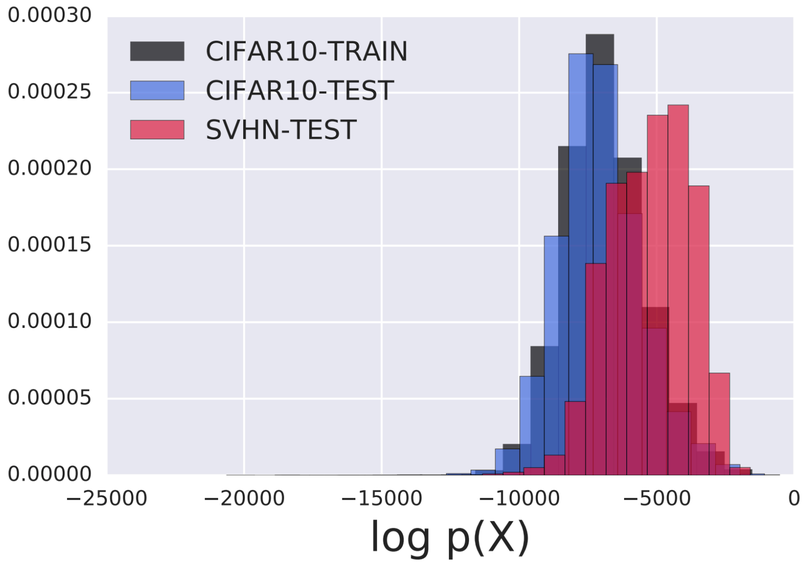
Variational Autoencoder



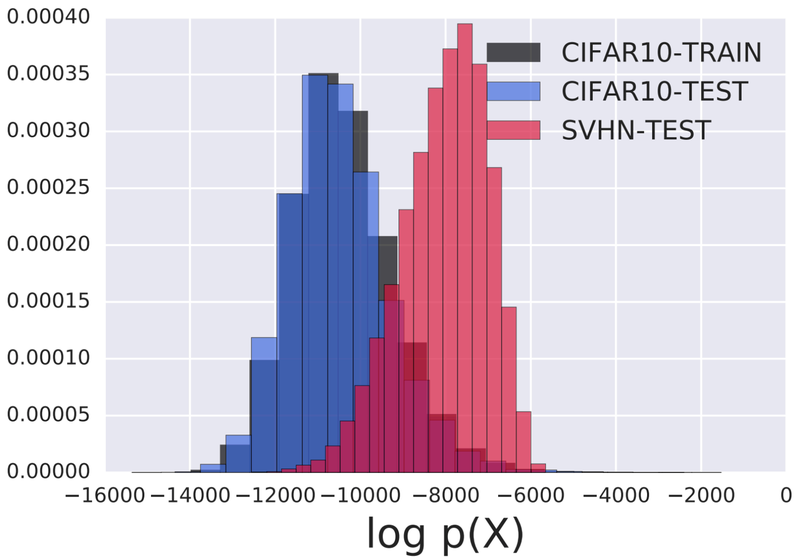
PixelCNN

Glow
(Normalizing Flow)

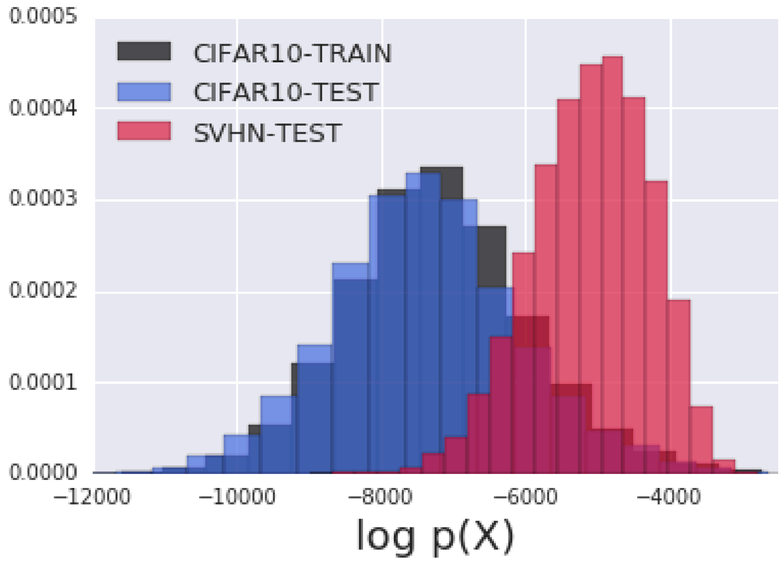
Experiment: CIFAR-10 vs SVHN



Variational Autoencoder

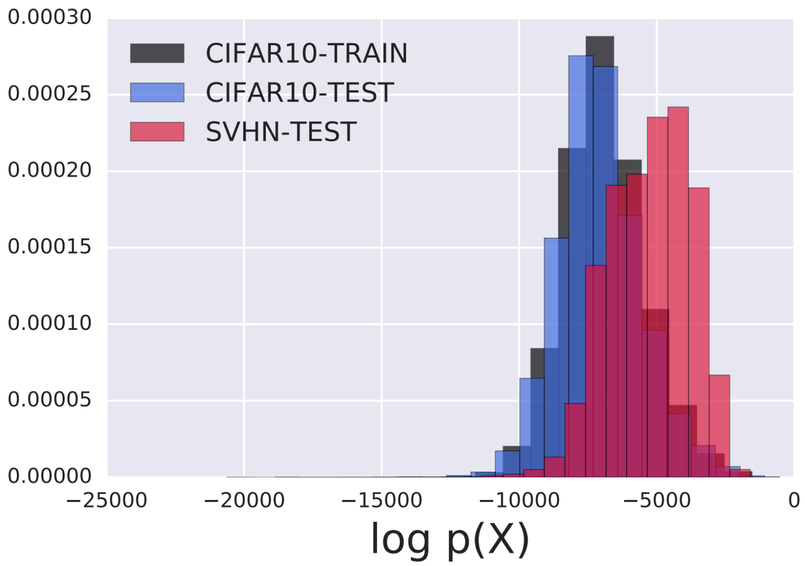


PixelCNN

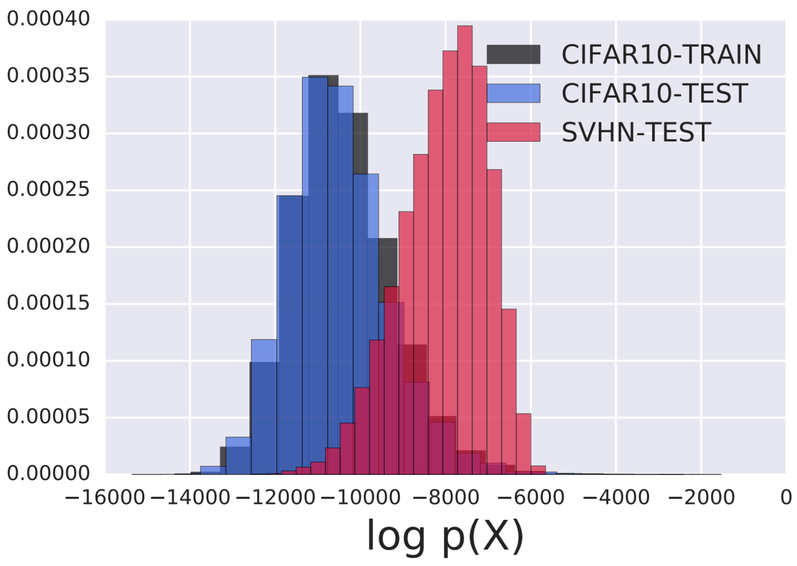


Glow
(Normalizing Flow)

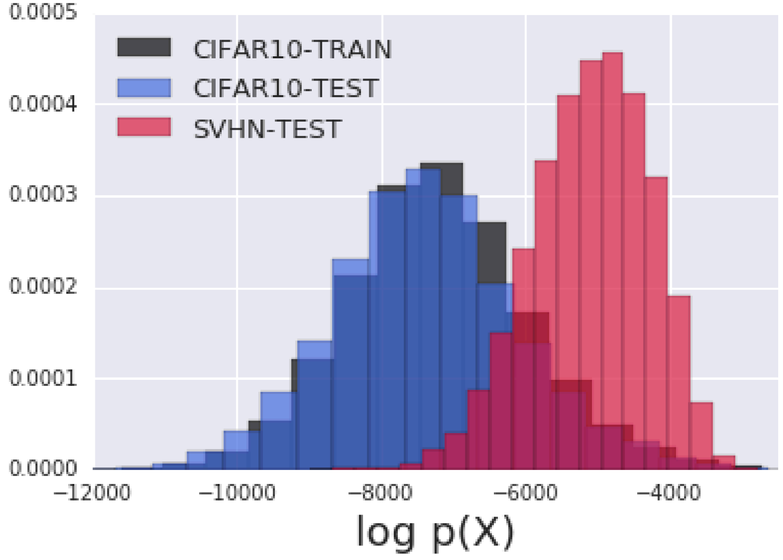
Experiment: CIFAR-10 vs SVHN



Variational Autoencoder



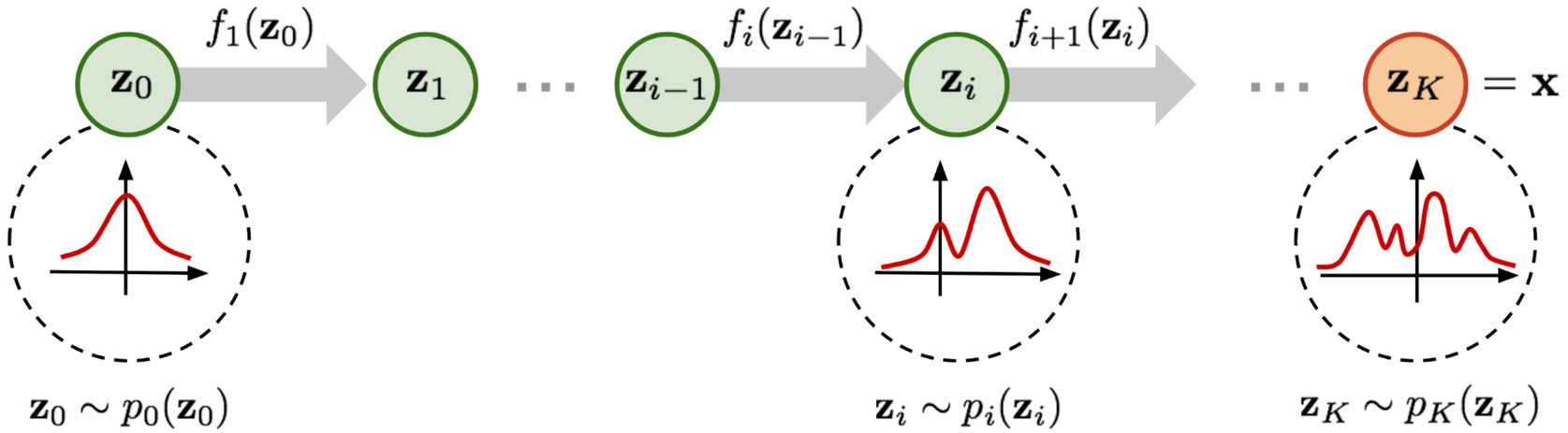
PixelCNN



Glow (Normalizing Flow)

- Exact likelihood calculations
- Additional structure (e.g. invertibility)

Digging Deeper into Flow-Based Models



Deriving Flow-Based Generative Models

Change of Variables Formula (X → Z):

$$p_z(f(X)) \left| \frac{df(X)}{dX} \right| = p(X)$$

Deriving Flow-Based Generative Models

Change of Variables Formula (X → Z):

$$p_z(f(X)) \left| \frac{df(X)}{dX} \right| = p(X)$$

So what's the catch?

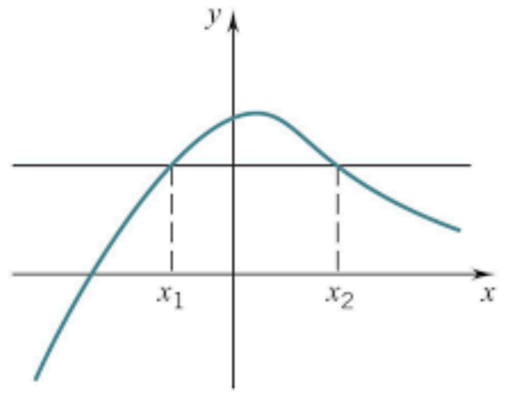
Deriving Flow-Based Generative Models

Change of Variables Formula (X → Z):

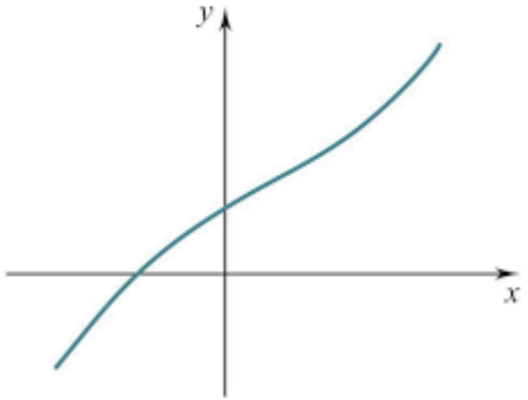
$$p_z(f(X))\left|\frac{df(X)}{dX}\right| = p(X)$$

So what's the catch?

f(**x**) must be a *bijection*



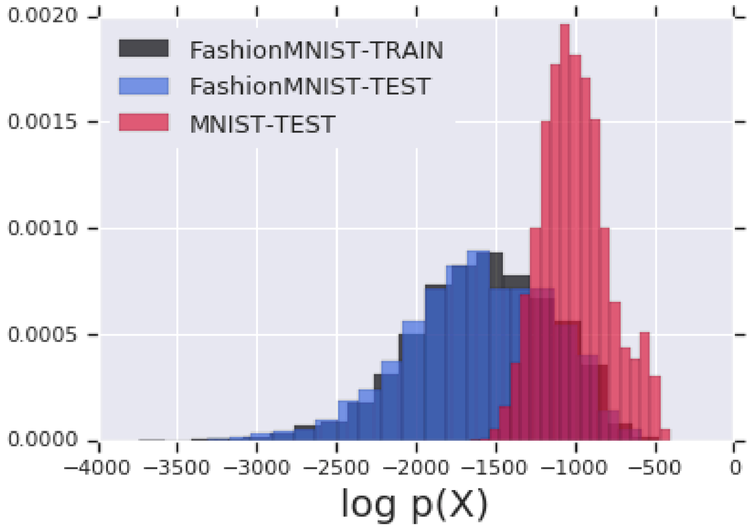
f is not one-to-one: $f(x_1) = f(x_2)$



f is one-to-one:

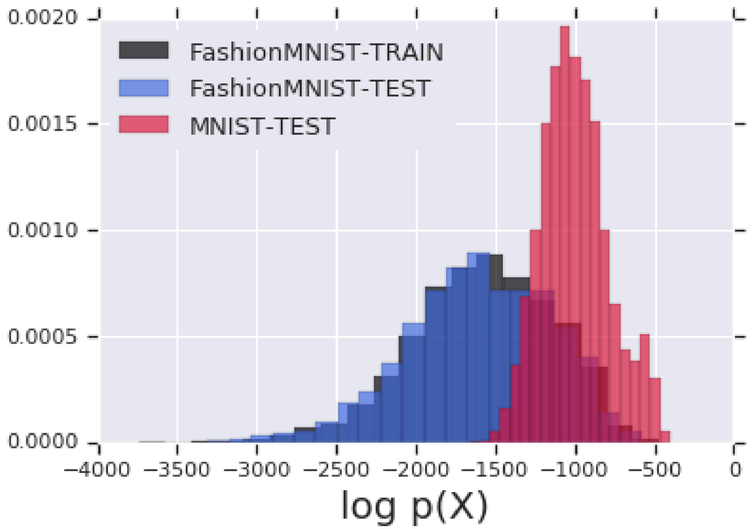
Additional Out-of-Distribution Tests

Additional Out-of-Distribution Tests

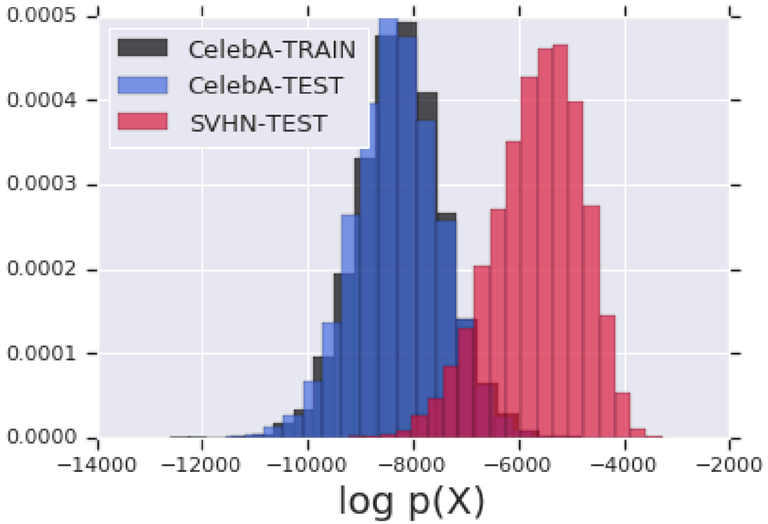


FashionMNIST vs MNIST

Additional Out-of-Distribution Tests

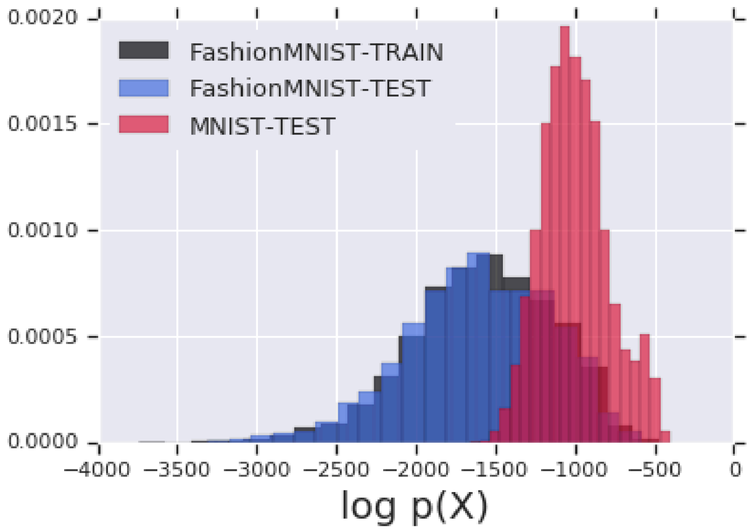


FashionMNIST vs MNIST

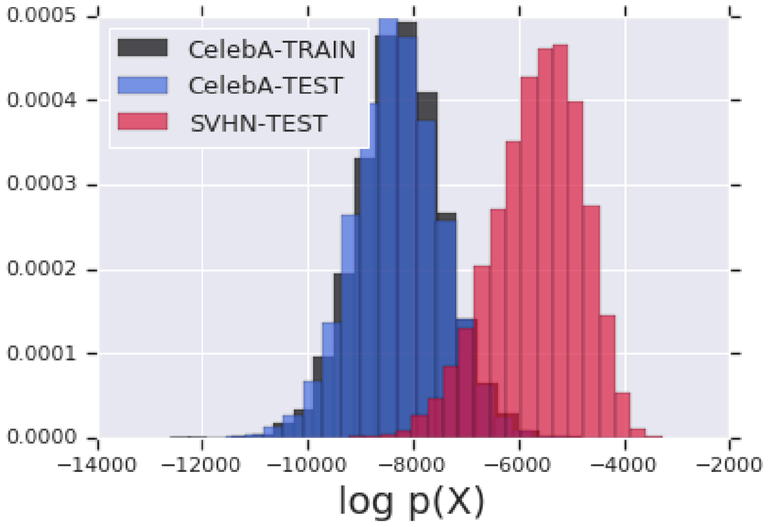


CelebA vs SVHN

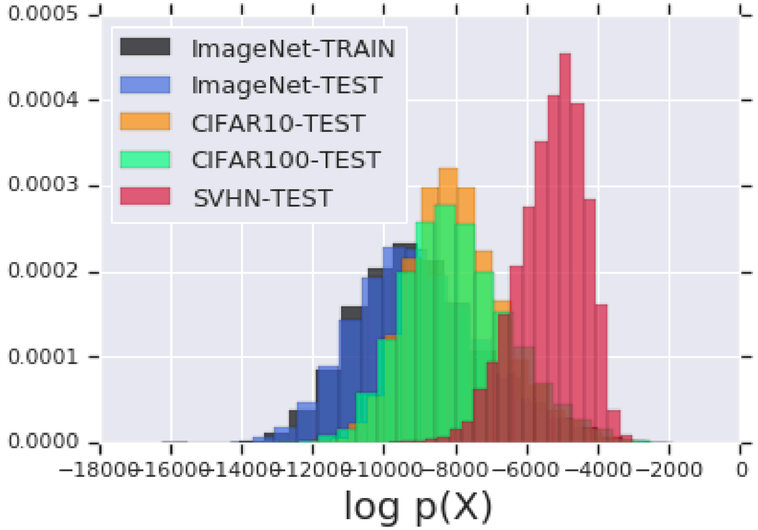
Additional Out-of-Distribution Tests



FashionMNIST vs MNIST

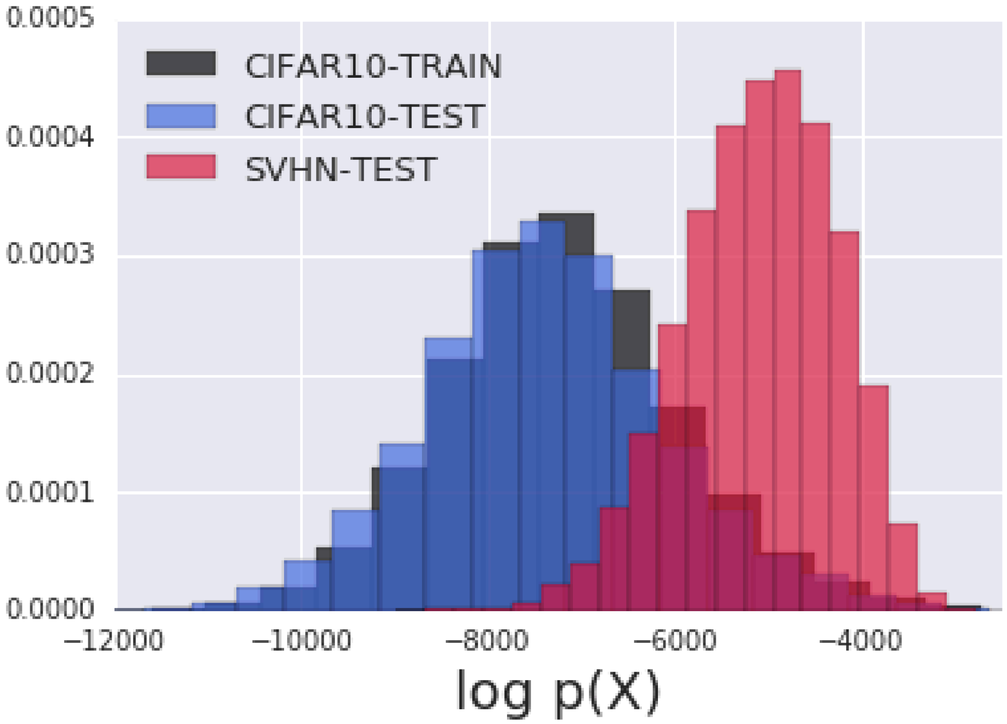


CelebA vs SVHN



ImageNet vs CIFAR-10 vs SVHN

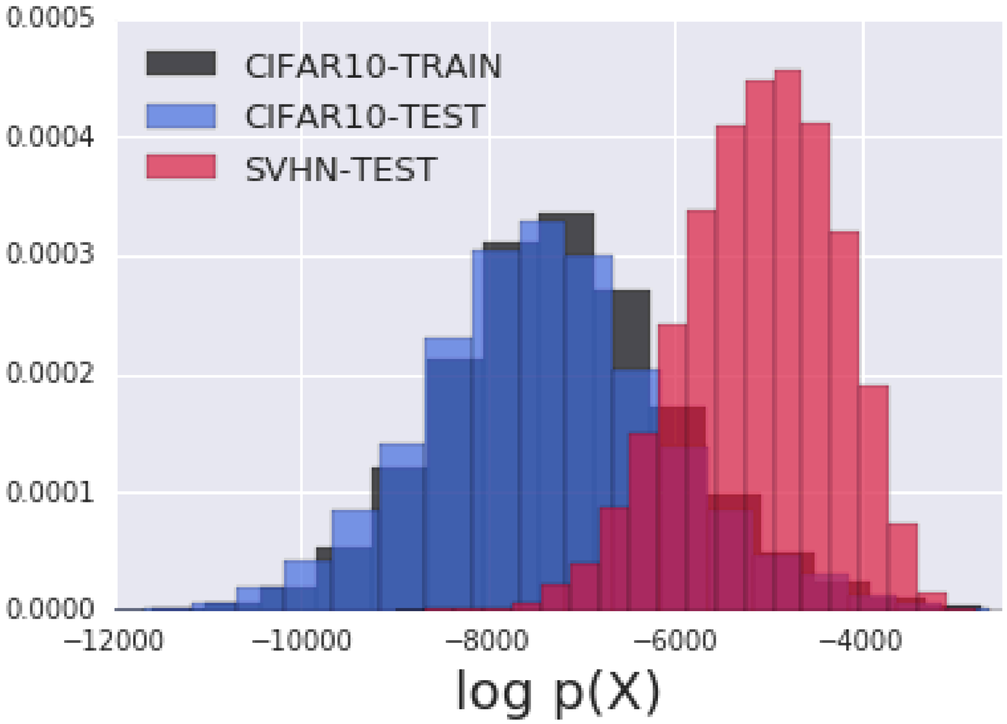
The Phenomenon is not Symmetric



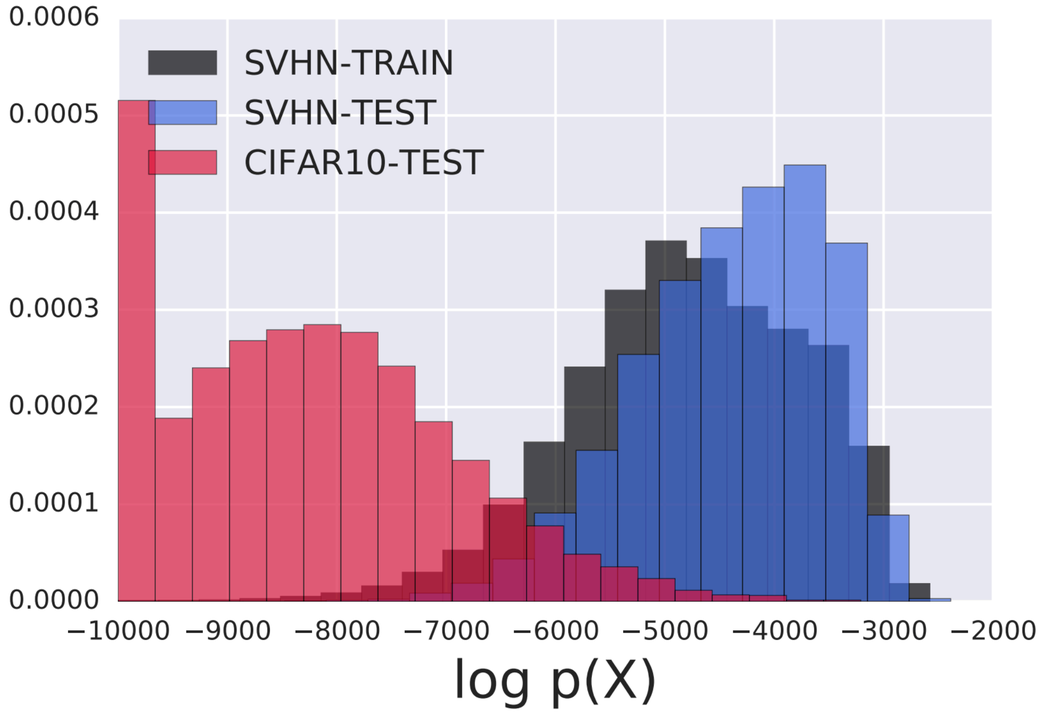
CIFAR-10 vs SVHN

SVHN vs CIFAR-10

The Phenomenon is not Symmetric

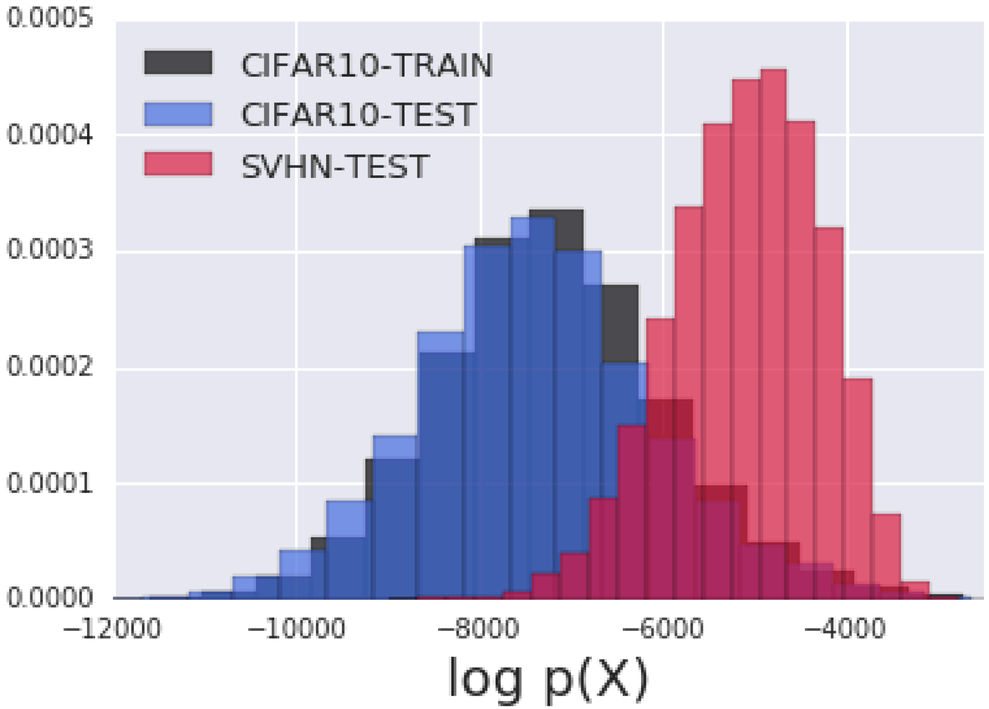


CIFAR-10 vs SVHN



SVHN vs CIFAR-10

Ensembles Do Not Help



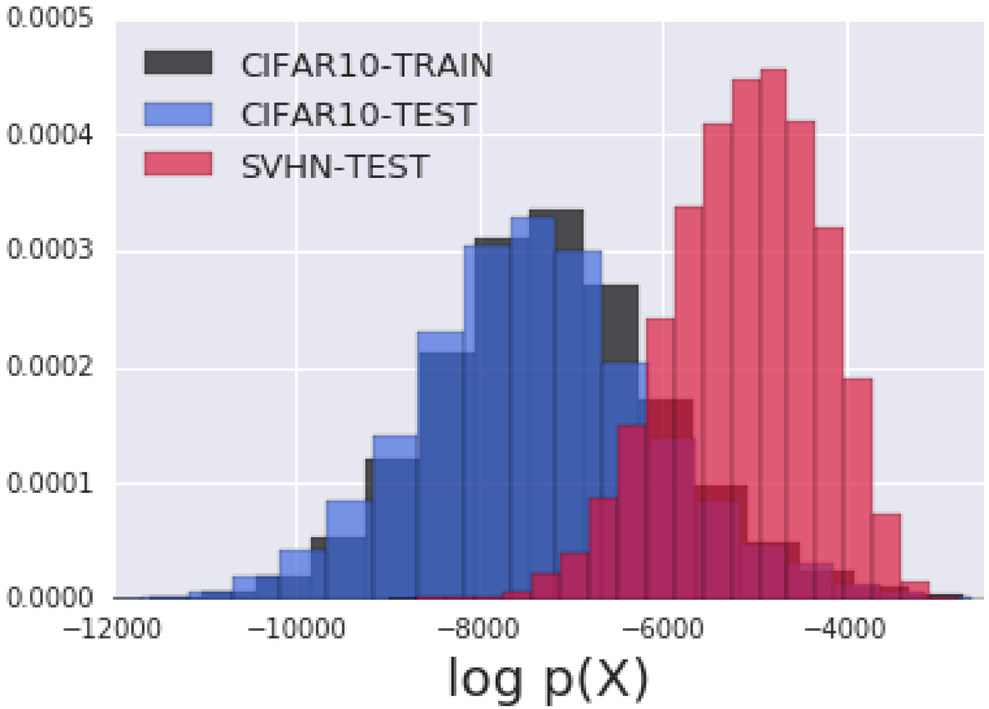
CIFAR-10 vs SVHN

1 Glow

CIFAR-10 vs SVHN

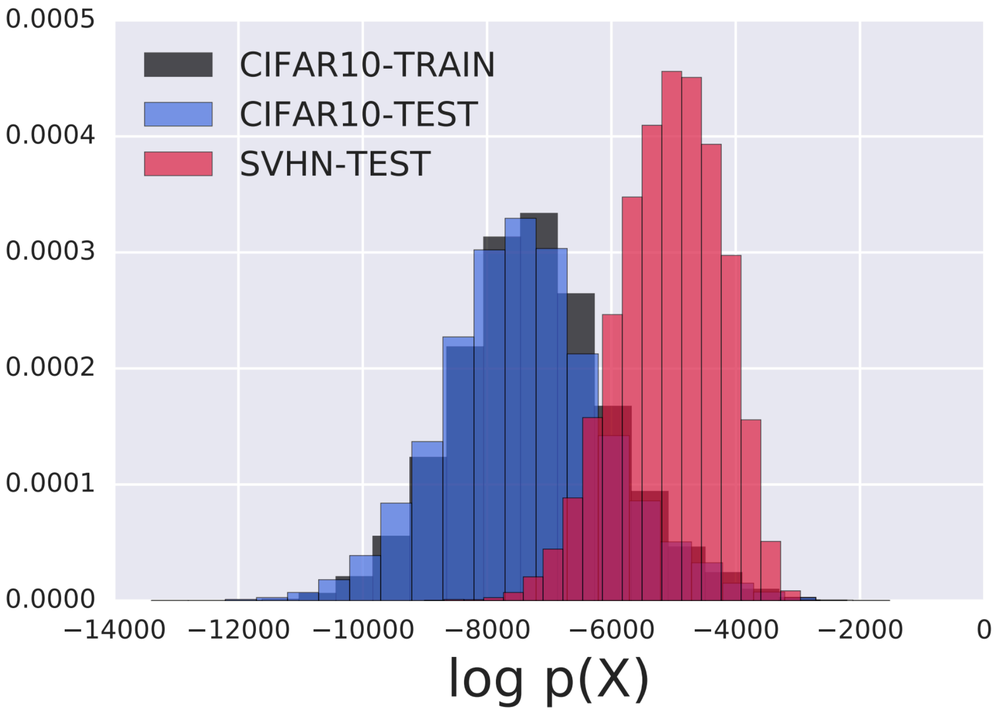
Ensemble of 10 Glows

Ensembles Do Not Help



CIFAR-10 vs SVHN

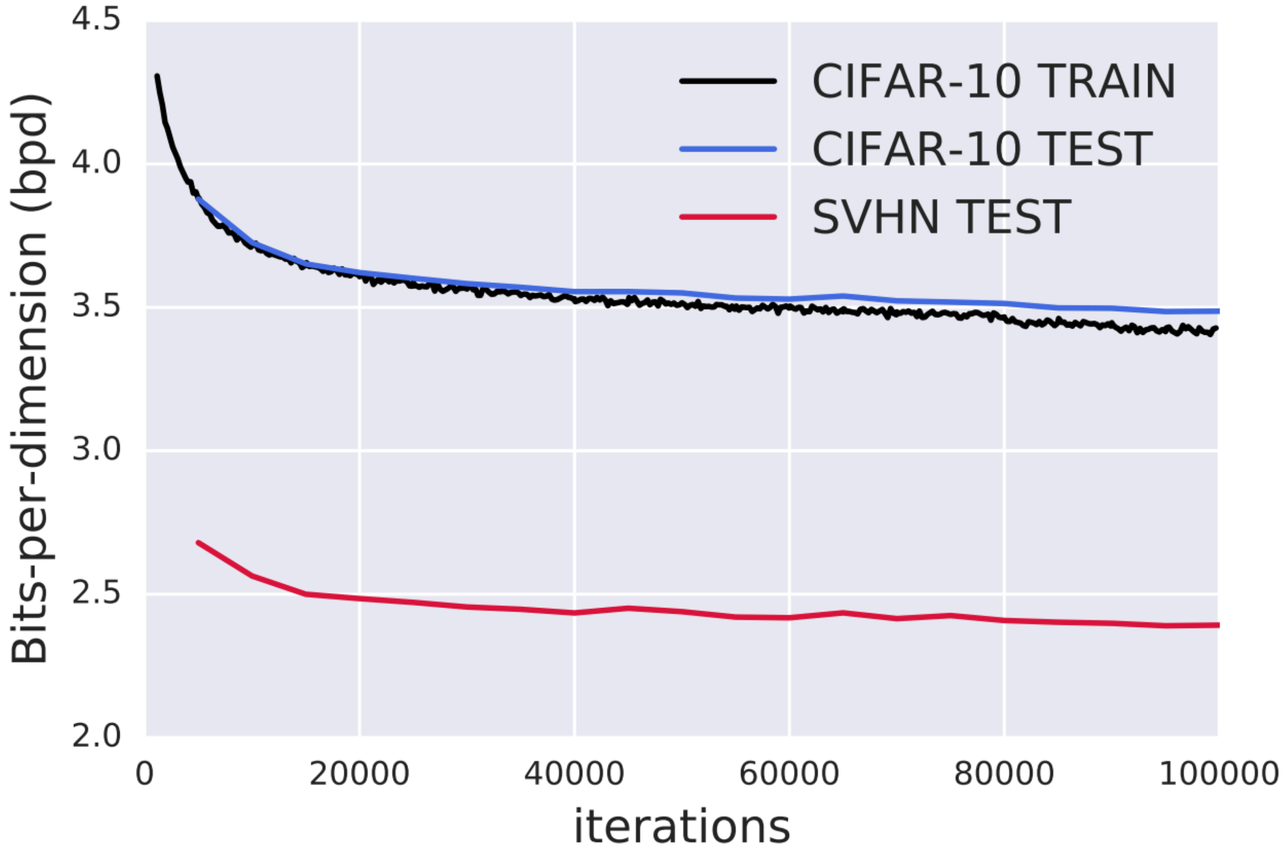
1 Glow



CIFAR-10 vs SVHN

Ensemble of 10 Glows

Early-Stopping Does Not Help



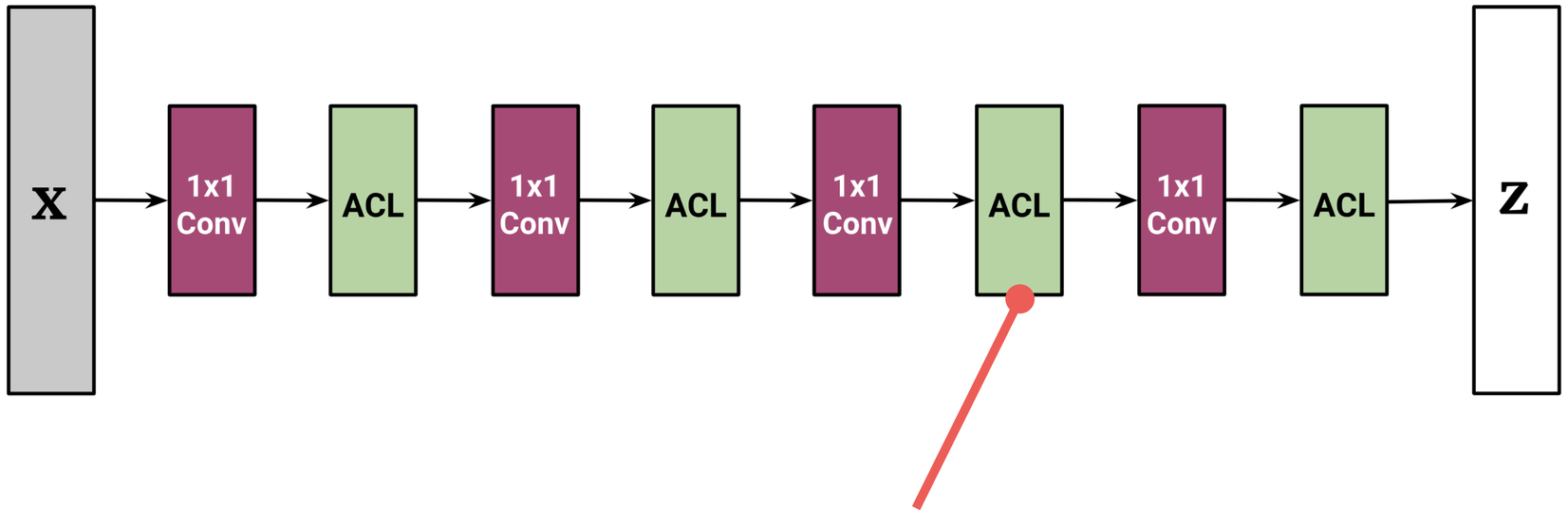
During Optimization

RESULT

Analytical Investigation

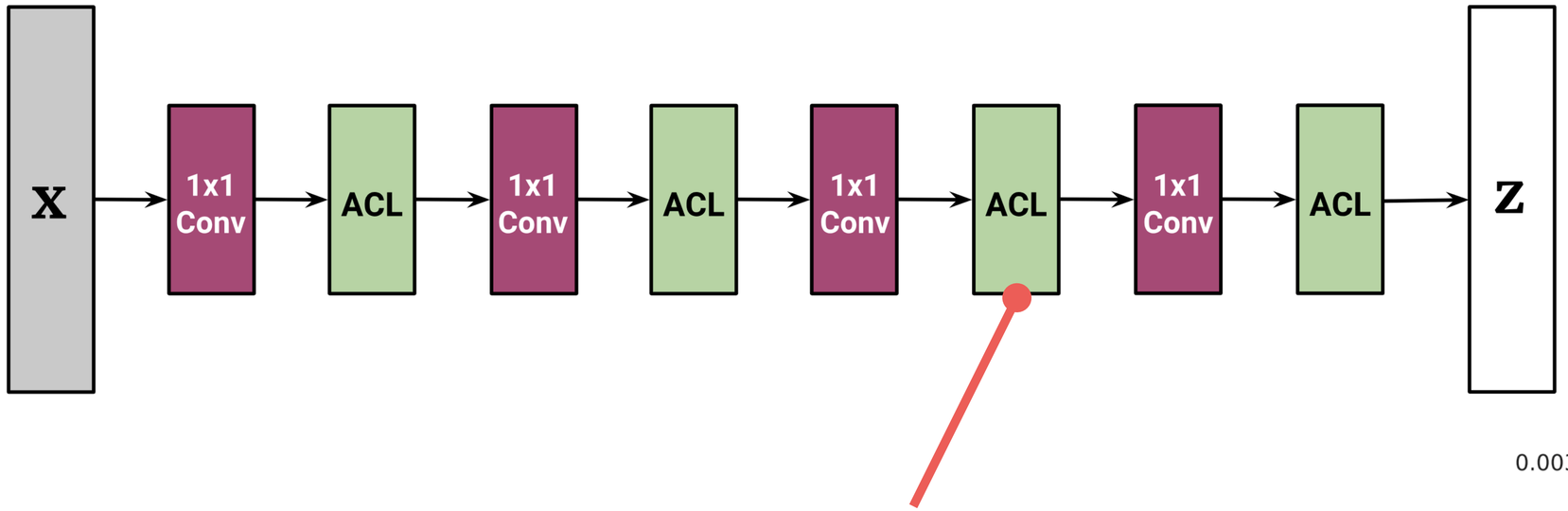
Constant-Volume Flows

We define a sub-class we term *constant-volume* (w.r.t. input) flows.



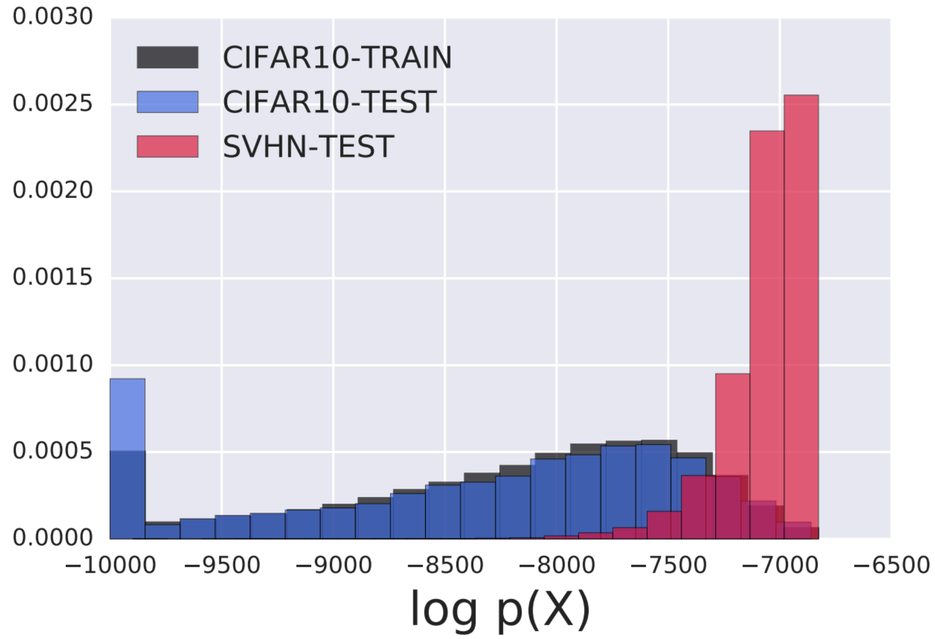
Constant-Volume Flows

We define a sub-class we term *constant-volume* (w.r.t. input) flows.



Use only translation operations.

CIFAR-10 vs SVHN



Constant-Volume Flow Analysis

Mathematical characterization:

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

Non-Training Distribution

Training Distribution

Constant-Volume Flow Analysis

Mathematical characterization:

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

Non-Training Distribution

Training Distribution

$$\approx \frac{1}{2}\operatorname{Tr}\left\{\left[\nabla^2_{\boldsymbol{x}_0}\log p_z(f(\boldsymbol{x}_0;\boldsymbol{\phi})) + \nabla^2_{\boldsymbol{x}_0}\log\left|\frac{\partial \boldsymbol{f}_{\boldsymbol{\phi}}}{\partial \boldsymbol{x}_0}\right|\right](\boldsymbol{\Sigma}_q - \boldsymbol{\Sigma}_{p^*})\right\}$$

Second Moment of Training Distribution

Second Moment of Non-Training Distribution

Change-of-Variable Terms

Constant-Volume Flow Analysis

Mathematical characterization:

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

Non-Training Distribution

Training Distribution

$$\approx \frac{1}{2}\mathrm{Tr}\left\{\left[\nabla^2_{\boldsymbol{x}_0}\log p_z(f(\boldsymbol{x}_0;\boldsymbol{\phi})) + \nabla^2_{\boldsymbol{x}_0}\log\left|\frac{\partial \boldsymbol{f}_{\boldsymbol{\phi}}}{\partial \boldsymbol{x}_0}\right|\right](\boldsymbol{\Sigma}_q - \boldsymbol{\Sigma}_{p^*})\right\}$$

Second Moment of Training Distribution

Second Moment of Non-Training Distribution

Change-of-Variable Terms

Constant-Volume Flow Analysis

Plugging in the CV-Glow transform:

$$\mathrm{Tr}\left\{\left[\nabla^2_{\boldsymbol{x}_0}\log p(\boldsymbol{x}_0;\boldsymbol{\theta})\right](\boldsymbol{\Sigma}_q-\boldsymbol{\Sigma}_{p^*})\right\}$$

$$=\frac{\partial^2}{\partial z^2}\log p(\boldsymbol{z};\boldsymbol{\psi})\sum_{c=1}^{C}\left(\prod_{k=1}^{K}\sum_{j=1}^{C}u_{k,c,j}\right)^2\sum_{h,w}(\sigma^2_{q,h,w,c}-\sigma^2_{p^*,h,w,c})$$

Second Moment of Non-Training Distribution

Second Moment of Training Distribution

1x1 Conv. Params

Constant-Volume Flow Analysis

Plugging in the CV-Glow transform:

$$\mathrm{Tr}\left\{\left[\nabla^2_{\boldsymbol{x}_0}\log p(\boldsymbol{x}_0;\boldsymbol{\theta})\right](\boldsymbol{\Sigma}_q-\boldsymbol{\Sigma}_{p^*})\right\}$$

$$=\frac{\partial^2}{\partial z^2}\log p(\boldsymbol{z};\boldsymbol{\psi})\sum_{c=1}^{C}\left(\prod_{k=1}^{K}\sum_{j=1}^{C}u_{k,c,j}\right)^2\sum_{h,w}(\sigma^2_{q,h,w,c}-\sigma^2_{p^*,h,w,c})$$

Second Moment of Non-Training Distribution

Second Moment of Training Distribution

1x1 Conv. Params

Sums over channel dimensions

Product over steps in flow

Sum over spatial dimensions

Constant-Volume Flow Analysis

Plugging in the CV-Glow transform:

$$\mathrm{Tr}\left\{\left[\nabla^2_{\boldsymbol{x}_0}\log p(\boldsymbol{x}_0;\boldsymbol{\theta})\right](\boldsymbol{\Sigma}_q-\boldsymbol{\Sigma}_{p^*})\right\}$$

$$=\frac{\partial^2}{\partial z^2}\log p(\boldsymbol{z};\boldsymbol{\psi})\sum_{c=1}^{C}\left(\prod_{k=1}^{K}\sum_{j=1}^{C}u_{k,c,j}\right)^2\sum_{h,w}(\sigma^2_{q,h,w,c}-\sigma^2_{p^*,h,w,c})$$

Second Moment of Non-Training Distribution

Second Moment of Training Distribution

< 0 for all log-concave densities (e.g. Gaussian)

Non-negative due to square

Constant-Volume Flow Analysis

Plugging in the CV-Glow transform:

$$\mathrm{Tr}\left\{\left[\nabla^2_{\boldsymbol{x}_0}\log p(\boldsymbol{x}_0;\boldsymbol{\theta})\right](\boldsymbol{\Sigma}_q-\boldsymbol{\Sigma}_{p^*})\right\}$$

$$=\frac{\partial^2}{\partial z^2}\log p(\boldsymbol{z};\boldsymbol{\psi})\sum_{c=1}^{C}\left(\prod_{k=1}^{K}\sum_{j=1}^{C}u_{k,c,j}\right)^2\sum_{h,w}(\sigma^2_{q,h,w,c}-\sigma^2_{p^*,h,w,c})$$

< 0 for all log-concave densities (e.g. Gaussian)

Non-negative due to square

Second Moment of Non-Training Distribution

Second Moment of Training Distribution

Sign boils down to difference in moments. Speaks to asymmetric behavior.

Constant-Volume Flow Analysis

Plugging in the CIFAR-10 and SVHN statistics:

$$\mathbb{E}_{\text{SVHN}}[\log p(\boldsymbol{x};\boldsymbol{\theta})] - \mathbb{E}_{\text{CIFAR10}}[\log p(\boldsymbol{x};\boldsymbol{\theta})]$$

$$\approx \frac{1}{2\sigma_{\boldsymbol{\psi}}^2}\left[\alpha_1^2 \cdot 12.3 + \alpha_2^2 \cdot 6.5 + \alpha_3^2 \cdot 14.5\right] \geq 0 \quad \text{where} \quad \alpha_c = \prod_{k=1}^{K}\sum_{j=1}^{C} u_{k,c,j}$$

Differences in variances in the three spatial dimensions

Constant-Volume Flow Analysis

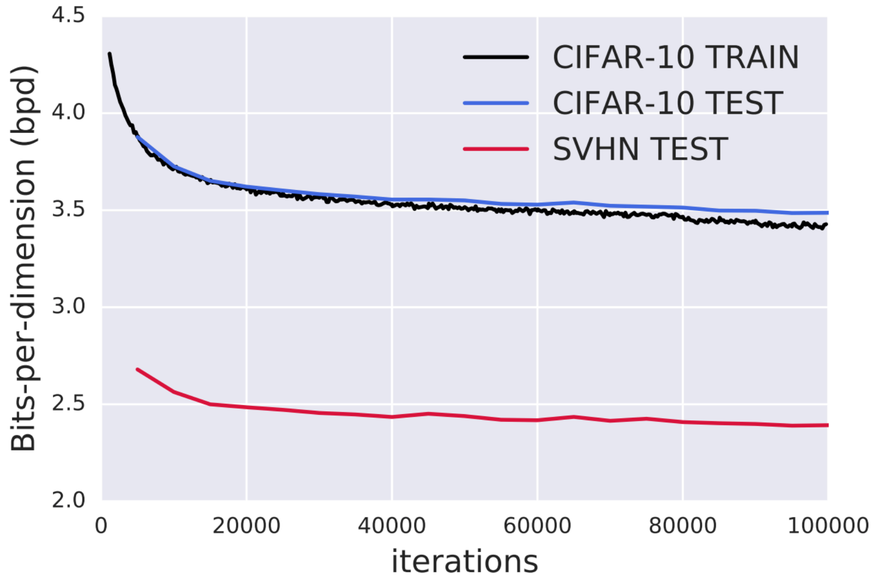
Plugging in the CIFAR-10 and SVHN statistics:

$$\mathbb{E}_{\text{SVHN}}[\log p(\boldsymbol{x};\boldsymbol{\theta})] - \mathbb{E}_{\text{CIFAR10}}[\log p(\boldsymbol{x};\boldsymbol{\theta})]$$

$$\approx \frac{1}{2\sigma_{\psi}^2}\left[\alpha_1^2 \cdot 12.3 + \alpha_2^2 \cdot 6.5 + \alpha_3^2 \cdot 14.5\right] \geq 0 \quad \text{where} \quad \alpha_c = \prod_{k=1}^{K} \sum_{j=1}^{C} u_{k,c,j}$$

Differences in variances in the three spatial dimensions

The expression will be non-negative **for any** parameter setting of the CV flow….



Constant-Volume Flow Analysis

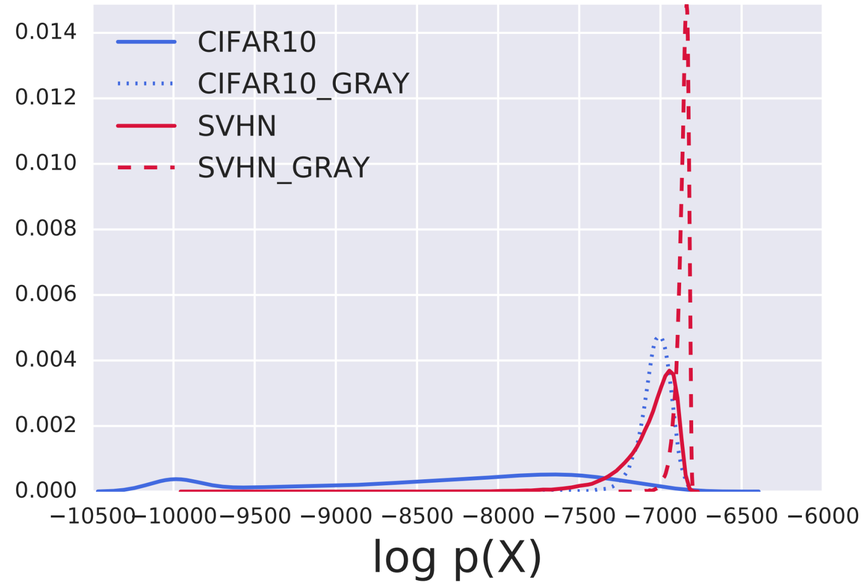
Plugging in the CIFAR-10 and SVHN statistics:

$$\mathbb{E}_{\text{SVHN}}[\log p(\boldsymbol{x};\boldsymbol{\theta})] - \mathbb{E}_{\text{CIFAR10}}[\log p(\boldsymbol{x};\boldsymbol{\theta})]$$

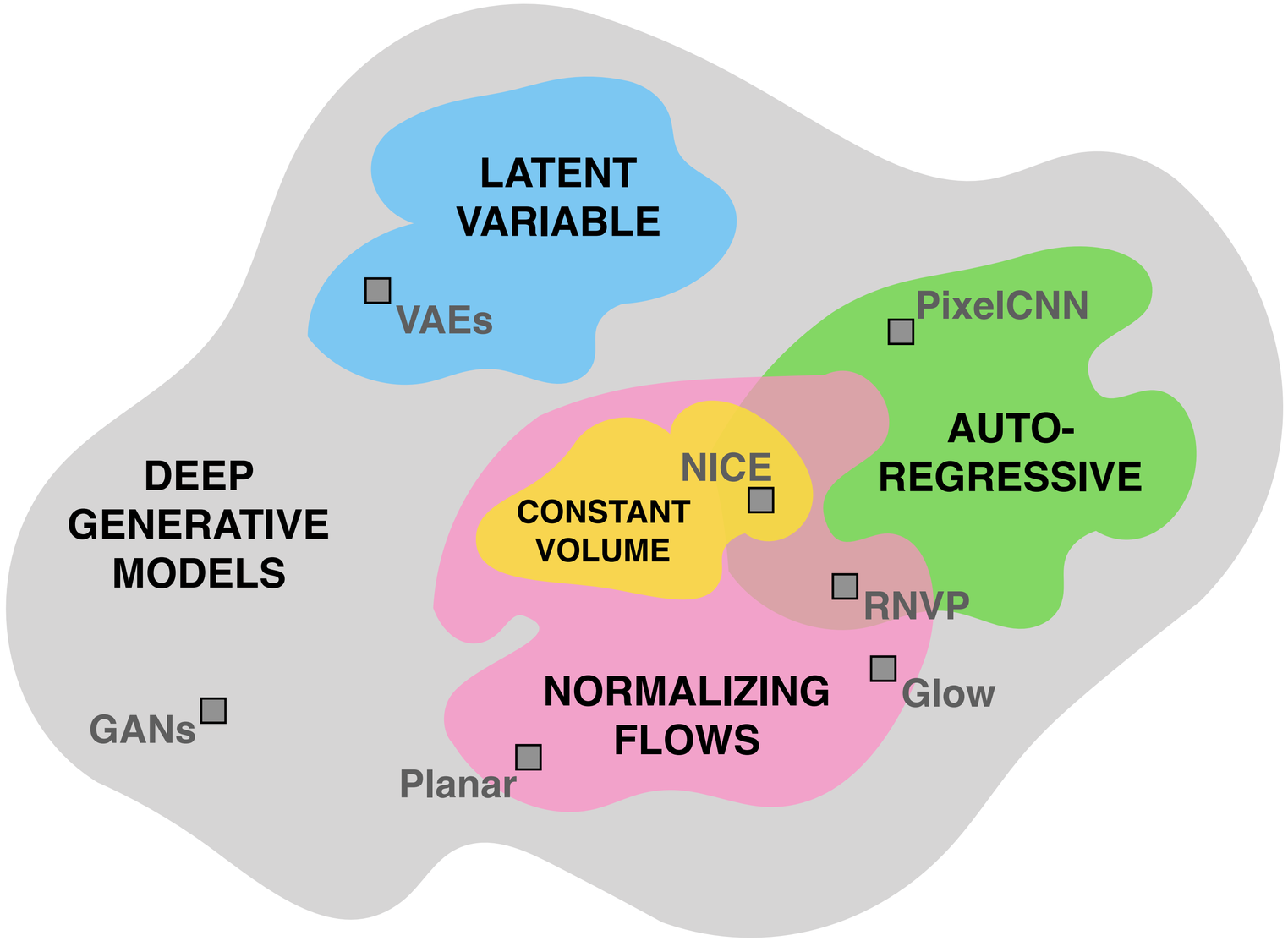
$$\approx \frac{1}{2\sigma_{\psi}^2}\left[\alpha_1^2 \cdot 12.3 + \alpha_2^2 \cdot 6.5 + \alpha_3^2 \cdot 14.5\right] \geq 0 \quad \text{where} \quad \alpha_c = \prod_{k=1}^{K} \sum_{j=1}^{C} u_{k,c,j}$$

Differences in variances in the three spatial dimensions

This also means that we can manipulate the relative log likelihoods just by changing the variance of the data. For natural images, this amounts to **graying**…

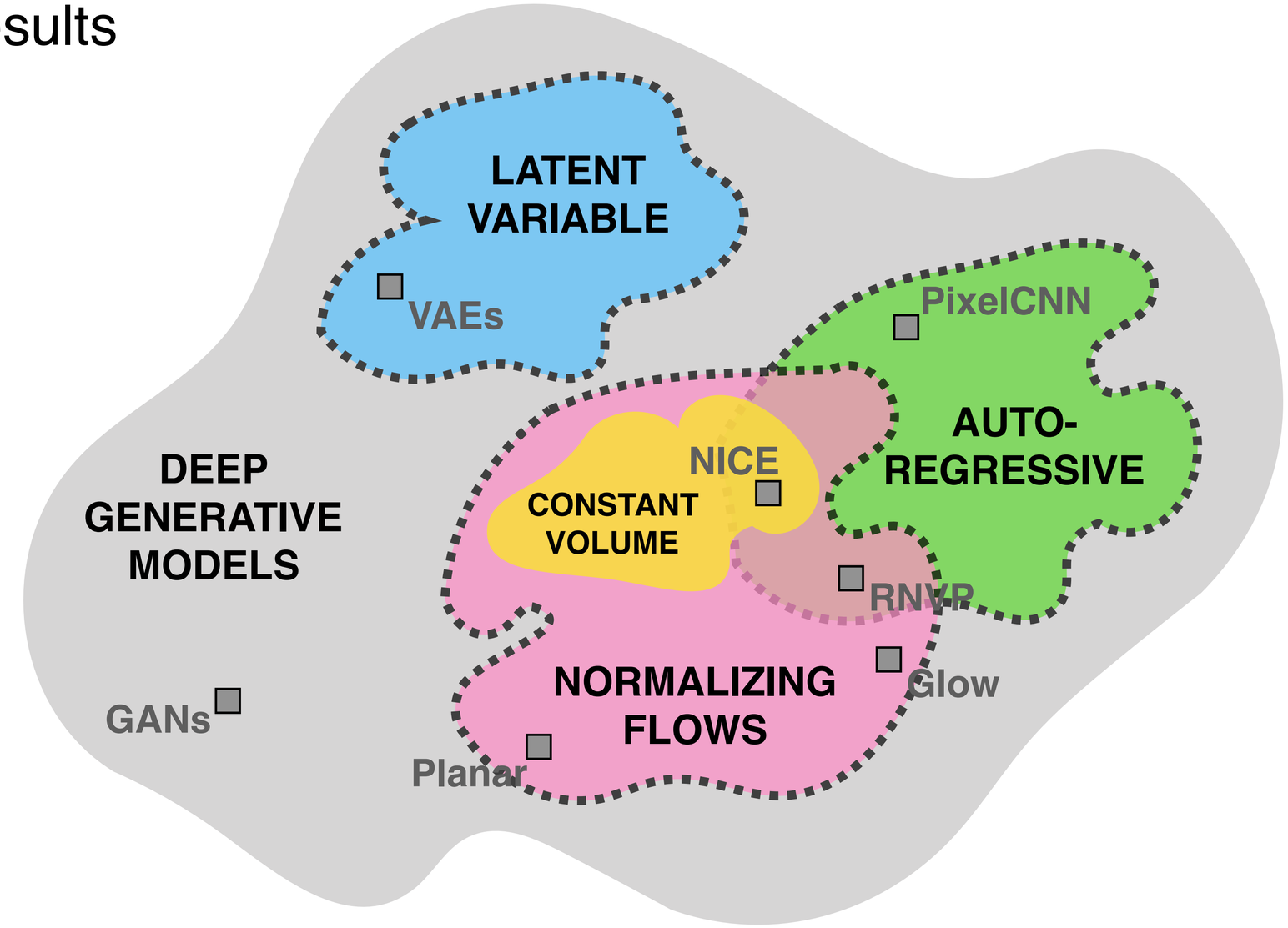


Summary of Results



Summary of Results

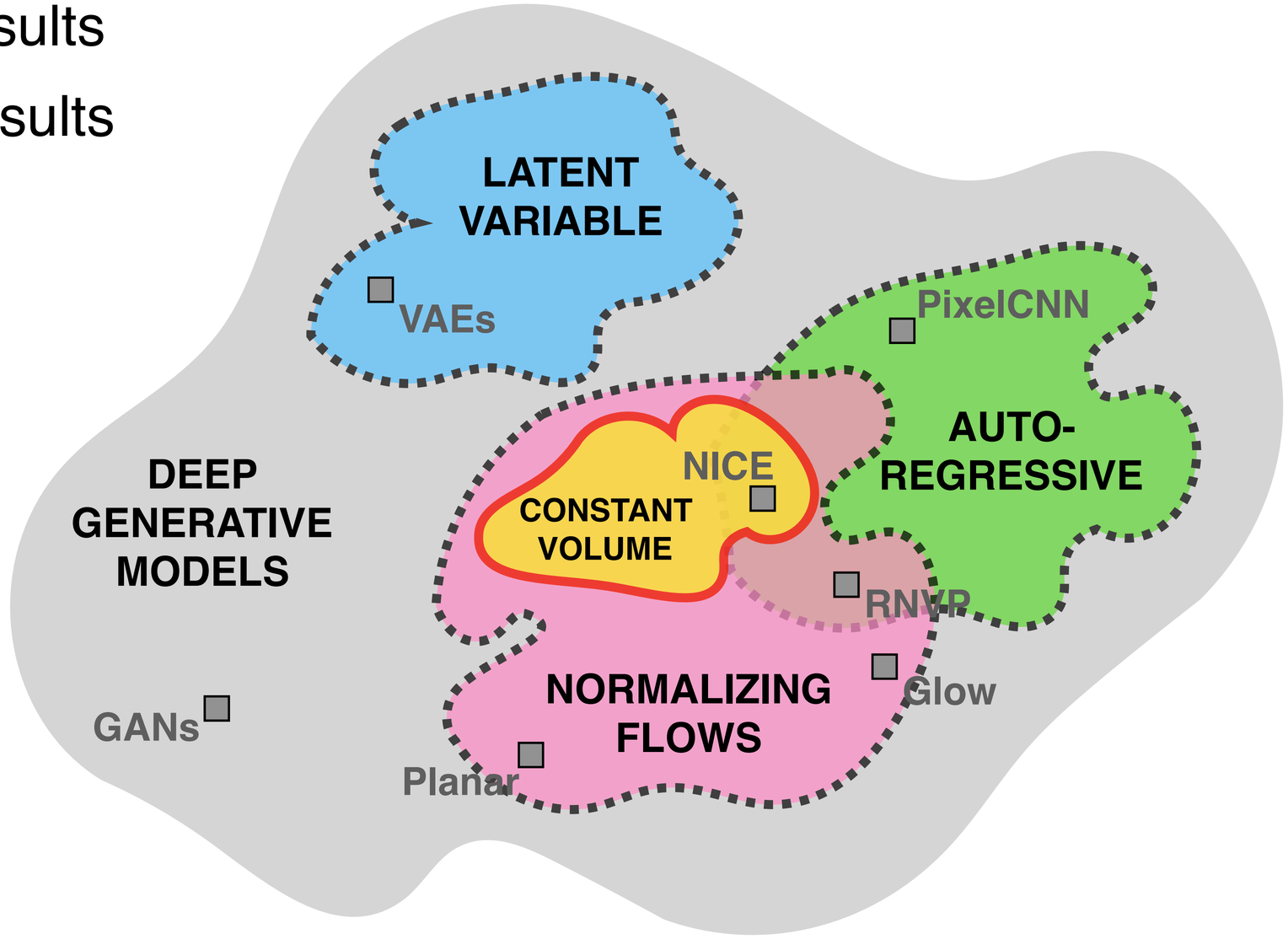
- Empirical Results



Summary of Results

Empirical Results

Analytical Results



FUTURE WORK

Open Questions

Include Entropy?

Should we include the **entropy term** that is dropped during model fitting?

$$\begin{aligned}\mathrm{KLD}[p^* \mid\mid p_{\boldsymbol{\theta}}] &= \mathbb{E}_{p^*}\left[-\log p(\mathbf{X};\boldsymbol{\theta})\right] - \mathbb{H}_{p^*}\left[\mathbf{X}\right] \\ &= \mathbb{E}_{p^*}\left[-\log p(\mathbf{X};\boldsymbol{\theta})\right] + \mathbb{E}_{p^*}\left[\log p^*(\mathbf{X})\right]\end{aligned}$$

Include Entropy?

Should we include the **entropy term** that is dropped during model fitting?

$$\begin{aligned}\mathrm{KLD}[p^* \mid\mid p_{\boldsymbol{\theta}}] &= \mathbb{E}_{p^*}\left[-\log p(\mathbf{X};\boldsymbol{\theta})\right] - \mathbb{H}_{p^*}\left[\mathbf{X}\right] \\ &= \mathbb{E}_{p^*}\left[-\log p(\mathbf{X};\boldsymbol{\theta})\right] + \mathbb{E}_{p^*}\left[\log p^*(\mathbf{X})\right]\end{aligned}$$

Not so easy to do…

Include Entropy?

Should we include the **entropy term** that is dropped during model fitting?

$$\begin{aligned}\mathrm{KLD}[p^* \;||\; p_{\boldsymbol{\theta}}] &= \mathbb{E}_{p^*}\left[-\log p(\mathbf{X};\boldsymbol{\theta})\right] - \mathbb{H}_{p^*}\left[\mathbf{X}\right] \\ &= \mathbb{E}_{p^*}\left[-\log p(\mathbf{X};\boldsymbol{\theta})\right] + \mathbb{E}_{p^*}\left[\log p^*(\mathbf{X})\right]\end{aligned}$$

Not so easy to do…

Approximate with model entropy?

$$\approx \mathbb{E}_{p^*}\left[-\log p(\mathbf{X};\boldsymbol{\theta})\right] - \mathbb{H}_{p_{\boldsymbol{\theta}}}\left[\mathbf{X}\right] \equiv \widetilde{\mathrm{KLD}}\left[p^* \;||\; p_{\boldsymbol{\theta}}\right]$$

Include Entropy?

Should we include the **entropy term** that is dropped during model fitting?

$$\begin{aligned}\mathrm{KLD}[p^* \mid\mid p_{\boldsymbol{\theta}}] &= \mathbb{E}_{p^*}\left[-\log p(\mathbf{X};\boldsymbol{\theta})\right] - \mathbb{H}_{p^*}\left[\mathbf{X}\right] \\ &= \mathbb{E}_{p^*}\left[-\log p(\mathbf{X};\boldsymbol{\theta})\right] + \mathbb{E}_{p^*}\left[\log p^*(\mathbf{X})\right]\end{aligned}$$

Not so easy to do…

Approximate with model entropy?

$$\approx \mathbb{E}_{p^*}\left[-\log p(\mathbf{X};\boldsymbol{\theta})\right] - \mathbb{H}_{p_{\boldsymbol{\theta}}}\left[\mathbf{X}\right] \equiv \widetilde{\mathrm{KLD}}\left[p^* \mid\mid p_{\boldsymbol{\theta}}\right]$$

But since max. likelihood is mass covering, probably…

$$\mathbb{H}_{p^*}\left[\mathbf{X}\right] \leq \mathbb{H}_{p_{\boldsymbol{\theta}}}\left[\mathbf{X}\right]$$

Include Entropy?

Should we include the **entropy term** that is dropped during model fitting?

$$\begin{aligned}\mathrm{KLD}[p^* \mid\mid p_{\boldsymbol{\theta}}] &= \mathbb{E}_{p^*}\left[-\log p(\mathbf{X};\boldsymbol{\theta})\right] - \mathbb{H}_{p^*}\left[\mathbf{X}\right] \\ &= \mathbb{E}_{p^*}\left[-\log p(\mathbf{X};\boldsymbol{\theta})\right] + \mathbb{E}_{p^*}\left[\log p^*(\mathbf{X})\right]\end{aligned}$$

Not so easy to do…

Approximate with model entropy?

$$\approx \mathbb{E}_{p^*}\left[-\log p(\mathbf{X};\boldsymbol{\theta})\right] - \mathbb{H}_{p_{\boldsymbol{\theta}}}\left[\mathbf{X}\right] \equiv \widetilde{\mathrm{KLD}}\left[p^* \mid\mid p_{\boldsymbol{\theta}}\right]$$

But since max. likelihood is mass covering, probably…

$$\mathbb{H}_{p^*}\left[\mathbf{X}\right] \leq \mathbb{H}_{p_{\boldsymbol{\theta}}}\left[\mathbf{X}\right] \implies \mathrm{KLD}\left[p^* \mid\mid p_{\boldsymbol{\theta}}\right] \geq \widetilde{\mathrm{KLD}}\left[p^* \mid\mid p_{\boldsymbol{\theta}}\right]$$

Wrong direction for a bound?

An Issue of Typical Sets?

Should we really be testing if an instance is in the model's **typical set**?

$$\text{For } \{\mathbf{X}_1, \ldots, \mathbf{X}_n, \ldots, \mathbf{X}_N\} \sim p_{\boldsymbol{\theta}}$$

$$\left| \frac{-1}{N} \sum_n \log p(\mathbf{X}_n; \boldsymbol{\theta}) - \mathbb{H}_{p_{\boldsymbol{\theta}}}[\mathbf{X}] \right| \leq \epsilon$$

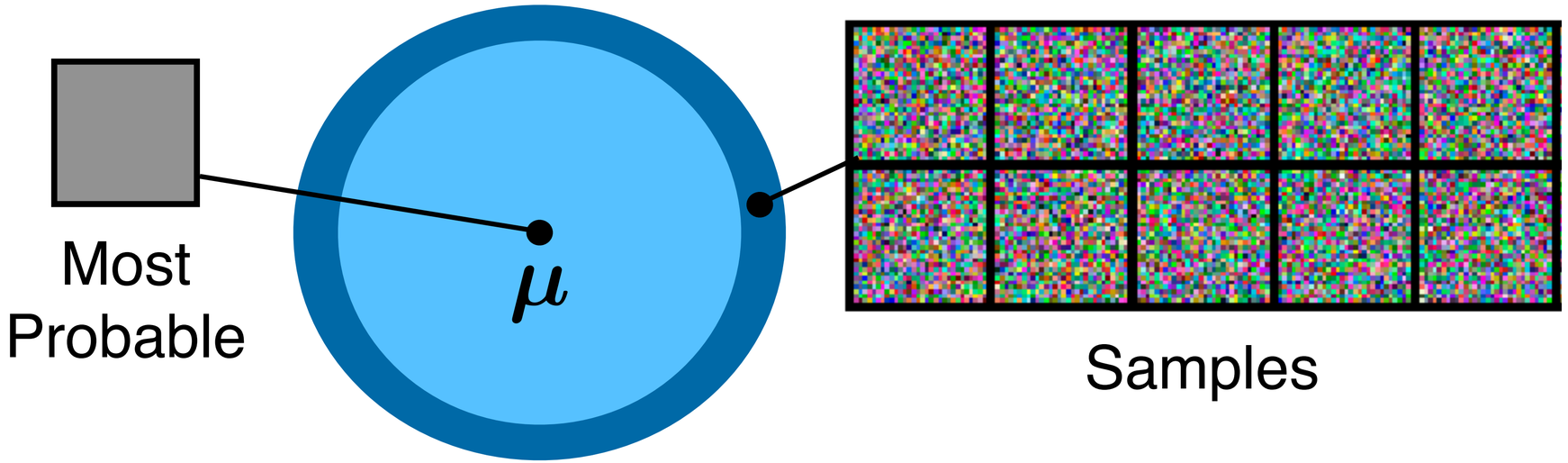
An Issue of Typical Sets?

Should we really be testing if an instance is in the model's **typical set**?

$$\text{For } \{\mathbf{X}_1, \ldots, \mathbf{X}_n, \ldots, \mathbf{X}_N\} \sim p_{\boldsymbol{\theta}}$$

$$\left| \frac{-1}{N} \sum_n \log p(\mathbf{X}_n; \boldsymbol{\theta}) - \mathbb{H}_{p_{\boldsymbol{\theta}}}[\mathbf{X}] \right| \leq \epsilon$$

Example: high-dimensional Gaussian density



An Issue of Typical Sets?

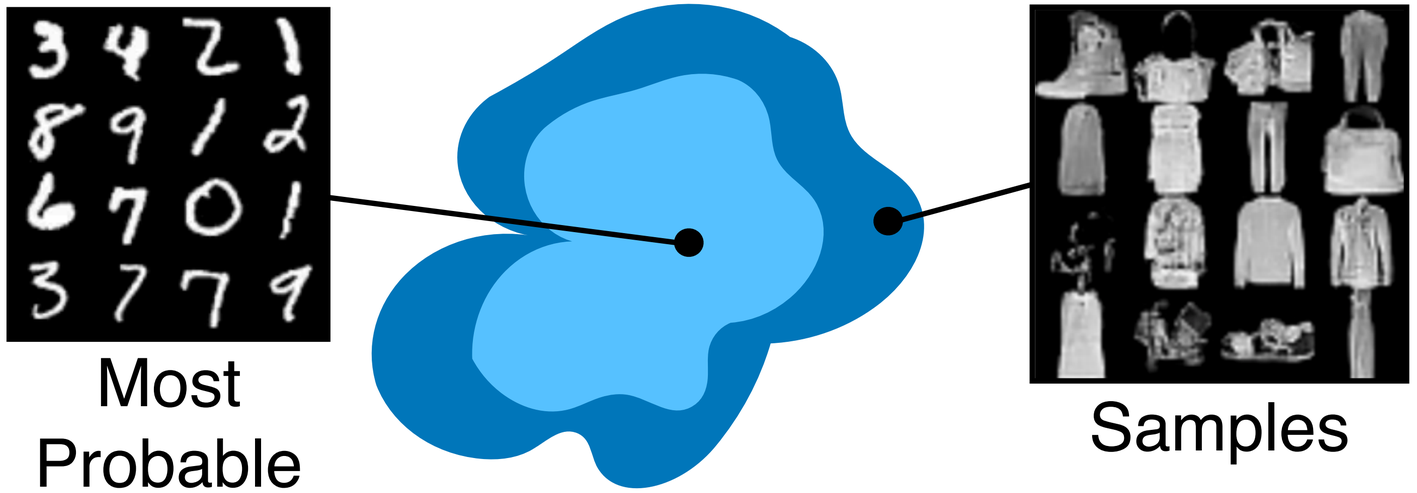
Should we really be testing if an instance is in the model's **typical set**?

$$\text{For } \{\mathbf{X}_1, \ldots, \mathbf{X}_n, \ldots, \mathbf{X}_N\} \sim p_{\boldsymbol{\theta}}$$

$$\left| \frac{-1}{N} \sum_n \log p(\mathbf{X}_n; \boldsymbol{\theta}) - \mathbb{H}_{p_{\boldsymbol{\theta}}}[\mathbf{X}] \right| \leq \epsilon$$

Similar behavior seems to be happening with deep generative models…

FashionMNIST
vs **MNIST**



*For purposes of illustration. We are not sure where the typical set of DGMs lives.

An Issue of Typical Sets?

Method: Replace samples with the possibly out-of-distribution observations:

$$\text{For } \{\mathbf{Y}_1, \ldots, \mathbf{Y}_n, \ldots, \mathbf{Y}_N\} \sim q^*$$

$$\left| \frac{-1}{N} \sum_n \log p(\mathbf{Y}_n; \boldsymbol{\theta}) - \mathbb{H}_{p_{\boldsymbol{\theta}}}[\mathbf{X}] \right| \overset{?}{\leq} \epsilon$$

An Issue of Typical Sets?

Method: Replace samples with the possibly out-of-distribution observations:

$$\text{For } \{\mathbf{Y}_1, \ldots, \mathbf{Y}_n, \ldots, \mathbf{Y}_N\} \sim q^*$$

$$\left| \frac{-1}{N} \sum_n \log p(\mathbf{Y}_n; \boldsymbol{\theta}) - \mathbb{H}_{p_{\boldsymbol{\theta}}}[\mathbf{X}] \right| \overset{?}{\leq} \epsilon$$

Issues:

1. Requires many (*N*) samples from q*.

An Issue of Typical Sets?

Method: Replace samples with the possibly out-of-distribution observations:

$$\text{For } \{\mathbf{Y}_1, \ldots, \mathbf{Y}_n, \ldots, \mathbf{Y}_N\} \sim q^*$$

$$\left| \frac{-1}{N} \sum_n \log p(\mathbf{Y}_n; \boldsymbol{\theta}) - \mathbb{H}_{p_{\boldsymbol{\theta}}}[\mathbf{X}] \right| \overset{?}{\leq} \epsilon$$

Issues:

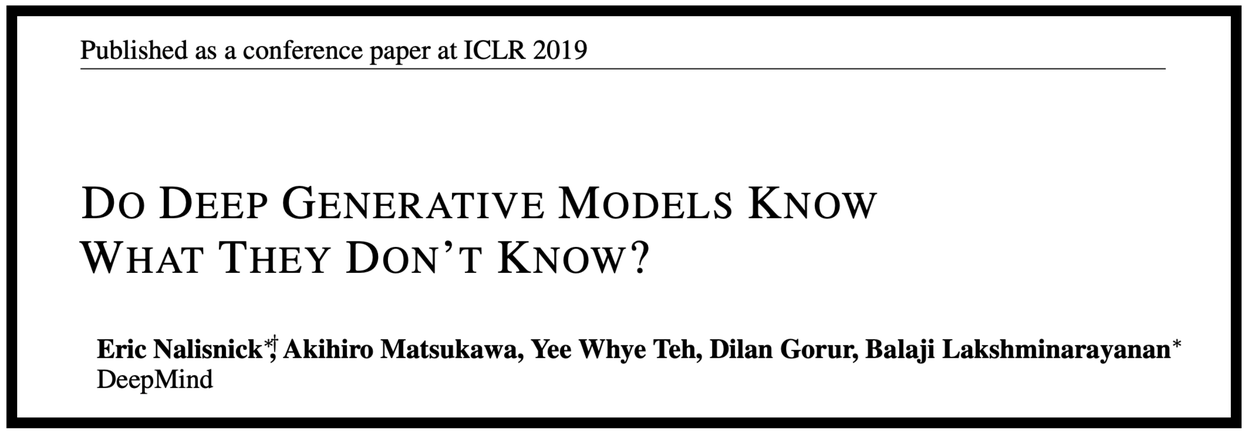
1. Requires many (*N*) samples from q*.
2. Since we care about outliers w.r.t. p*, the model entropy must be close to the data entropy, i.e.

$$\mathbb{H}_{p_{\boldsymbol{\theta}}}[\mathbf{X}] \approx \mathbb{H}_{p^*}[\mathbf{X}]$$

Conclusions

1

2



Published as a conference paper at ICLR 2019

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

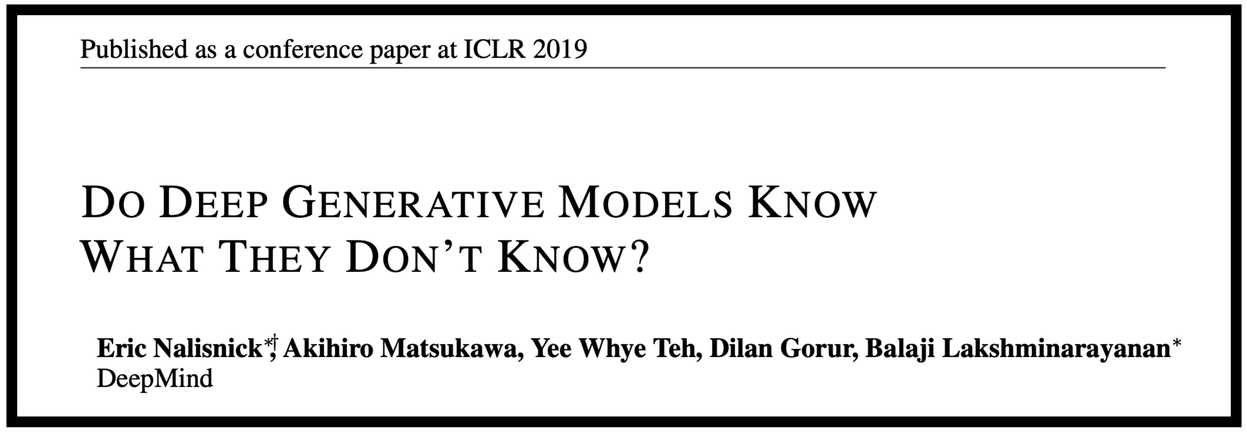
Eric Nalisnick[*†], **Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, Balaji Lakshminarayanan**[*]
DeepMind

https://openreview.net/forum?id=H1xwNhCcYm

Conclusions

1. **Failure of Log-Likelihood-Based Anomaly Detection**: Several classes of deep generative models cannot detect anomalous inputs using likelihood alone.

2.



Published as a conference paper at ICLR 2019

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

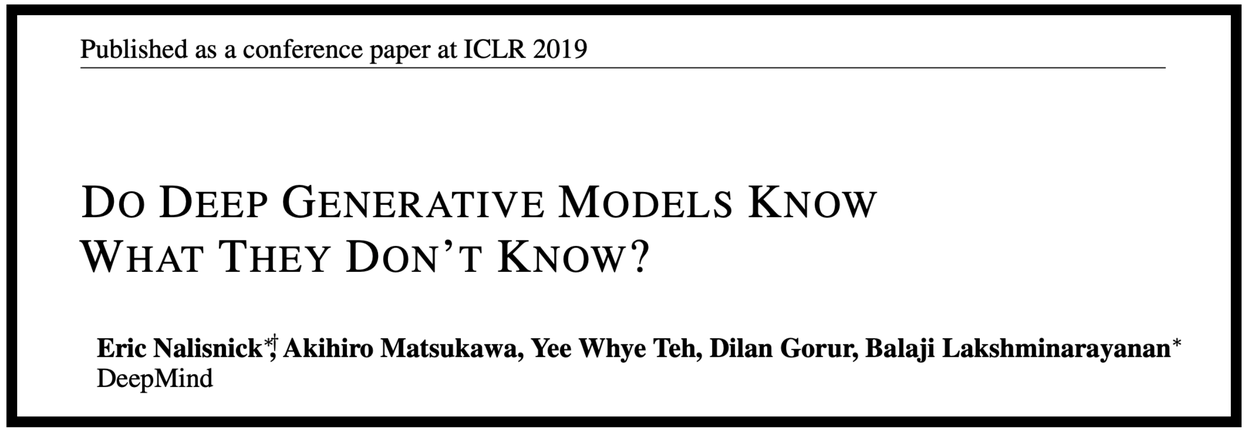
Eric Nalisnick[*†], **Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, Balaji Lakshminarayanan**[*]
DeepMind

https://openreview.net/forum?id=H1xwNhCcYm

Conclusions

1. **Failure of Log-Likelihood-Based Anomaly Detection**: Several classes of deep generative models cannot detect anomalous inputs using likelihood alone.

2. **Next Steps**: Investigate methods based on entropy and typical sets. How do we do this for small samples of potentially anomalous inputs?



https://openreview.net/forum?id=H1xwNhCcYm

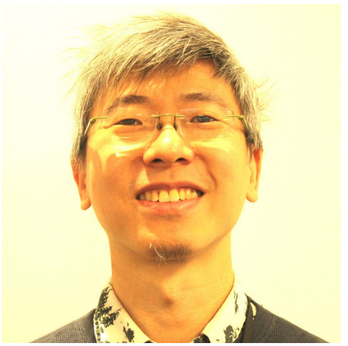
Thank you. Questions?

In collaboration with…

Aki Matsukawa

Balaji Lakshminarayanan

Dilan Gorur



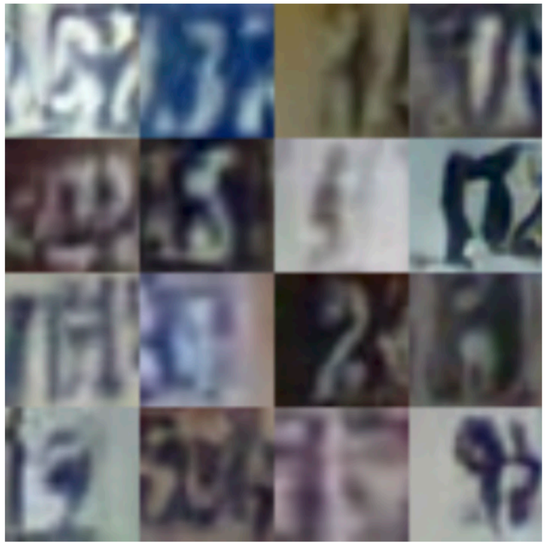
Yee Whye Teh

Appendix

(a) MNIST samples

(b) FashionMNIST samples

(c) CIFAR-10 samples



(d) SVHN samples

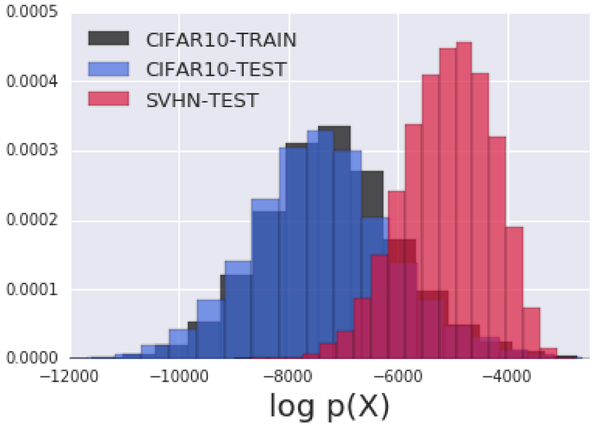
Figure 13: *Samples.* Samples from CV-Glow models used for analysis.

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

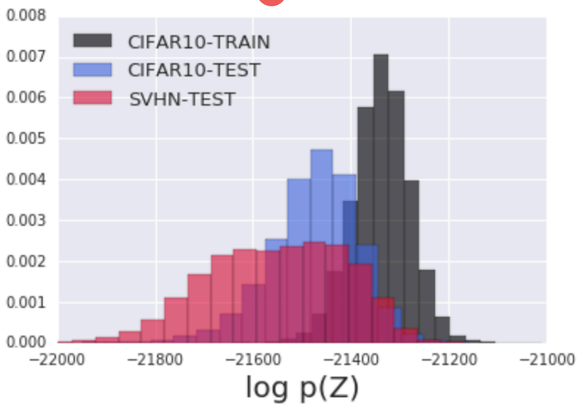
Data Set	Avg. Bits Per Dimension
Glow Trained on CIFAR-10	
CIFAR10-Train	3.386
CIFAR10-Test	3.464
SVHN-Test	**2.389**
Glow Trained on SVHN	
SVHN-Test	2.057

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.

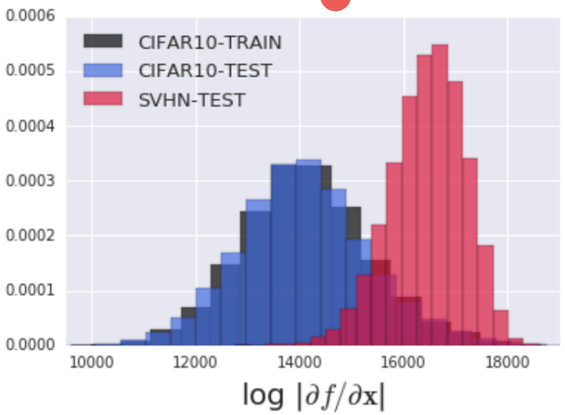
Decomposing the Change-of-Variables



CIFAR-10 vs SVHN



Distribution Term



Volume Term