UNDERSTANDING SEMANTIC ANOMALY DETECTION WITH DEEP CONVOLUTIONAL GENERATIVE NETWORKS

ML Collective Reading Group

Robin Tibor Schirrmeister Polina Kirichenko

UNIVERSITÄTS KLINIKUM FREIBURG

NYU

GENERATIVE MODELS

- Supervised learning: $\{X_i, y_i\}_{i=1}^N$ regression, classification, segmentation
- Unsupervised learning: $\{X_i\}_{i=1}^N$ approximate data distribution

$$p_\theta(x) \approx p^*(x)$$

- Generative models are trained with maximum likelihood:

$$\sum_{i=1}^{N} \log p_\theta(X_i) \to \max_\theta$$

GENERATIVE MODELS

- Simple density estimation models:

$$p_\theta(x) = \mathcal{N}(x|\mu, \Sigma) \qquad \text{or} \qquad p_\theta(x) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{N}(x|\mu_i, \Sigma_i)$$

- Deep likelihood-based generative models:
  - Autoregressive models
  - Variational Autoencoder
  - Normalizing flows

NORMALIZING FLOWS

- Deep generative models based on invertible neural networks

$$z \sim p_Z \quad x = f^{-1}(z)$$

- Base distribution p_Z is usually Gaussian $\mathcal{N}(0, I)$
- We can compute density in the data space exactly via *change of variable formula*:

$$p(x) = p_Z(f(x)) \cdot \left| \det \left(\frac{\partial f}{\partial x} \right) \right|$$

- Fast sampling and density estimation in coupling layers based flows (RealNVP, Glow, ...)

NORMALIZING FLOWS

- Coupling layers:

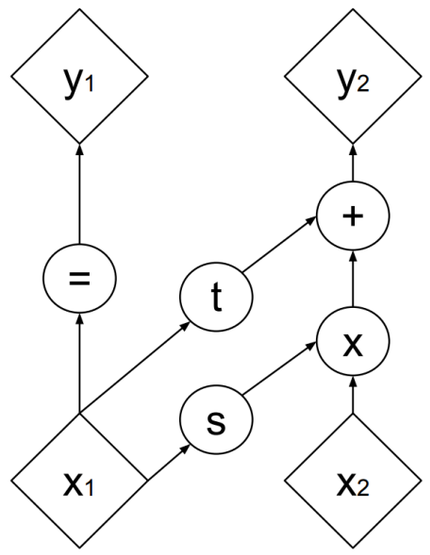
neural network

e.g. ResNet

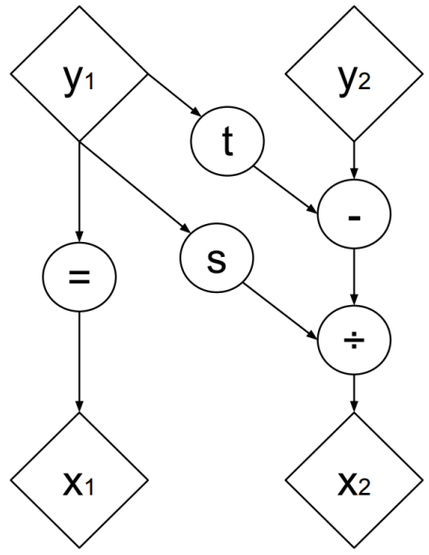
$$f_{\text{aff}}^{-1}(x_{\text{id}}, x_{\text{change}}) = (y_{\text{id}}, y_{\text{change}}), \quad \begin{cases} y_{\text{id}} = x_{\text{id}} \\ y_{\text{change}} = (x_{\text{change}} + t(x_{\text{id}})) \odot \exp(s(x_{\text{id}})) \end{cases}$$

Tractable det Jacobian

$$\frac{\partial y}{\partial x^T} = \begin{bmatrix} \mathbb{I}_d & 0 \\ \frac{\partial y_{d+1:D}}{\partial x_{1:d}^T} & \operatorname{diag}\left(\exp\left[s\left(x_{1:d}\right)\right]\right) \end{bmatrix}$$



(a) Forward propagation



(b) Inverse propagation

Dinh et al, Density estimation using RealNVP

NORMALIZING FLOWS

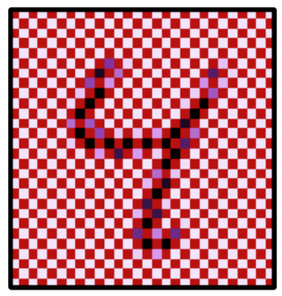
- Coupling layers:

neural network

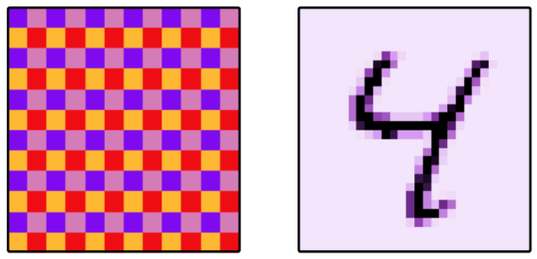
e.g. ResNet

$$f_{\text{aff}}^{-1}(x_{\text{id}}, x_{\text{change}}) = (y_{\text{id}}, y_{\text{change}}), \quad \begin{cases} y_{\text{id}} = x_{\text{id}} \\ y_{\text{change}} = (x_{\text{change}} + t(x_{\text{id}})) \odot \exp(s(x_{\text{id}})) \end{cases}$$

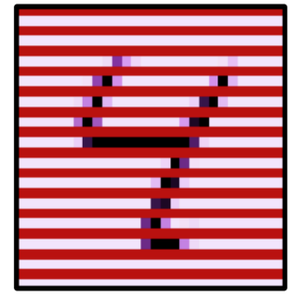
- x is split into x_{id} and x_{change} using masking



checkerboard



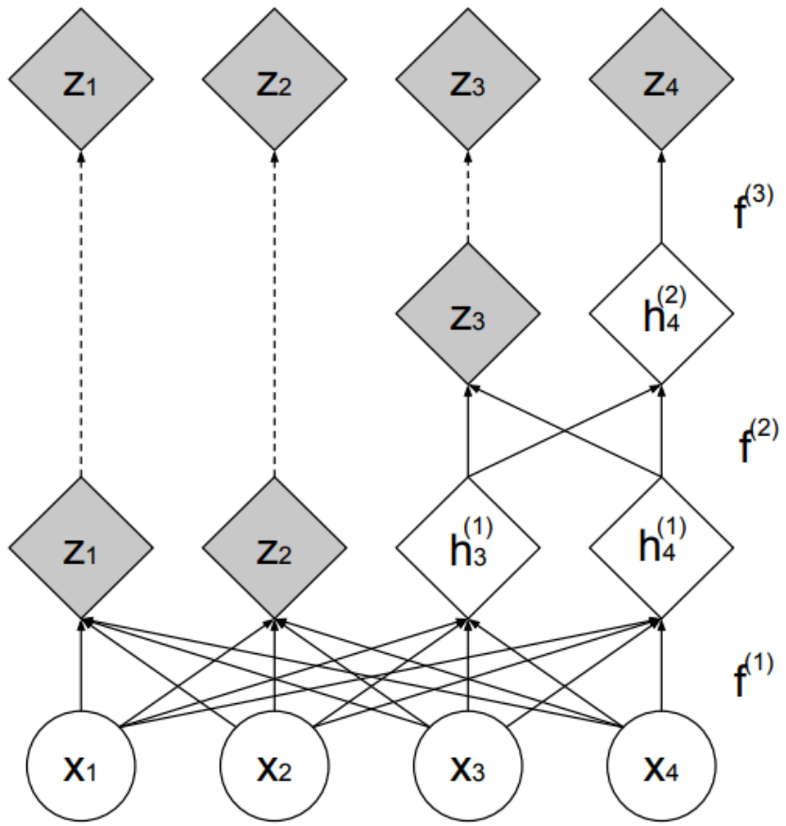
squeeze layer



channel-wise

NORMALIZING FLOWS

- Multi-scale architecture: at each scale half the variables are directly modeled as Gaussians while other half undergo further transformations

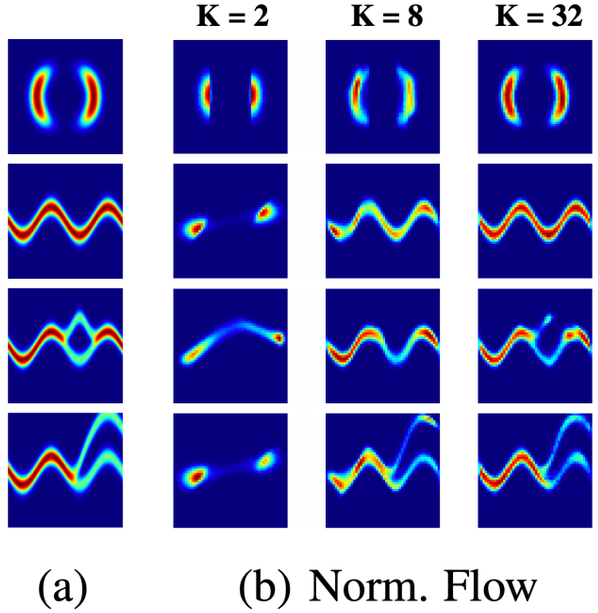


Dinh et al, Density estimation using RealNVP

NORMALIZING FLOWS

- Normalizing flows can model complex distributions
- NFs are successfully applied in modeling and inference

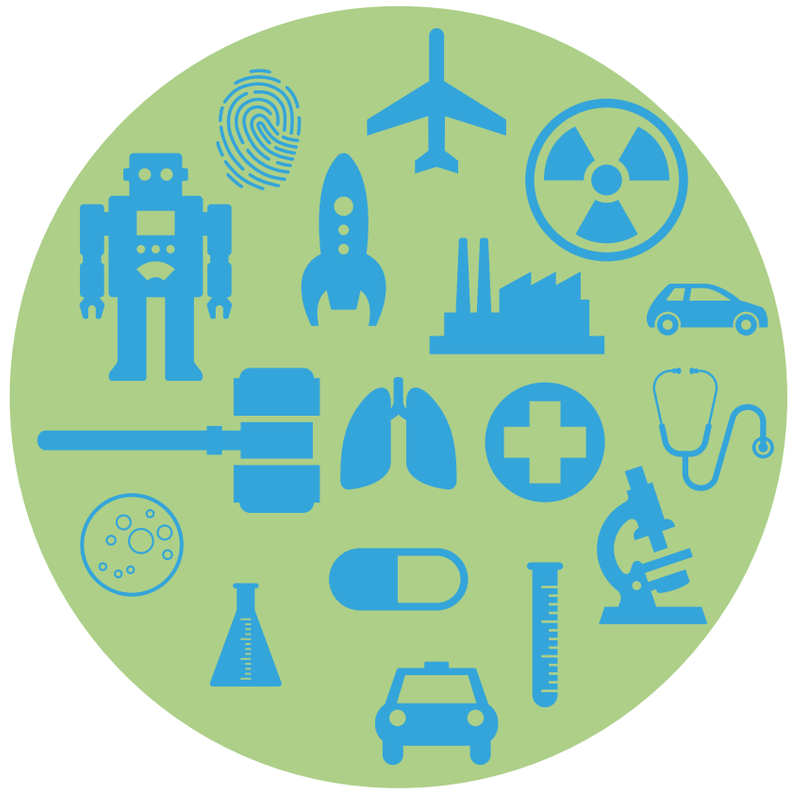
Kingma et al, Glow: Generative Flow with Invertible 1×1 Convolutions



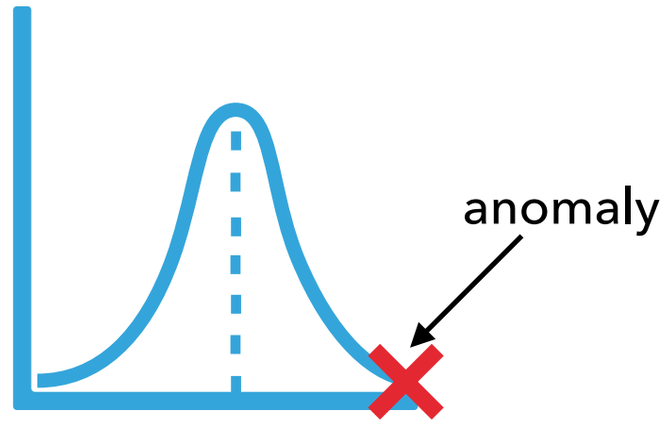
Rezende et al, Variational Inference with Normalizing Flows

OUT-OF-DISTRIBUTION DETECTION

- Anomaly detection is important in safety critical applications: medicine, autonomous cars, fraud detection and many more

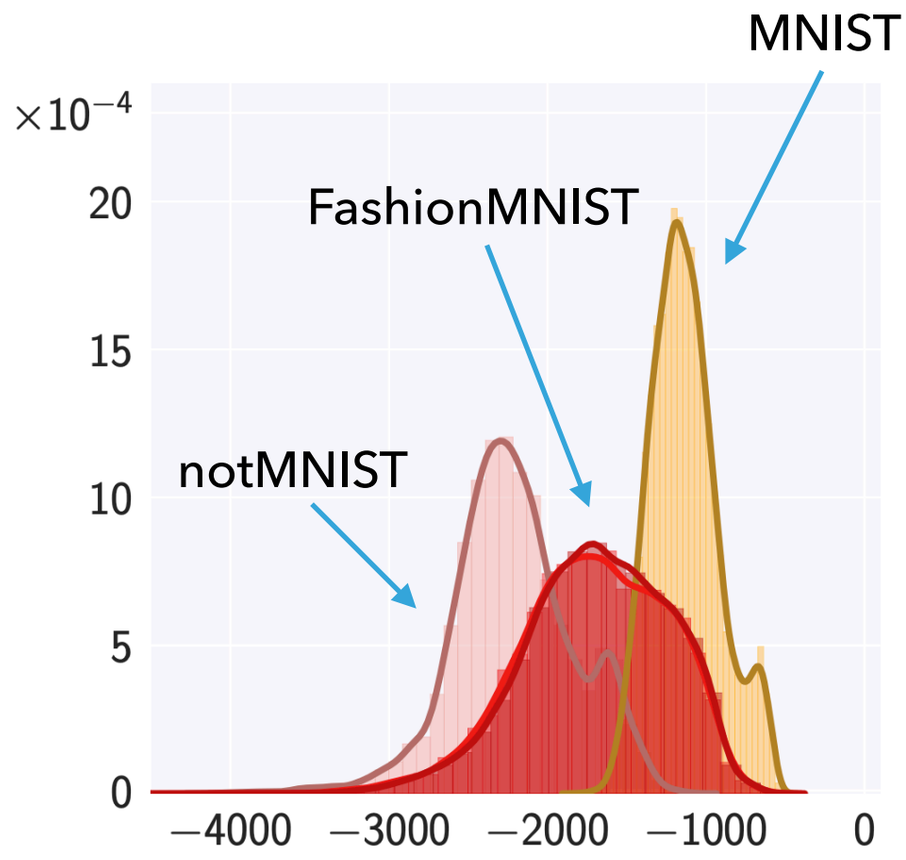


- Generative models present an attractive approach, OOD data is detected by low likelihood

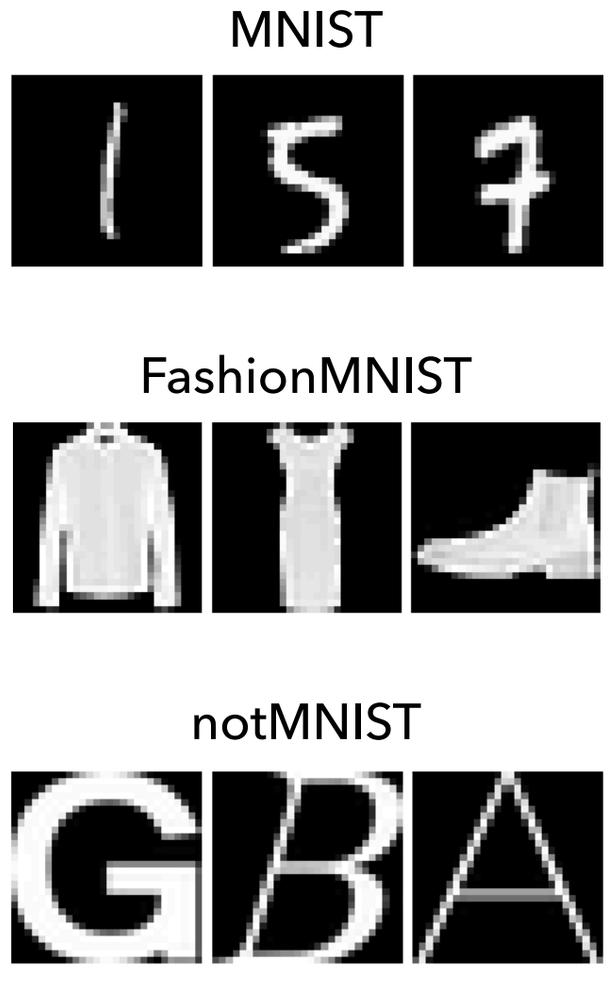


NORMALIZING FLOWS

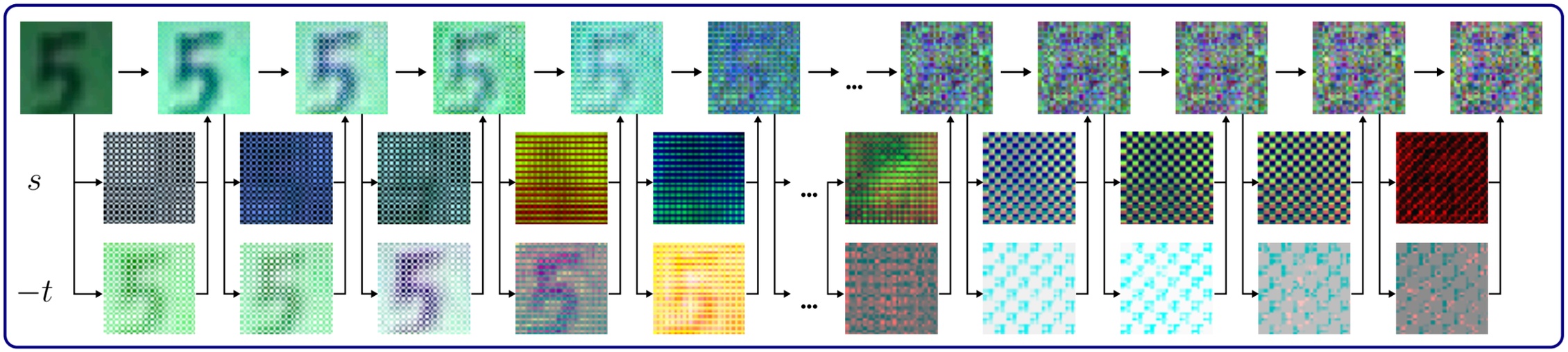
- Nalisnick et al. showed that flows sometimes assign higher likelihood to out-of-distribution data



RealNVP trained on FashionMNIST



Nalisnick et al, Do Deep Generative Models Know What They Don't Know?



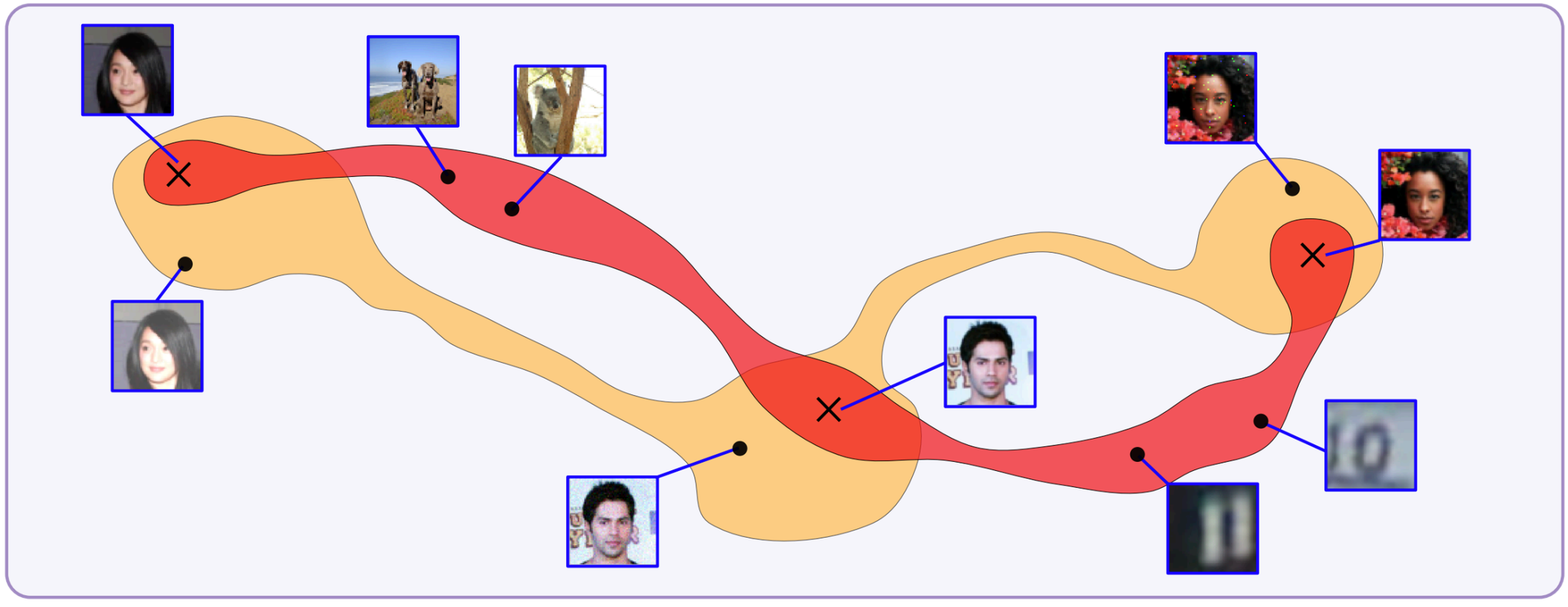
WHY NORMALIZING FLOWS FAIL TO DETECT OUT-OF-DISTRIBUTION DATA

NeurIPS | 2020

Polina Kirichenko Pavel Izmailov Andrew Gordon Wilson

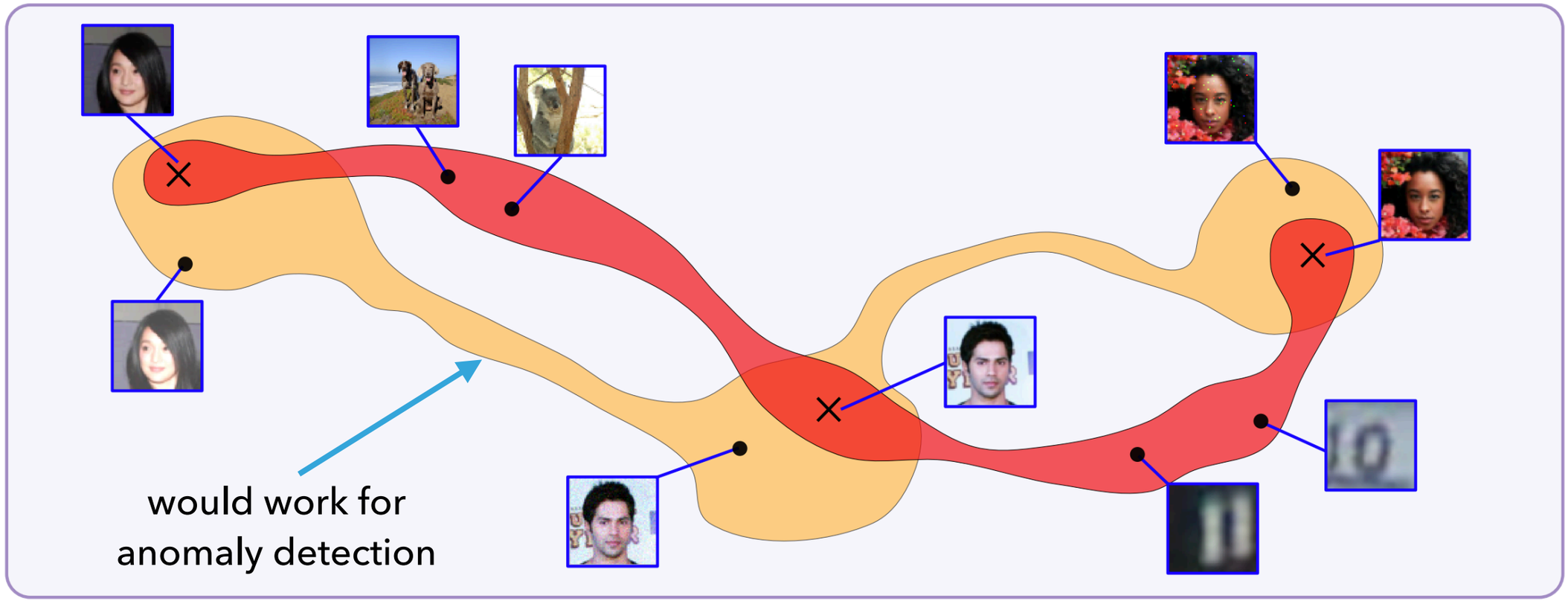
New York University

INDUCTIVE BIASES



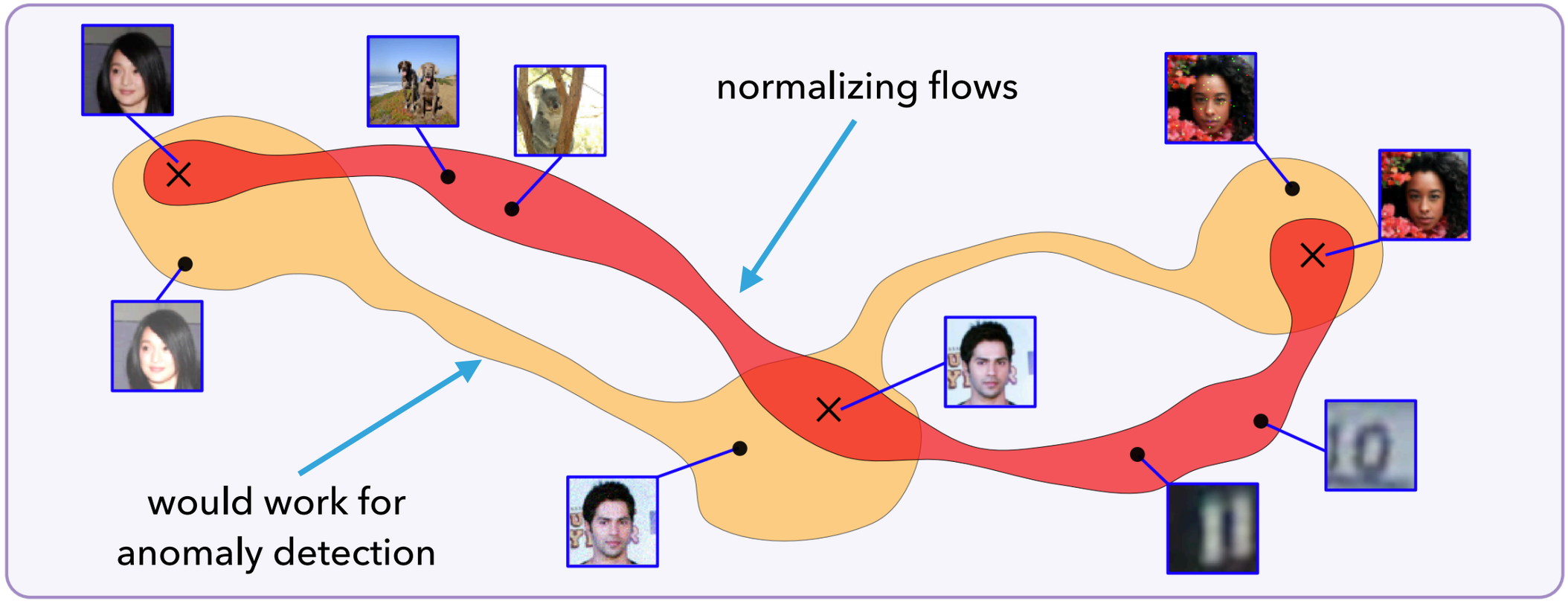
- Maximum likelihood encourages solutions which concentrate all mass on training data, i.e. overfit
- The likelihood assignment outside training data will be determined by inductive biases of the model

INDUCTIVE BIASES



- Maximum likelihood encourages solutions which concentrate all mass on training data, i.e. overfit
- The likelihood assignment outside training data will be determined by inductive biases of the model

INDUCTIVE BIASES



- Maximum likelihood encourages solutions which concentrate all mass on training data, i.e. overfit
- The likelihood assignment outside training data will be determined by inductive biases of the model

LATENT SPACE STRUCTURE

There is a direct correspondence between image and latent representation coordinates, the input shape is often visible in the latent representation

INDUCTIVE BIASES OF FLOWS

- Coupling layers:

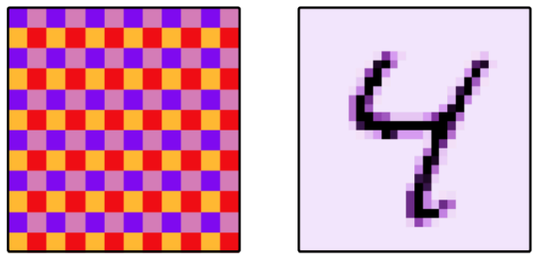
neural network

e.g. ResNet

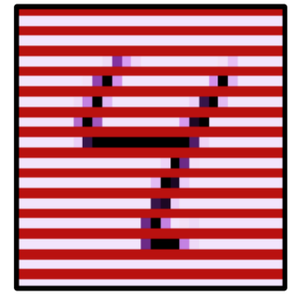
$$f_{\text{aff}}^{-1}(x_{\text{id}}, x_{\text{change}}) = (y_{\text{id}}, y_{\text{change}}), \quad \begin{cases} y_{\text{id}} = x_{\text{id}} \\ y_{\text{change}} = (x_{\text{change}} + t(x_{\text{id}})) \odot \exp(s(x_{\text{id}})) \end{cases}$$

- x is split into x_{id} and x_{change} using masking

checkerboard



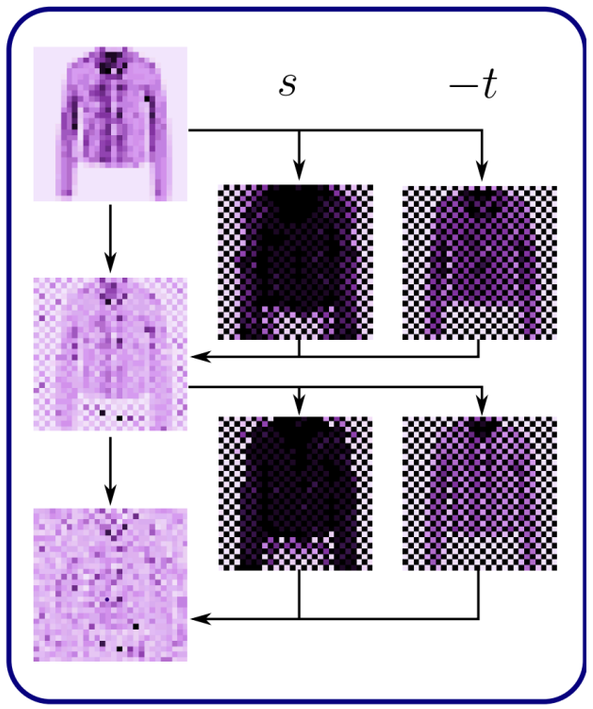
squeeze layer



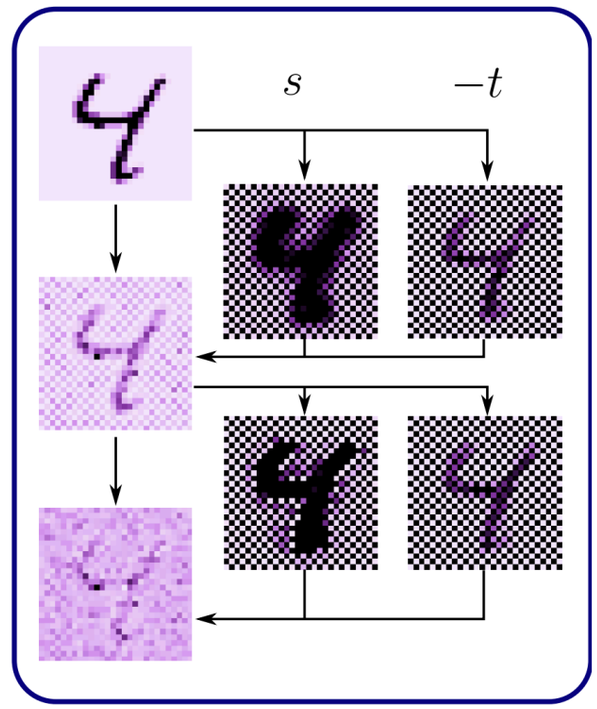
channel-wise

TRANSFORMATIONS OF COUPLING LAYERS

Affine coupling layer network predicts the scale and shift which directly model the masked pixels



(a) Checkerboard



(b) Checkerboard, OOD

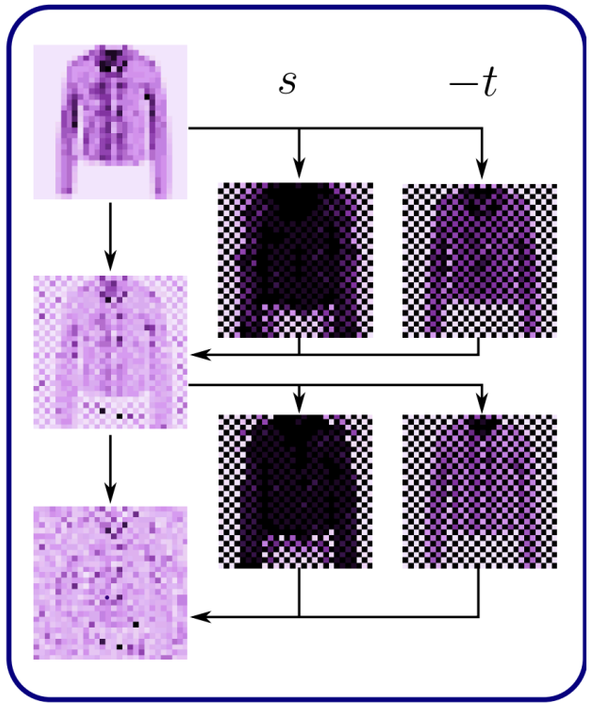
TRANSFORMATIONS OF COUPLING LAYERS

Intuitively:

$$\log p_X(x) = \log p_Z(f^{-1}(x)) + \log \left| \det \frac{\partial f^{-1}}{\partial x} \right|$$

$$\log \mathcal{N}(z|0, I) + \sum_{i=1}^{dim(x_{change})} s(x_{id})_i$$

The objective wants values s to be large while keeping the norm of z small: as a results $-t$ is approximating masked input and s represents the confidence of the approximation



(a) Checkerboard (b) Checkerboard, OOD

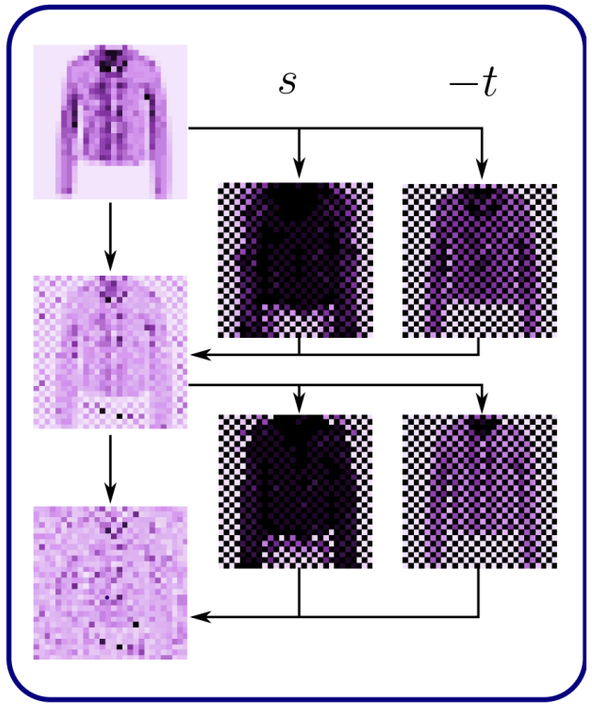
TRANSFORMATIONS OF COUPLING LAYERS

$$\log p_X(x) = \log p_Z(f^{-1}(x)) + \log \left| \det \frac{\partial f^{-1}}{\partial x} \right|$$

$$\log \mathcal{N}(z|0, I) + \sum_{i=1}^{dim(x_{change})} s(x_{id})_i$$

If $-t$ is an accurate approximation, $x+t$ will be small and this we can afford large values for s

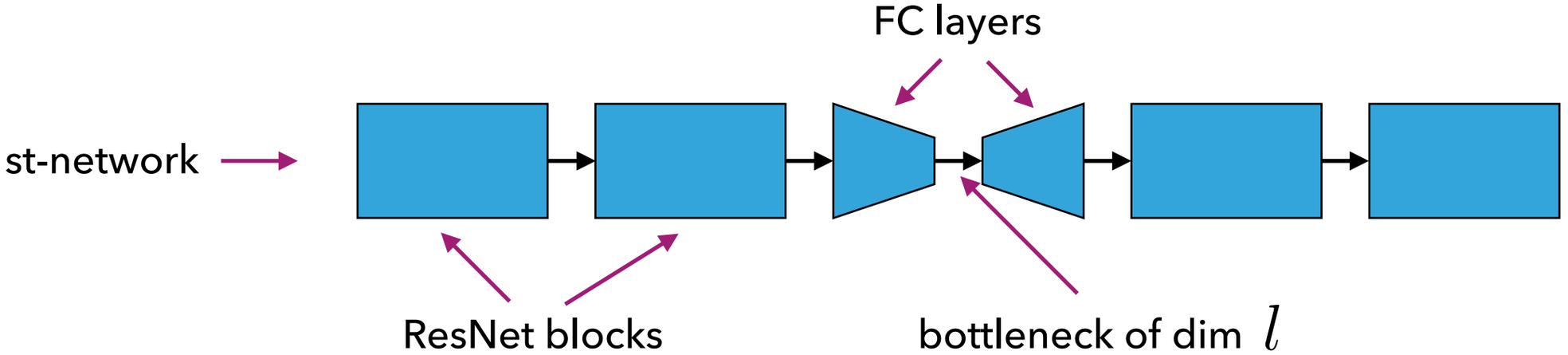
$$\begin{cases} y_{\text{id}} = x_{\text{id}} \\ y_{\text{change}} = (x_{\text{change}} + t(x_{\text{id}})) \odot \exp(s(x_{\text{id}})) \end{cases}$$

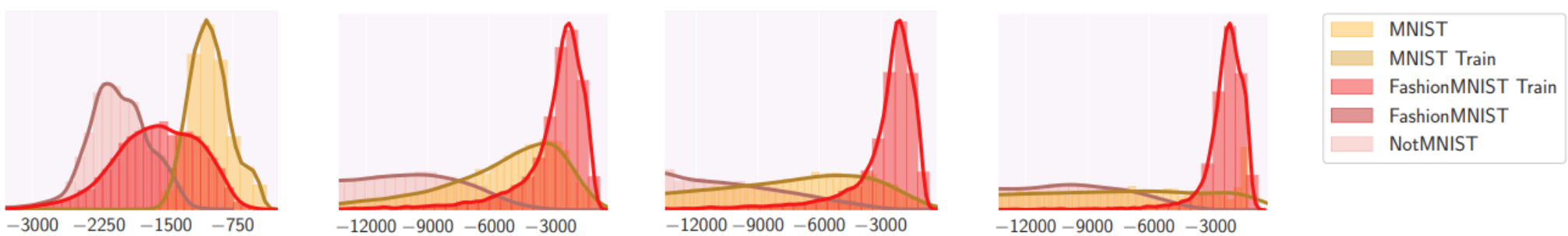


(a) Checkerboard (b) Checkerboard, OOD

CHANGING INDUCTIVE BIASES: BOTTLENECK

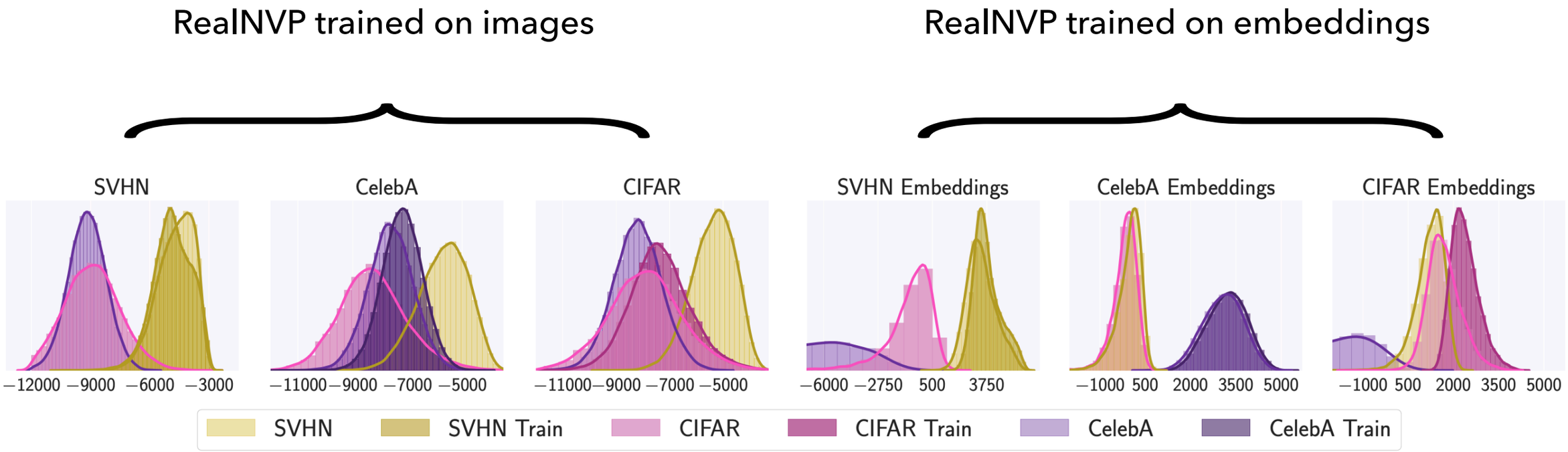
- We can restrict the capacity of coupling layer network to prevent it from predicting masked pixels easily for all structured images





(a) Baseline (b) $l = 100$ (c) $l = 50$ (d) $l = 10$

IMAGE EMBEDDINGS



The flow trained on image embeddings can detect OOD data

TAKEAWAYS

- Inductive biases (architectural choices) of generative models determine where they generalize (i.e. assign high likelihood) and what they consider out-of-distribution
- Normalizing flow based on coupling layers increase likelihood on all structured images when trained on one image dataset
- When trained on semantic embeddings, flows can detect anomalies

CONCLUSION

- See the paper arxiv.org/abs/2006.08545 for more details and experiments
- PyTorch Code available at github.com/PolinaKirichenko/flows_ood

Thank you!

@polkirichenko

polkirichenko.githhub.io

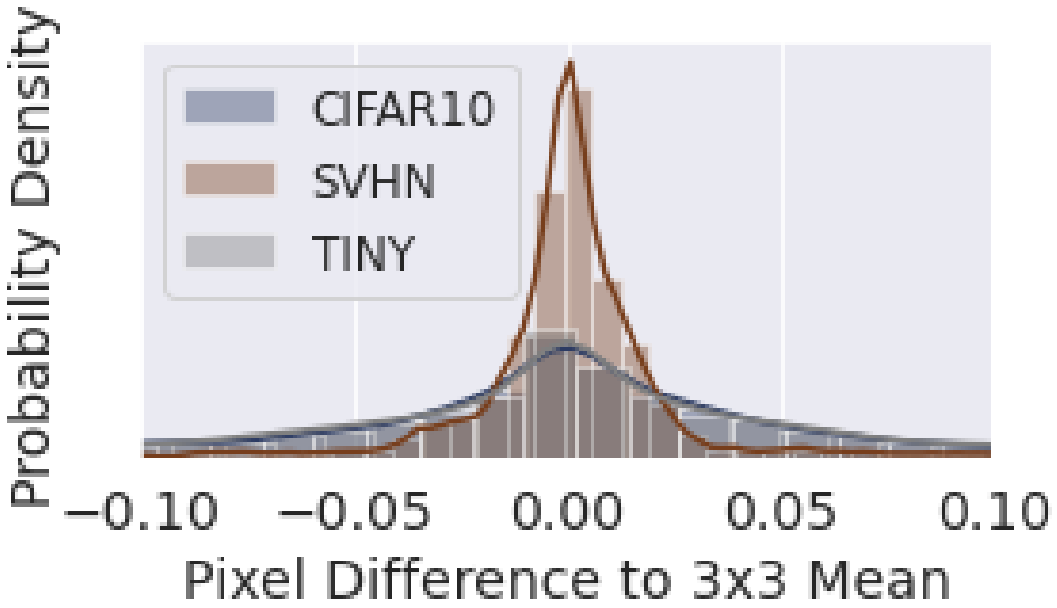
Understanding Anomaly Detection with Deep Invertible Networks through Hierarchies of Distributions and Features

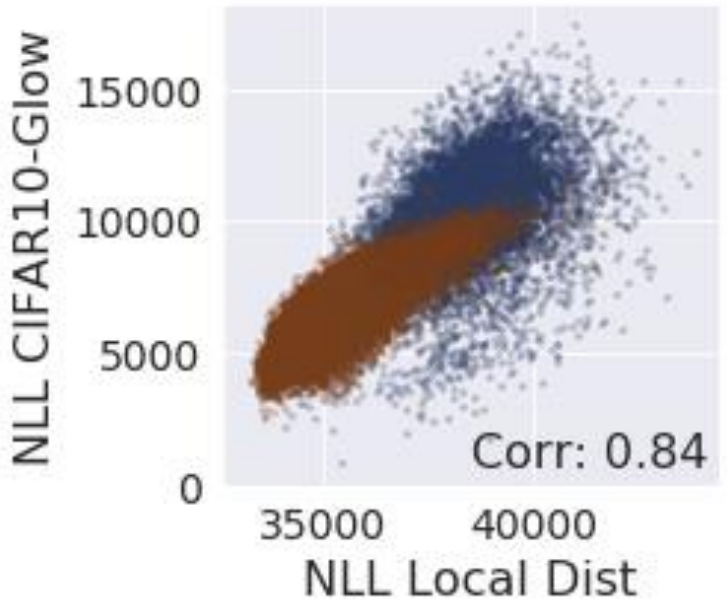
Robin Tibor Schirrmeister, Yuxuan Zhou, Tonio Ball, Dan Zhang

robintibor@gmail.com

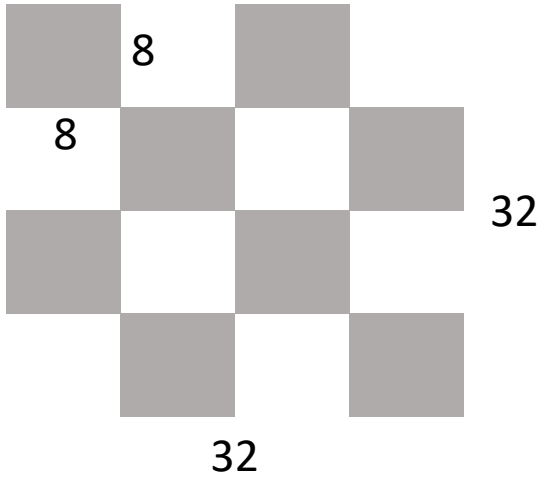
Further Analyses

Distribution of differences to neighbour pixels predicts Glow likelihood

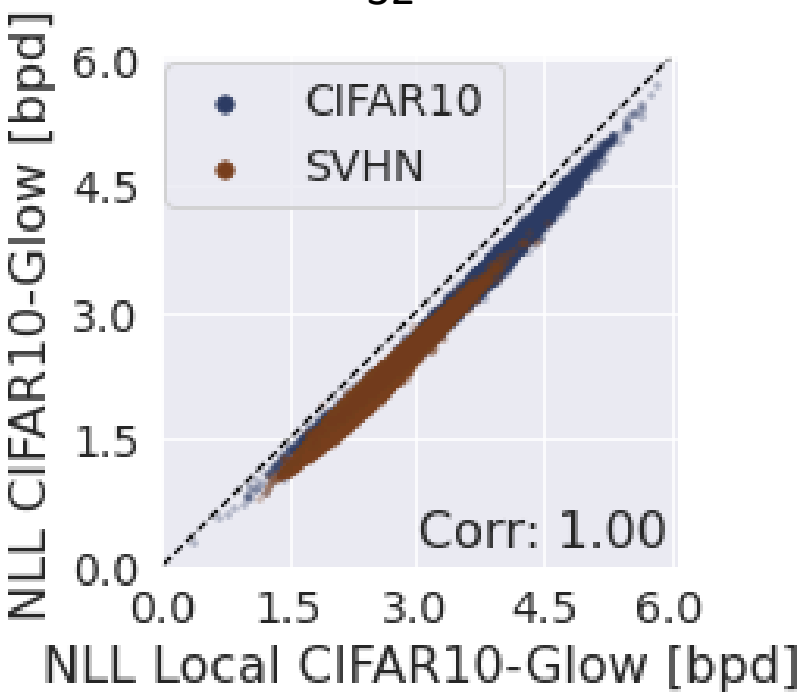


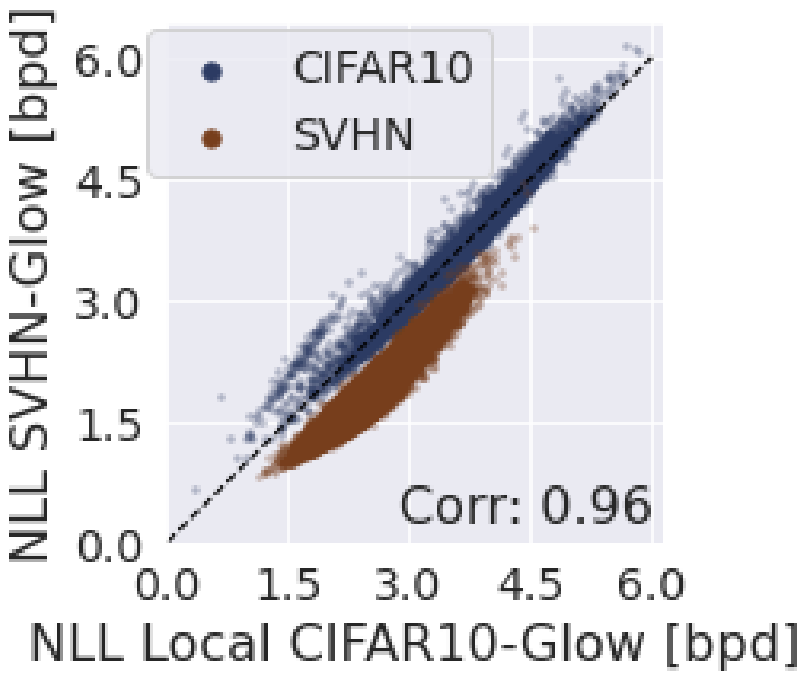


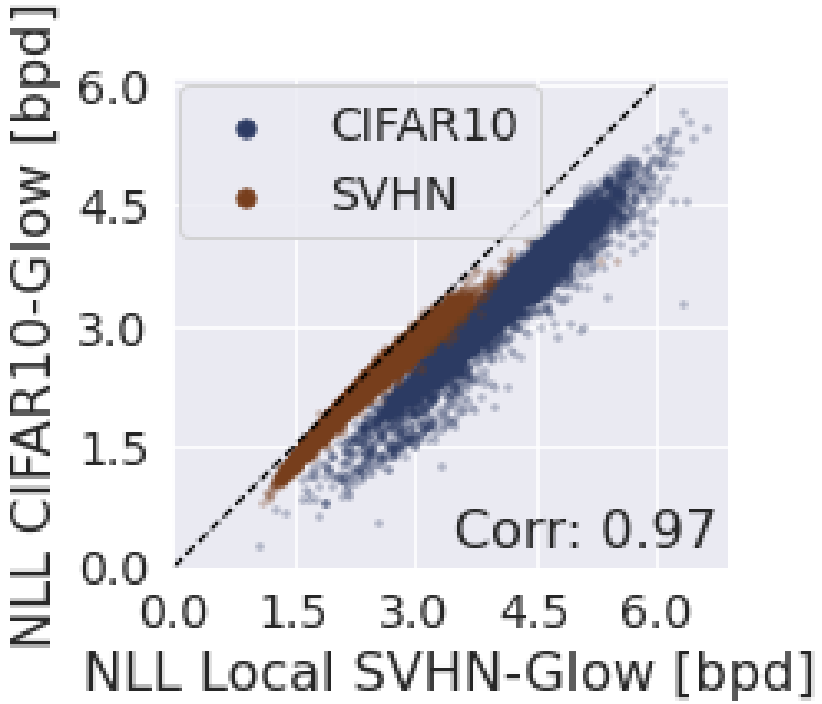
Common Local Features Dominate Model Likelihood



Local Glow processes 8x8 patches individually, never sees the full 32x32 image







Fully Connected Models Less Correlated

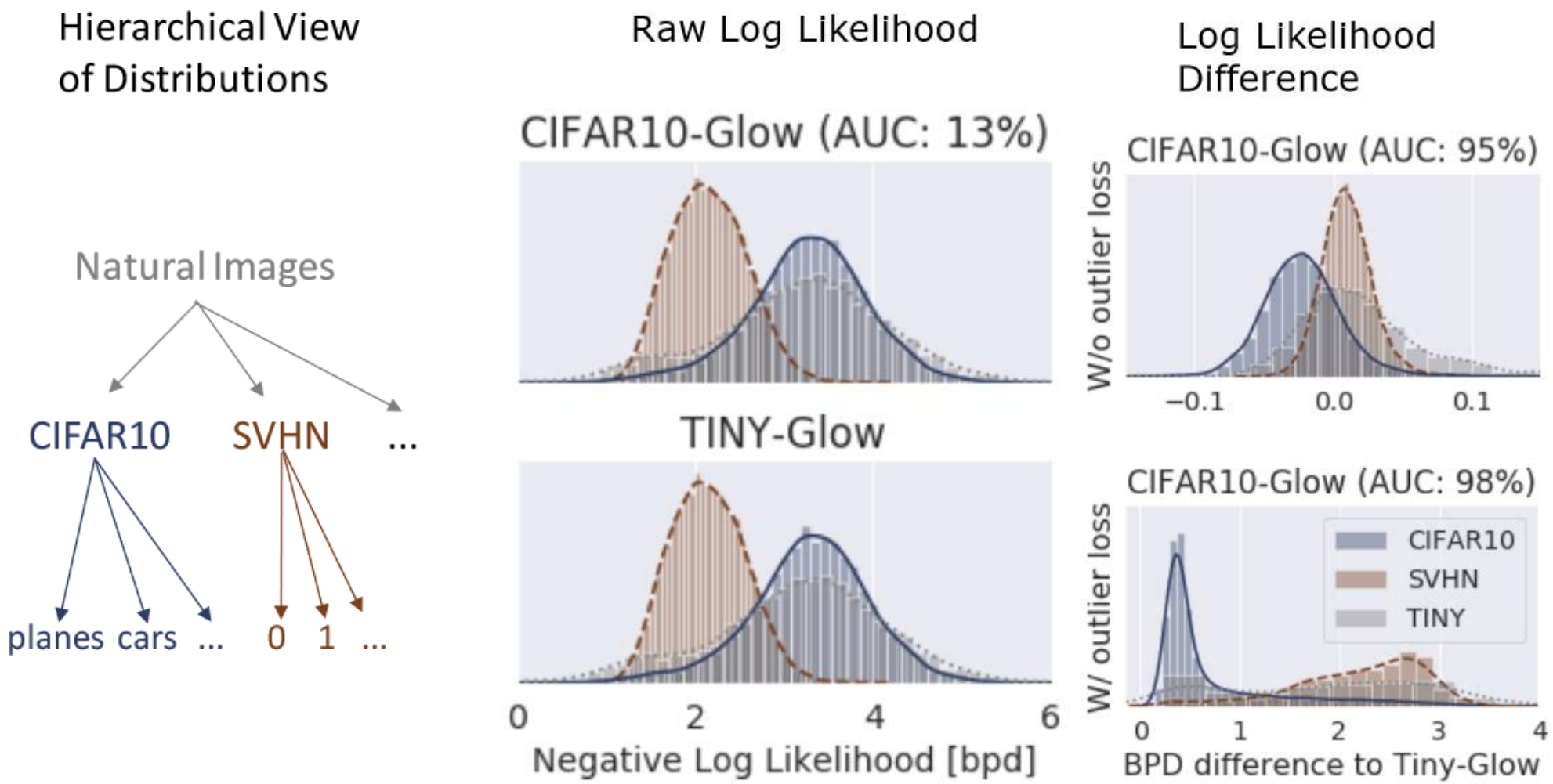
Likelihoods Rank Correlation

Conv Glow Trained on:	Local Glow (CIFAR10)	Dense Glow (CIFAR10)	Conv Glow (CIFAR10)
CIFAR10	100%	86%	100%
SVHN	96%	90%	97%
TINY	100%	86%	100%

Dense Glow improves Fashion-MNIST vs. MNIST AUC from 15% to 81%!
Unfortunately, does not work as well for CIFAR10

Solutions

Hierarchy of Distributions



Outlier Loss $\mathrm{L}_{\mathrm{out}} = -\lambda \cdot log(sigmoid(\frac{\log(p_g(x_g)) - \log(p_{in}(x_g))}{T}))$

https://arxiv.org/abs/1812.04606

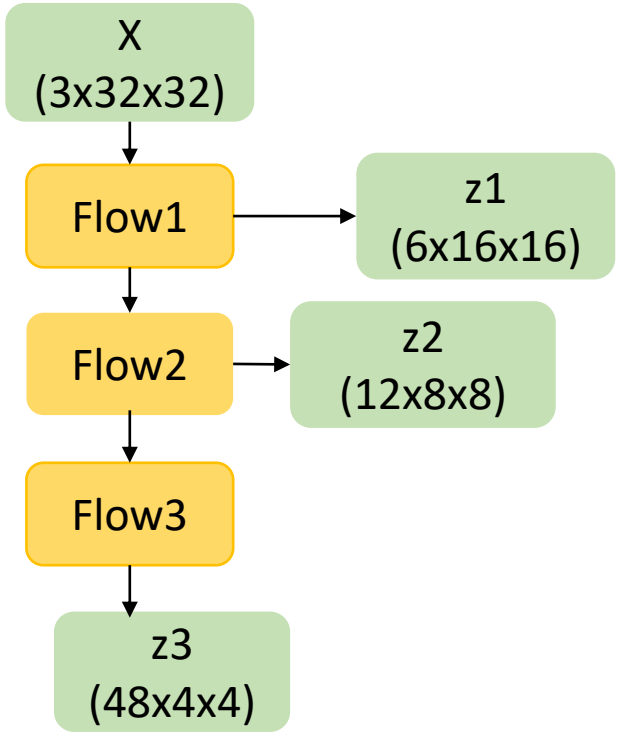
Log Likelihood Ratio and Outlier Loss

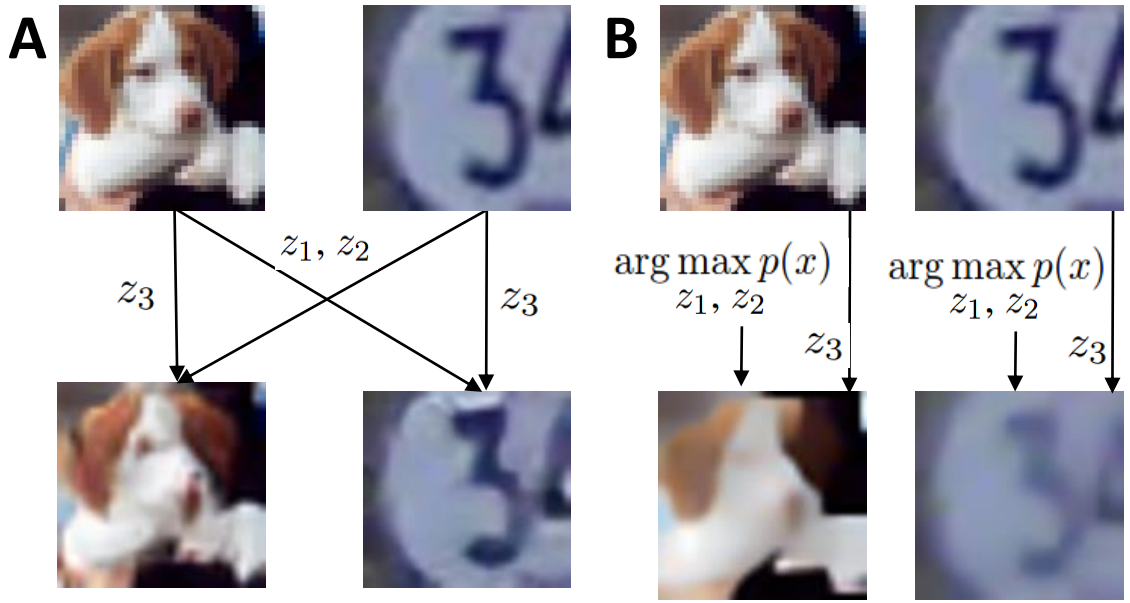
Log Likelihood Ratio $inlier\ score(x) = \log\left(\frac{p_{in}(x)}{p_g(x)}\right) = \log\big(p_{in}(x)\big) - \log(p_g(x))$

Outlier Loss $L_{out} = -\lambda \cdot log(sigmoid(\frac{\log\big(p_g(x_g)\big) - \log(p_{in}(x_g))}{T}))$

Total Loss $L = -\log(p_{in}(x_{in})) - \lambda \cdot log(sigmoid(\frac{\log\big(p_g(x_g)\big) - \log(p_{in}(x_g))}{T}))$

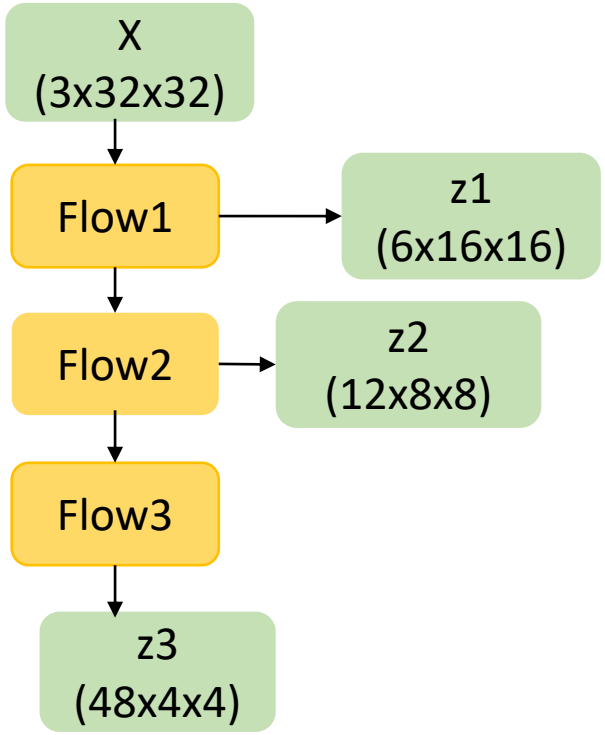
Last Glow-scale contains Semantic Information





More Maximization Examples

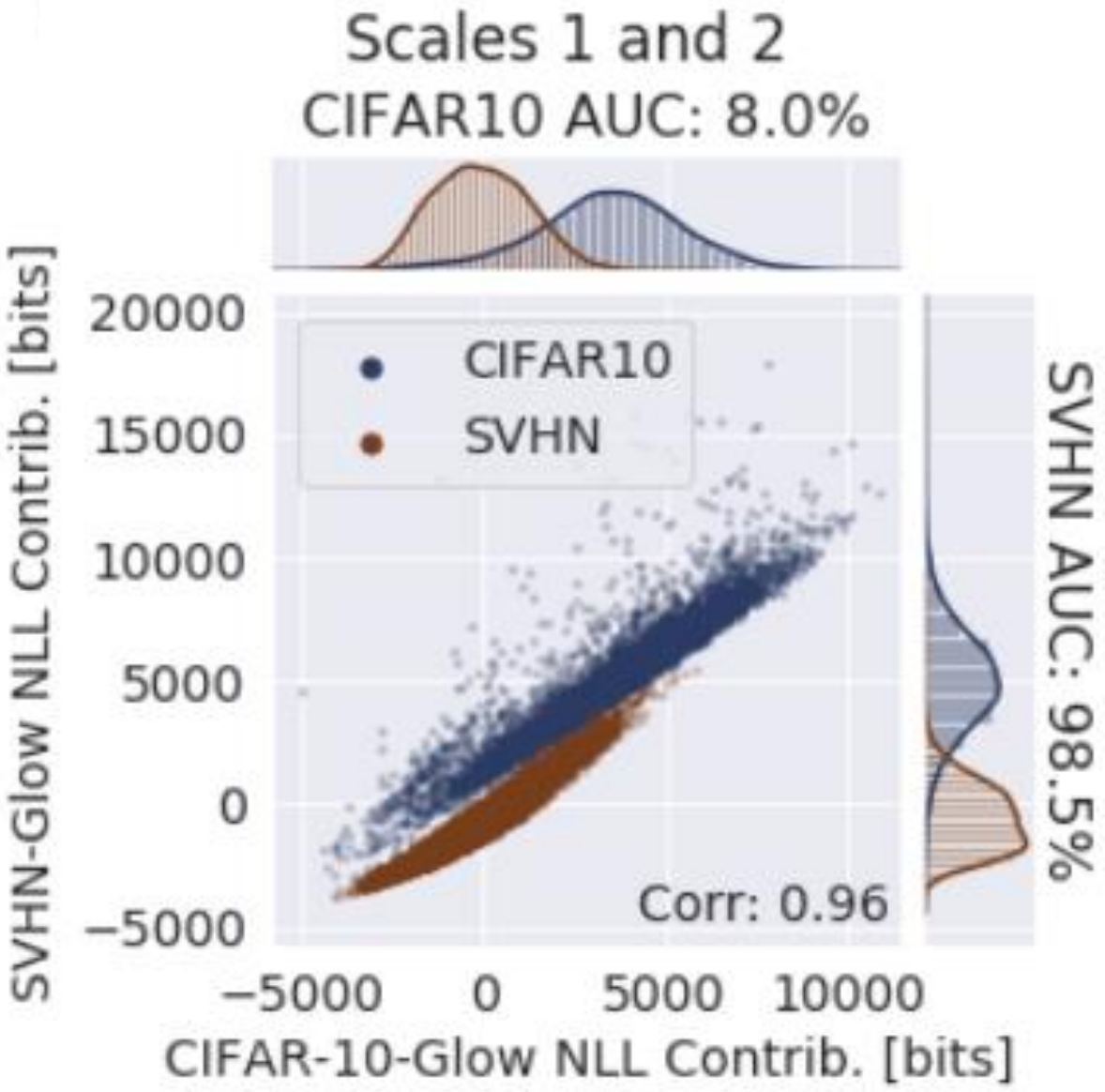
Glow Log Likelihood can be decomposed

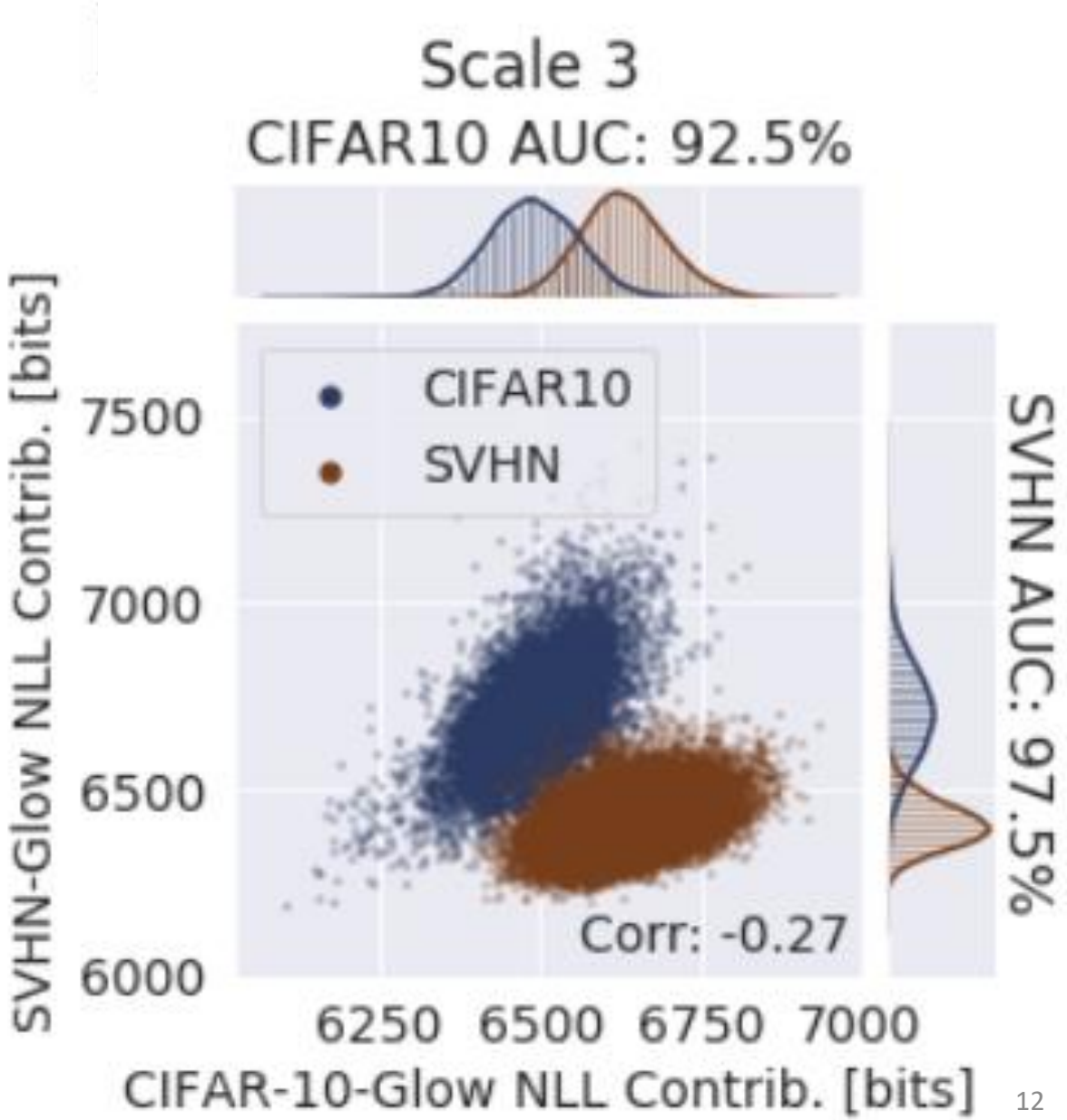


Log Likelihood Decomposition

$$\log p(x) = \sum_i c_i(x) = \sum_i \log p_z(z_i) + \log \left| det \left(\frac{\partial y_i}{\partial h_{i-1}} \right) \right|$$

Hierarchy of Features





Results

LLR Method works best for matching models

	Setting	Glow (in-dist) diff to:				PCNN (in-dist) diff to:			
In-dist	Out-dist	None	PNG	Tiny-Glow	Tiny-PCNN	None	PNG	Tiny-Glow	Tiny-PCNN
SVHN	CIFAR10	98.3	74.4	**100.0**	**100.0**	97.9	76.8	**100.0**	**100.0**
	CIFAR100	97.9	79.5	**100.0**	**100.0**	97.4	81.3	**100.0**	**100.0**
	LSUN	99.6	96.8	**100.0**	**100.0**	99.4	98.1	**100.0**	**100.0**
CIFAR10	SVHN	8.8	75.4	**93.9**	16.6	12.6	82.3	**94.8**	94.4
	CIFAR100	51.7	57.3	**66.8**	53.4	51.7	57.1	57.5	**63.5**
	LSUN	69.3	83.6	**89.2**	16.8	74.8	87.6	**93.6**	92.9
	Mean	61.3	73.0	**85.7**	53.2	63.2	76.3	86.2	**87.0**

Last Scale Best for Raw

CIFAR10 In-distribution (AUC %)

Out-dist		Full	16x16	8x8	4x4
SVHN	Raw	8.8	7.0	13.5	92.9
	Diff	**93.9**	84.6	48.9	83.6
CIFAR100	Raw	51.7	50.7	53.5	60.0
	Diff	**66.8**	55.7	56.3	66.1
LSUN	Raw	69.3	70.3	56.5	82.8
	Diff	**89.2**	63.6	74.0	75.1

Strong Results without Class Labels

	Setting	No Class labels				With Class Labels			
In-dist	Out-dist	4x4	Diff	Diff+	OE	4x4	Diff	Diff+	OE
CIFAR10	SVHN	96.4	98.6	**99.0**	75.8	96.1	98.6	**99.1**	98.4
	CIFAR100	85.4	84.5	**86.8**	68.5	88.3	87.4	88.5	**93.3**
	LSUN	95.1	94.1	**95.8**	90.9	95.3	94.1	96.2	**97.6**
	Mean	92.3	92.4	**93.8**	78.4	93.3	93.4	94.6	**96.4**
CIFAR100	SVHN	84.5	82.2	**85.4**	-	**89.6**	88.6	89.4	86.9
	CIFAR10	61.9	59.8	**62.5**	-	67.0	64.9	65.3	**75.7**
	LSUN	84.6	82.4	**85.4**	-	85.7	84.3	**86.3**	83.4
	Mean	77.0	74.8	**77.8**	-	80.8	79.3	80.3	**82.0**
	Mean	84.7	83.6	**85.8**	-	87.0	86.3	87.5	**89.2**

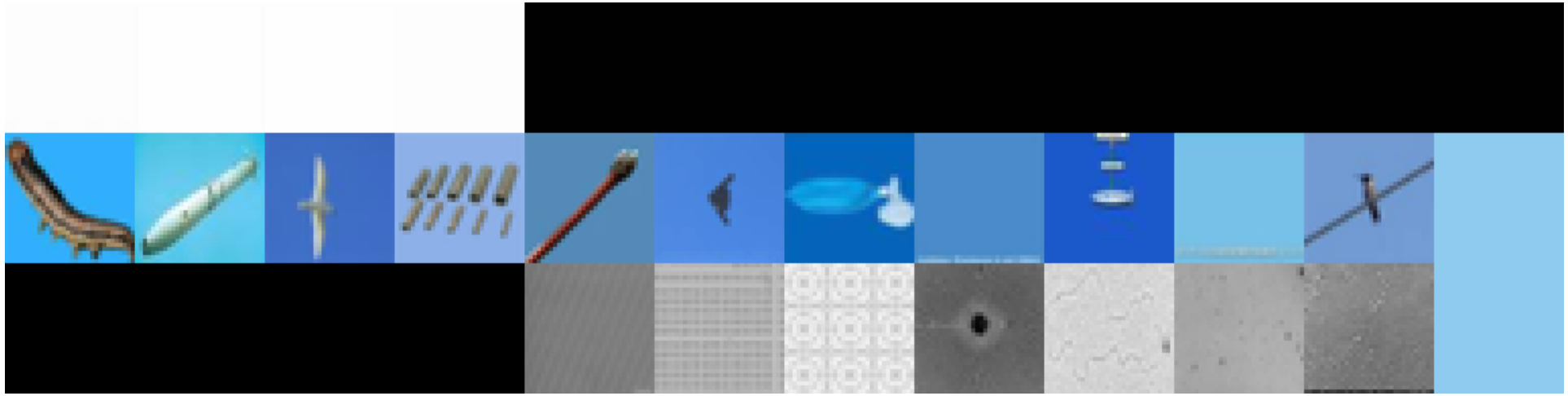
Different Metrics Rank Images Differently

CIFAR10 Glow

Raw Log Likelihood

Difference to Tiny-Glow

4x4 LL Contribution



Raw Log Likelihood

Difference to Tiny-Glow

4x4 LL Contribution

With outlier loss

Related Work

Likelihood Ratios for Out-of-Distribution Detection – Use in-distribution + semantic-destroying noise as general distribution

Input complexity and out-of-distribution detection with likelihood-based generative models – Use PNG as general distribution

BIVA: A Very Deep Hierarchy of Latent Variables for Generative Modeling – Investigate different scales of a hierarchical VAE for anomaly detection

Takeaways

- Low-level local features dominate likelihood
  - Hinders anomaly detection
- Likelihood difference of in-distribution model to general distribution model improves anomaly detection
- Likelihood contribution of last scale improves anomaly detection
- Open question how to best combine these approaches

Thank You

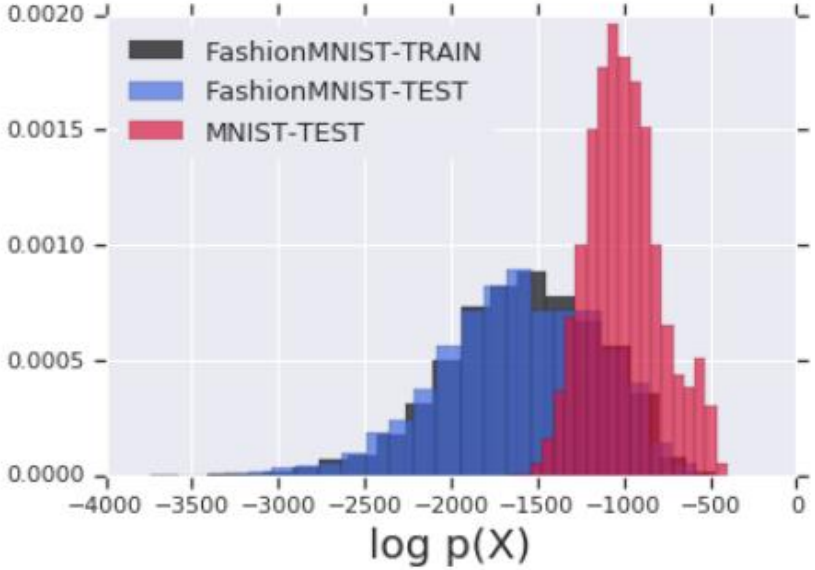
[Code at https://github.com/boschresearch/hierarchical_anomaly_detection](https://github.com/boschresearch/hierarchical_anomaly_detection)

Backup Slides

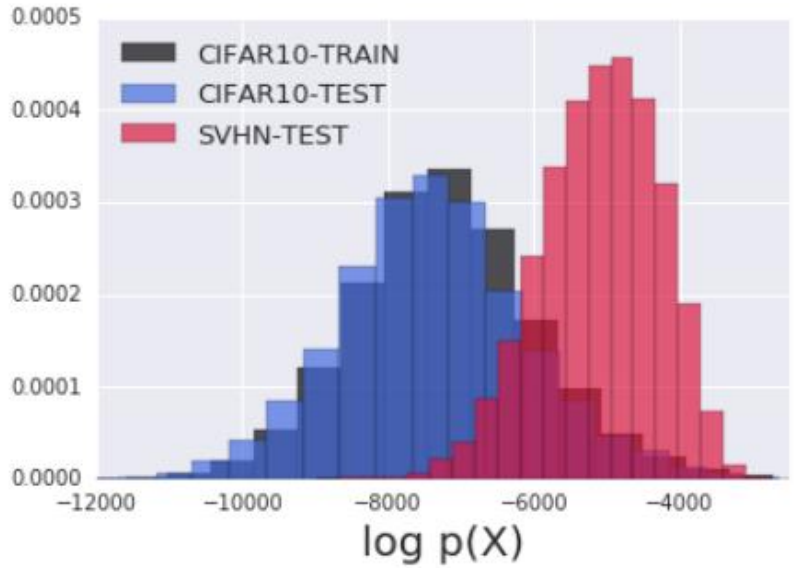
Other Related Work

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

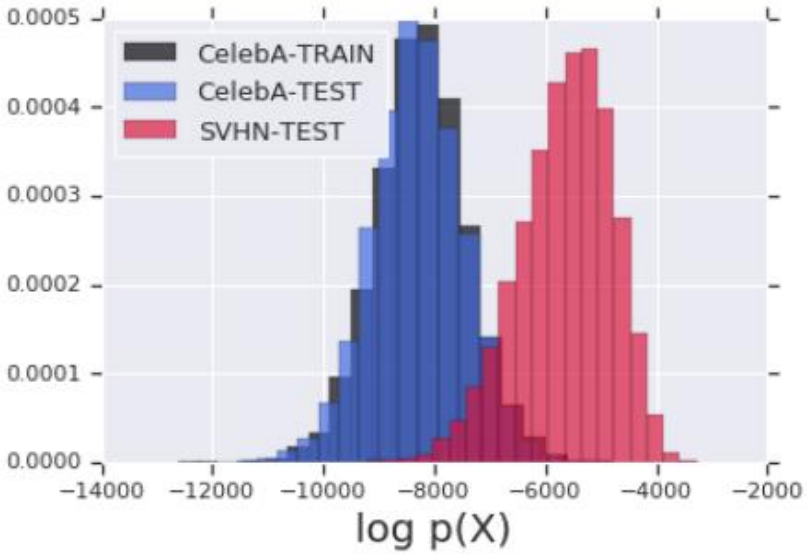
Eric Nalisnick[*†], **Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, Balaji Lakshminarayanan**[*]
DeepMind



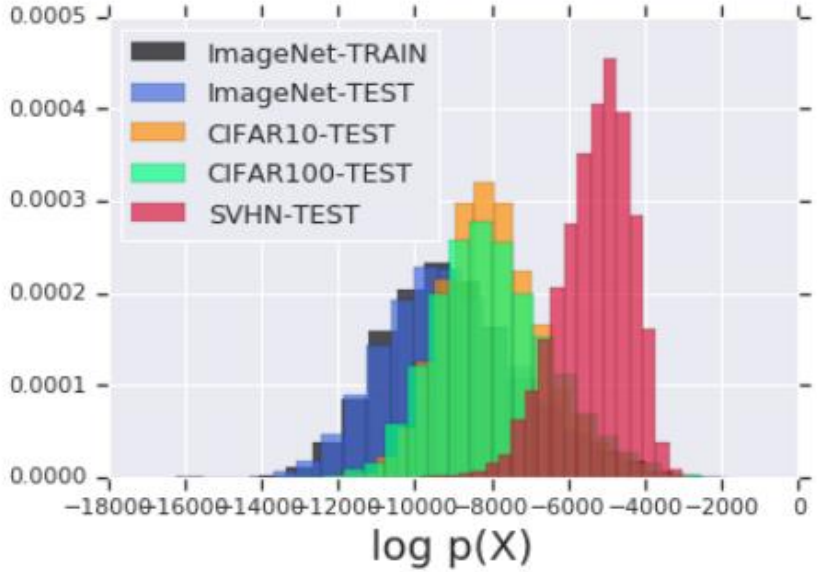
(a) Train on FashionMNIST, Test on MNIST



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



(c) Train on CelebA, Test on SVHN

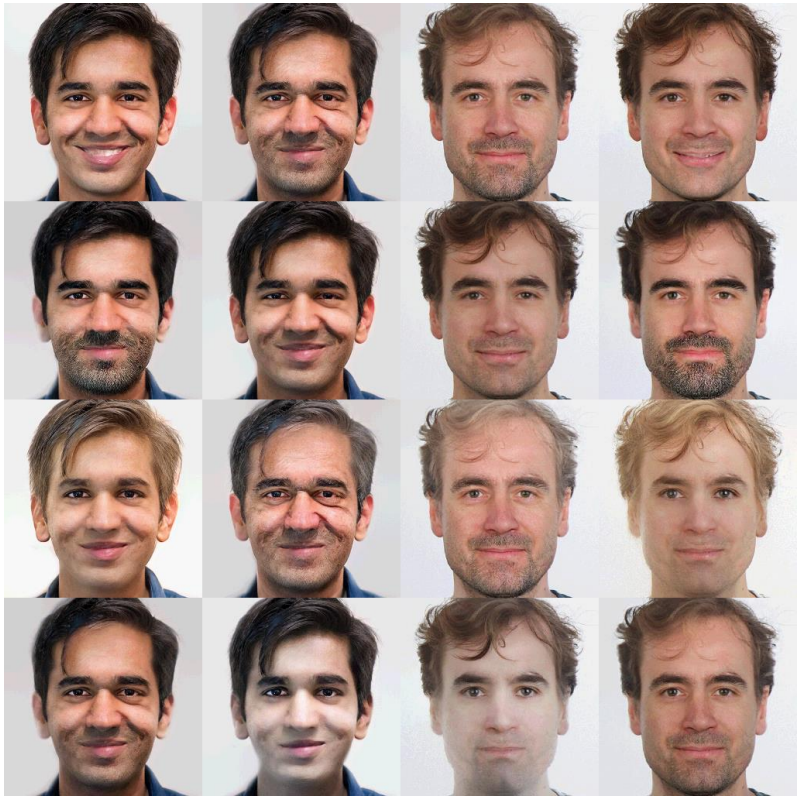


(d) Train on ImageNet,
Test on CIFAR-10 / CIFAR-100 / SVHN

Glow Model

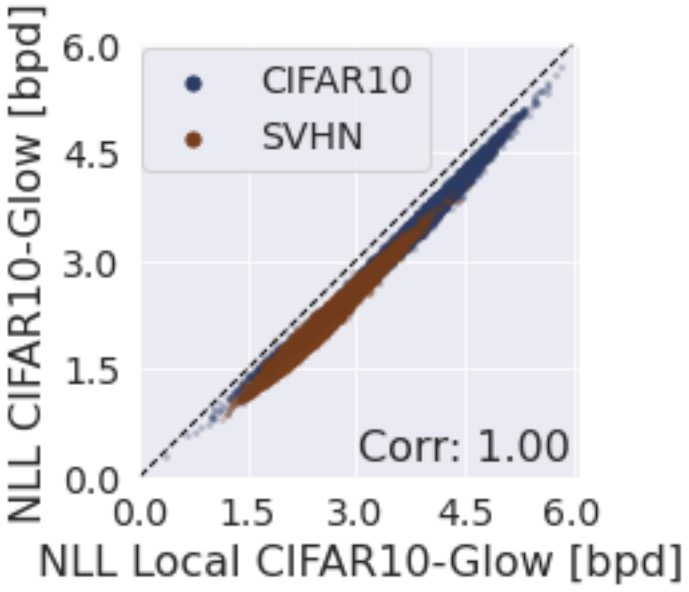
Glow: Generative Flow with Invertible 1×1 Convolutions

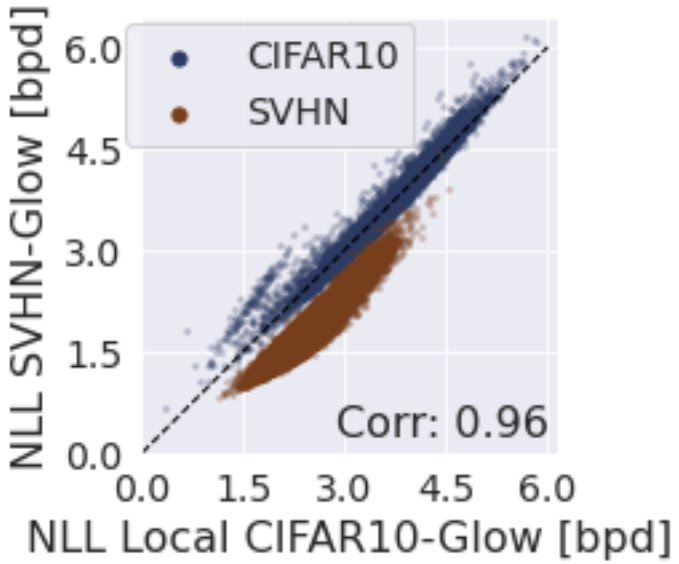
Diederik P. Kingma*, Prafulla Dhariwal*
OpenAI, San Francisco

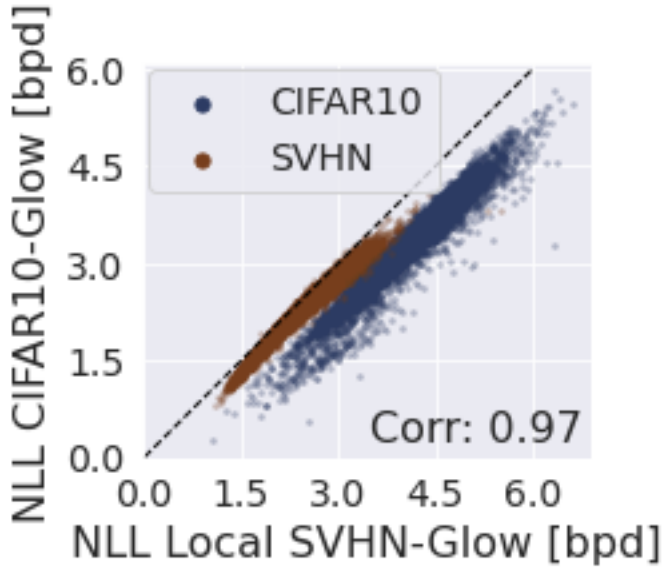


Extended Results

Common Local Features Dominate Model Likelihood







Rank Correlations between Image Likelihoods/Densities of Glow Models

Conv Glow Trained on:	Local Glow (CIFAR10)	Dense Glow (CIFAR10)	Conv Glow (CIFAR10)
CIFAR10	100%	86%	100%
SVHN	96%	90%	97%
TINY	100%	86%	100%

Fully Connected Models Less Correlated

Likelihoods Rank Correlation

Conv Glow Trained on:	Local Glow (CIFAR10)	Dense Glow (CIFAR10)	Conv Glow (CIFAR10)
CIFAR10	100%	86%	100%
SVHN	96%	90%	97%
TINY	100%	86%	100%

Dense Glow improves Fashion-MNIST vs. MNIST AUC from 15% to 81%!
Unfortunately, does not work as well for CIFAR10

More results

	Setting	Glow (in-dist) diff to:				PCNN (in-dist) diff to:			
In-dist	Out-dist	None	PNG	Tiny-Glow	Tiny-PCNN	None	PNG	Tiny-Glow	Tiny-PCNN
SVHN	CIFAR10	98.3	74.4	**100.0**	**100.0**	97.9	76.8	**100.0**	**100.0**
	CIFAR100	97.9	79.5	**100.0**	**100.0**	97.4	81.3	**100.0**	**100.0**
	LSUN	99.6	96.8	**100.0**	**100.0**	99.4	98.1	**100.0**	**100.0**
CIFAR10	SVHN	8.8	75.4	**93.9**	16.6	12.6	82.3	**94.8**	94.4
	CIFAR100	51.7	57.3	**66.8**	53.4	51.7	57.1	57.5	**63.5**
	LSUN	69.3	83.6	**89.2**	16.8	74.8	87.6	**93.6**	92.9
CIFAR100	SVHN	10.3	68.4	**87.4**	18.3	13.7	76.4	**91.3**	90.0
	CIFAR10	49.2	44.1	52.8	**54.2**	49.1	44.2	48.3	**54.5**
	LSUN	66.3	77.5	**81.0**	19.1	71.7	82.7	**90.0**	87.6
	Mean	61.3	73.0	**85.7**	53.2	63.2	76.3	86.2	**87.0**

Table S3: Anomaly detection performance for additional OOD datasets CelebA and Tiny-Imagenet. Conventions as in Table main manuscript.

	Setting	Unsupervised			Supervised		
In-dist	**Out-dist**	**Raw [4x4]**	**Diff**	**Diff†**	**Raw [4x4]**	**Diff**	**Diff†**
CIFAR10	CelebA	96.6 (1.2)	96.1 (1.2)	97.6 (0.5)	96.6 (1.9)	96.2 (2.1)	97.8 (1.0)
	Tiny-Imagenet	90.7 (0.9)	90.6 (0.7)	92.1 (0.4)	91.1 (0.8)	91.3 (0.9)	92.7 (0.4)
CIFAR100	CelebA	80.9 (1.3)	76.4 (2.4)	80.4 (1.1)	81.9 (5.4)	79.1 (7.2)	81.7 (4.7)
	Tiny-Imagenet	77.3 (0.5)	77.5 (0.5)	79.7 (0.3)	79.5 (0.4)	79.7 (0.5)	80.6 (0.5)

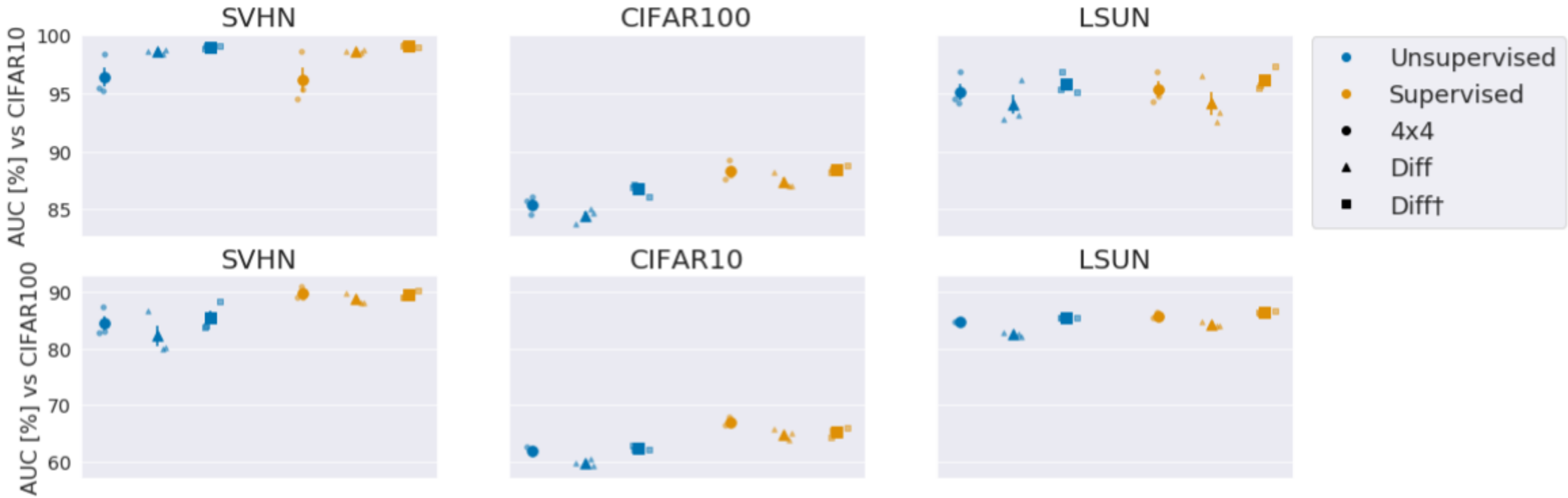


Figure S5: Graphical Overview over Anomaly Detection Results. Markers indicate mean result over three seeds, error bars indicate standard error of that mean. Type of marker indicates type of anomaly metric (defined as before and as in Table S2). Color indicates supervised or unsupervised setting. Rows are in-distribution dataset and columns are OOD datasets. Supervised setting outperforms unsupervised setting, especially on CIFAR10 vs. CIFAR100 and vice versa. Using a general-distribution model trained with outlier loss on the in-distribution (Diff†) always outperforms general-distribution model trained without outlier loss (Diff). Relative performance of final-scale method (4×4) compared with log-likelihood-difference methods (Diff and Diff†) varies between dataset pairs.

Further Results

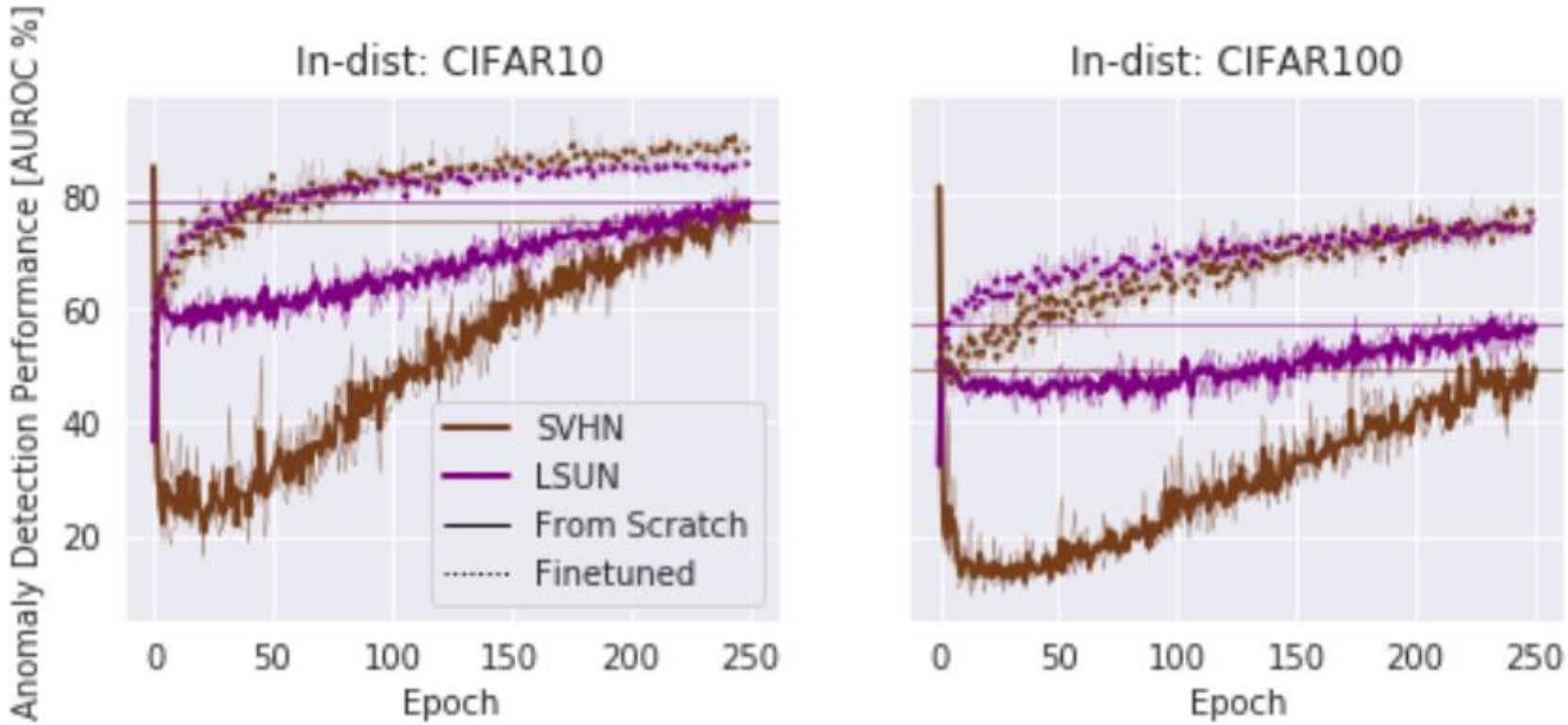
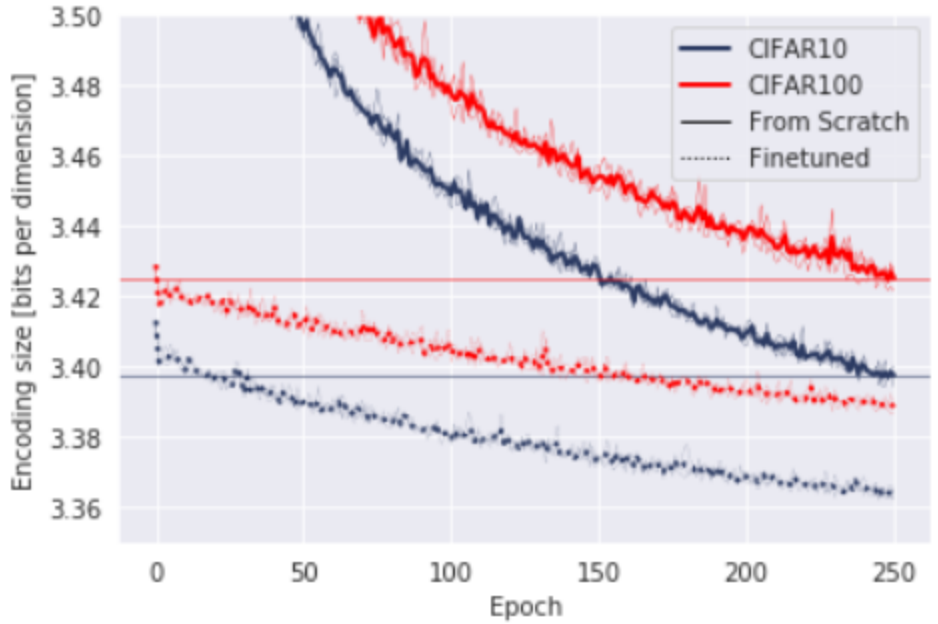
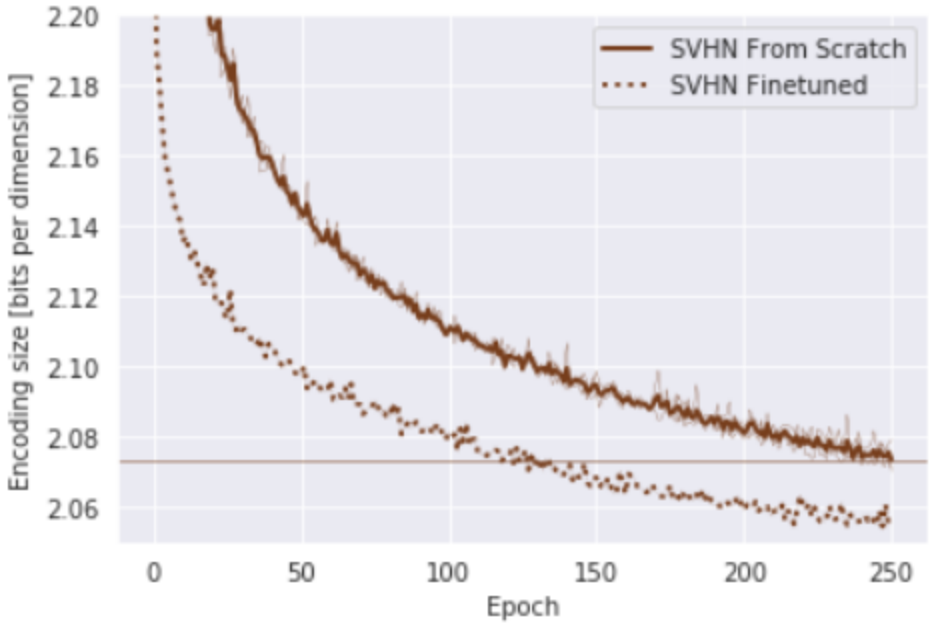


Figure S2: Training Curves Anomaly Detection From Scratch vs. Finetuned. Conventions as in Fig. S3. Glow networks are trained without any outlier loss. AUROC refers to AUROC computed from our log-likelihood ratio metric using another Glow-network trained on 80 Million Tiny Images. Note that the finetuned Glow networks outperform the final from-scratch trained Glow networks after less than 20% of the training epochs. Note that due to different evaluation (not noise-free) and different subsets used for intermediate results, results in this figures vary from final results in result tables.



CIFAR10/100



SVHN

Figure S3: Training Curves CIFAR10/100 and SVHN From Scratch vs. Finetuned. Transparent, thin lines indicate single-seed runs, solid, think lines indicate means over these runs. Solid horizontal lines indicate final mean performance of from-scratch trained models. Note that (i) finetuned Glow networks are better in each epoch; (ii) for CIFAR10/100 the finetuned Glow networks outperform the final from-scratch trained Glow networks after less than 20% of the training epochs and (iii) for SVHN, the finetuned Glow network outperforms the final from-scratch-trained Glow network after about 50% of the training epochs.

Table S1: Maximum likelihood performance in bits per dimension. Results obtained using single samples of uniform dequantization noise. Tiny is the Glow network trained on 80 Million Tiny Images. Retr refers to from-scratch training on the in-distribution dataset, Finet refers to finetuning aforementeioned Glow network trained on 80 Million Tiny Images. Note the original Glow paper [12] reached 3.35 bpd on CIFAR-10 with multi-GPU training. The Glow network and training setup we use is optimized for single-GPU training and not for maximum performance. The public implementation we originally based our implementation on (and uses the explicit conditioning step discussed in S1.3) reaches 3.39 bpd on CIFAR10.

In-dist	Tiny	Retr	Finet
SVHN	2.34	2.07	2.06
CIFAR10	3.41	3.40	3.36
CIFAR100	3.43	3.43	3.39

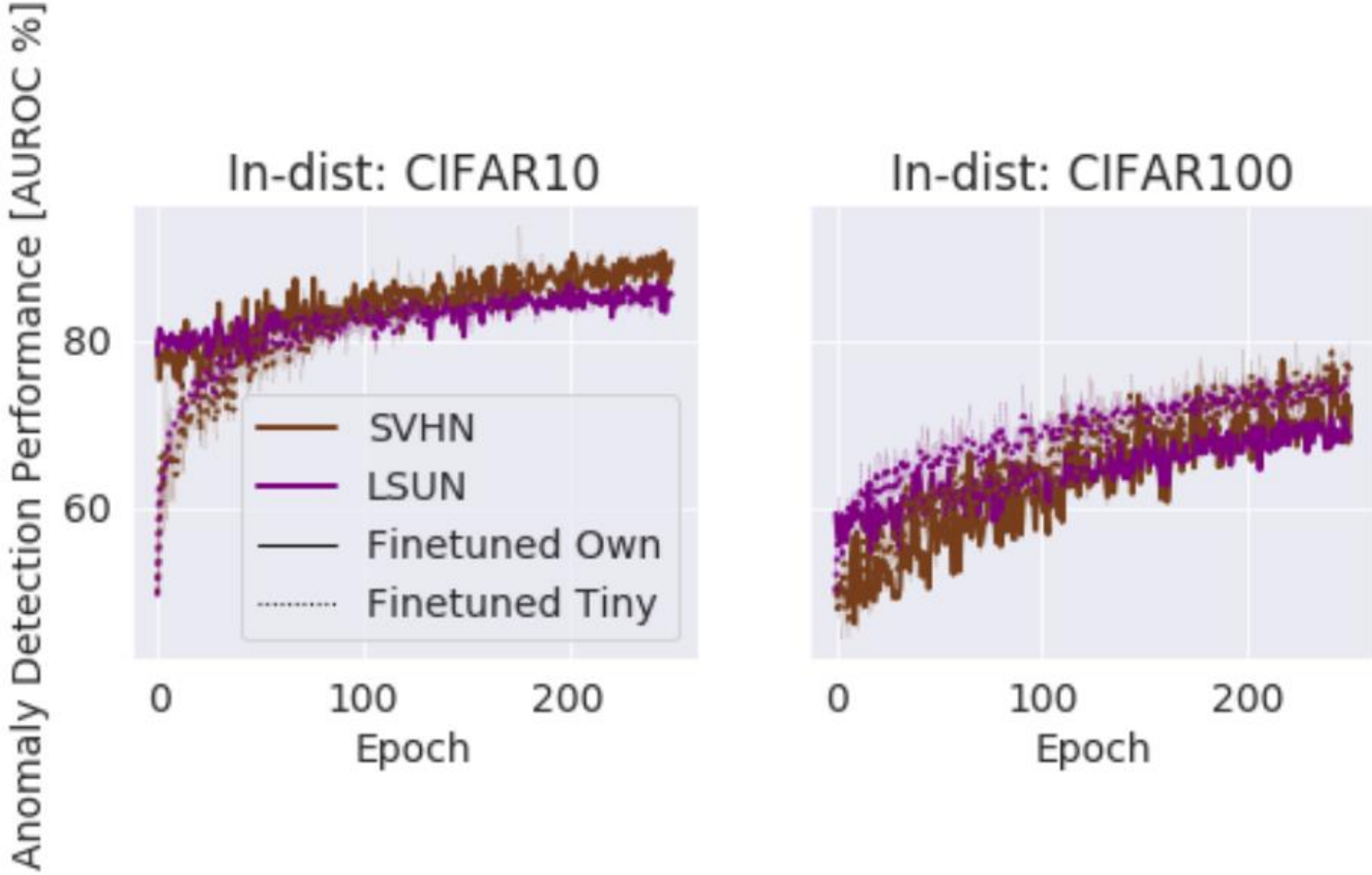


Figure S4: Training Curves Anomaly Detection Finetuned from Own model vs Finetuned from Tiny. Conventions as in Fig. S2. Finetuned from own model is the same as simply training twice as long on the in-distribution. At the end of training for CIFAR100, the model finetuned from Tiny still performs ~6% better on anomaly detection.

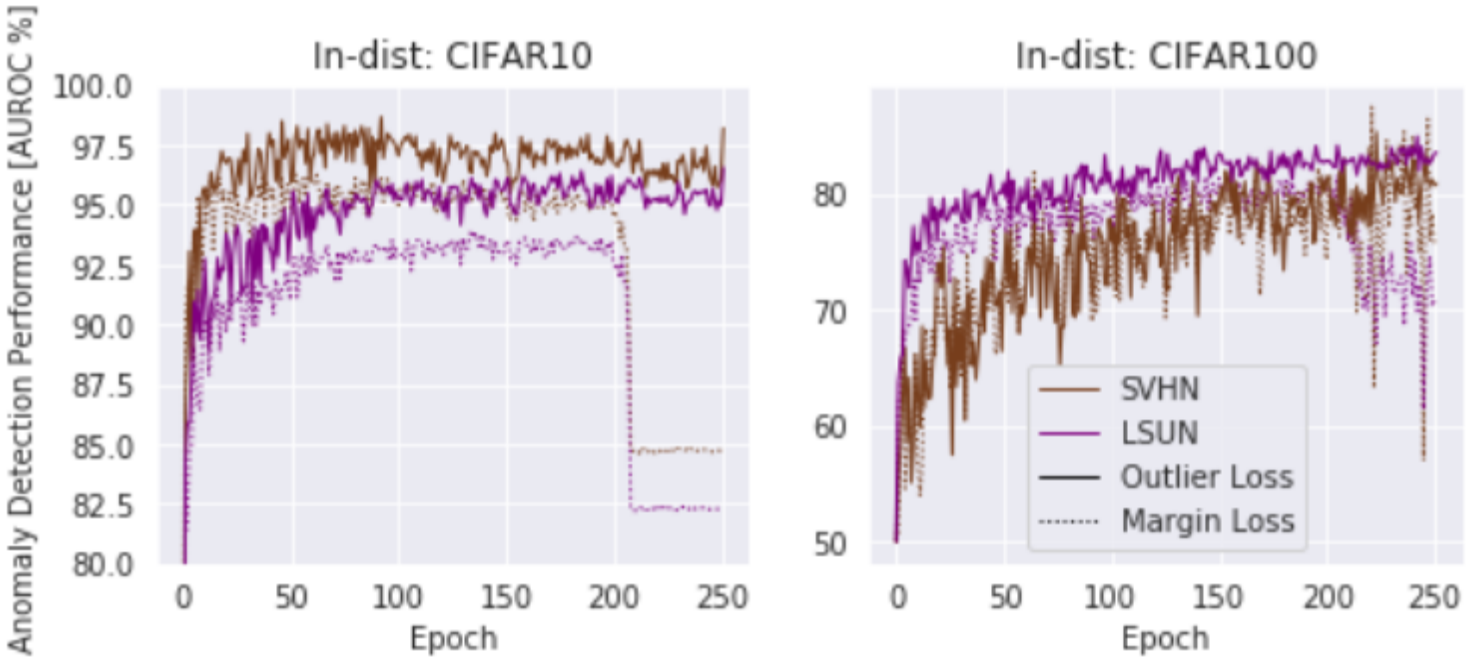


Figure S6: Training Curves Anomaly Detection Margin Loss vs Outlier Loss. AUROC refers to AUROC computed from our log-likelihood-difference metric using another Glow-network trained on 80 Million Tiny Images. Note Glow networks trained with margin loss experience substantial drops in anomaly detection performance in later stages of the training.

Table S4: Binary Classifier Anomaly Detection Results. Wide-ResNet classifier trained on 80 Million Tiny Images vs in-distribution as binary classification. We use $p_{ResNet}(y_{indist}|x)$ as our anomaly metric after training for the AUC computations.

In-dist	OOD	AUC
CIFAR-10	SVHN	93
	CIFAR-100	89
	LSUN	93
CIFAR-100	SVHN	73
	CIFAR-10	70
	LSUN	89
SVHN	CIFAR-10	100
	CIFAR-100	100
	LSUN	100

Amplitude Phase Analysis

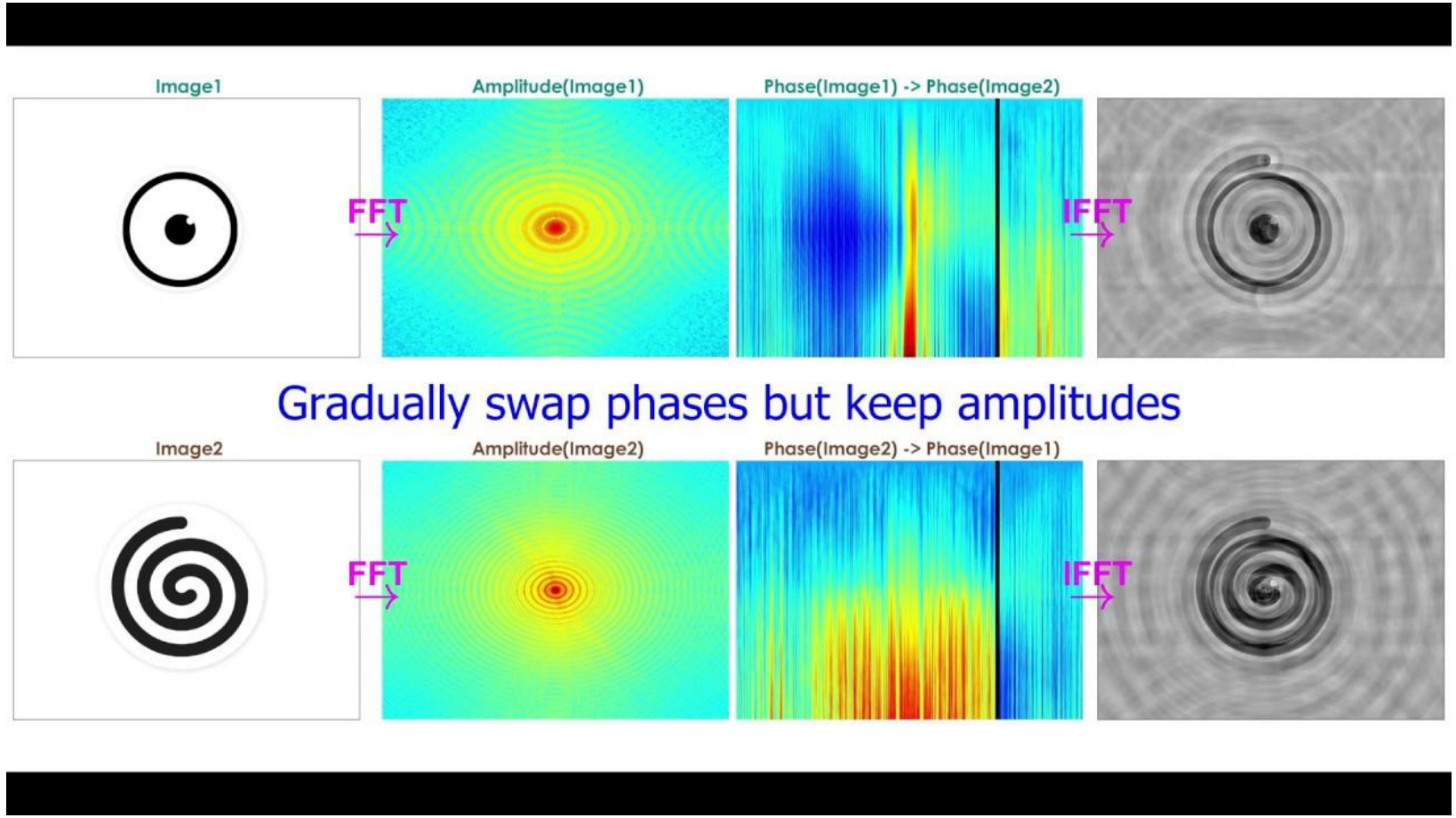


Image1
Amplitude(Image1)
Phase(Image1) -> Phase(Image2)
FFT
IFFT
Gradually swap phases but keep amplitudes
Image2
Amplitude(Image2)
Phase(Image2) -> Phase(Image1)
FFT
IFFT

Figure 4a shows the original image $a[m,n]$, Figure 4b the magnitude in a scaled form as $\log(|A(\Omega,\Psi)|)$, and Figure 4c the phase $\varphi(\Omega,\Psi)$.

Figure 4a

Original

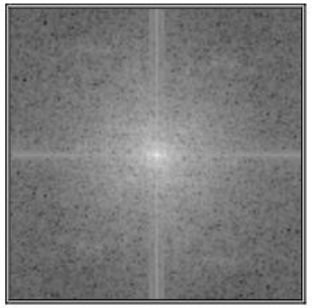


Figure 4b

$\log(|A(\Omega,\Psi)|)$

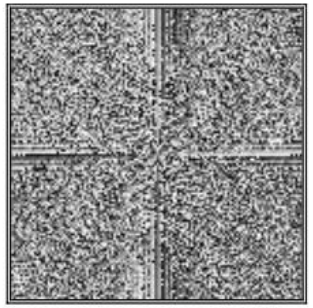


Figure 4c

$\varphi(\Omega,\Psi)$

Both the magnitude and the phase functions are necessary for the complete reconstruction of an image from its Fourier transform. Figure 5a shows what happens when Figure 4a is restored solely on the basis of the magnitude information and Figure 5b shows what happens when Figure 4a is restored solely on the basis of the phase information.

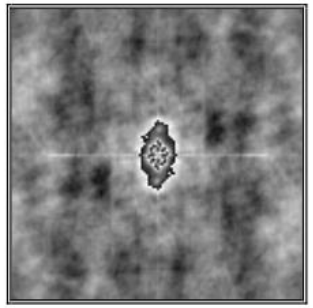


Figure 5a

$\varphi(\Omega,\Psi) = 0$

Figure 5b

$|A(\Omega,\Psi)| = constant$

Neither the magnitude information nor the phase information is sufficient to restore the image. The magnitude–only image (Figure 5a) is unrecognizable and has severe dynamic range problems. The phase-only image (Figure 5b) is barely recognizable, that is, severely degraded in quality.

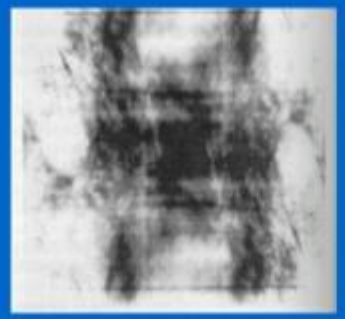
https://dsp.stackexchange.com/a/9092

Amplitude Dominates Likelihood

Corr > 0.8

Corr < 0.05

Magnitude and Phase of DFT (cont'd)



Reconstructed image using
magnitude only
(i.e., magnitude determines the
contribution of each component!)

Reconstructed image using
phase only
(i.e., phase determines
which components are present!)