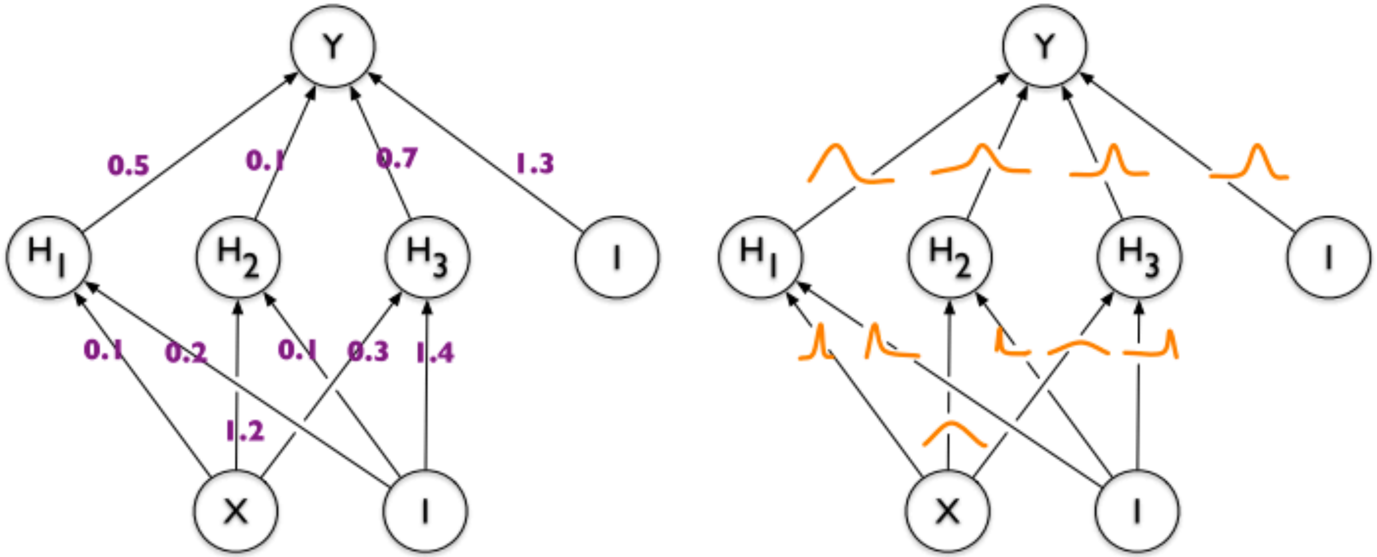
# Uncertainty in Bayesian Neural Nets

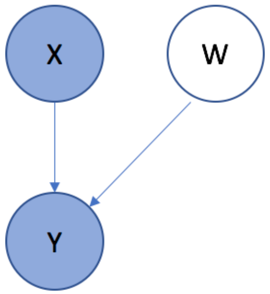
August 4 2017

# Overview

- BNN review
- Visualization experiments
- BNN results

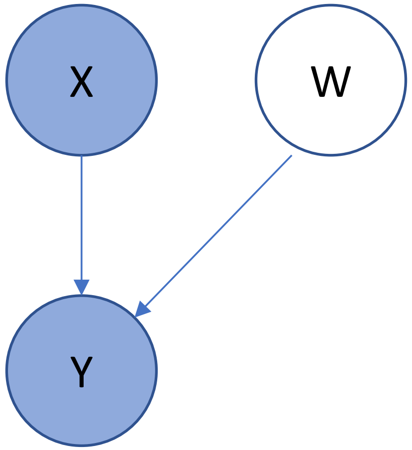
# BNN




Prior: p(W)
Likelihood: p(Y|X,W)
Approximate Posterior: q(W)
Posterior Predictive: $E_{q(W)}[p(y|x,W)]$

# BNN




- Variational Inference
  - Maximize lower bound on the marginal log-likelihood

$$\log p(Y|X) \geq E_{q(W)}[\log p(Y|X,W) + \log p(W) - \log q(W)]$$

Likelihood

Prior

Posterior Approx

Dependent on the number of data points

$$\frac{1}{M}\sum_{n=1}^{M} \log p(Y_n|X_n,W) + \frac{1}{N}\log\frac{p(W)}{q(W)}$$

# Different priors and posterior approximations

- Priors p(W):
  - $N(0, \sigma^2)$
  - Scale-mixtures of Normals
  - Sparsity Inducing

- Posterior Approximations q(W):
  - Delta peak $\mathrm{q(W)} = \delta W$
  - Fully Factorized Gaussians $\mathrm{q(W)} = \prod N(w_i | \mu_i, \sigma_i^2)$
  - Bernoulli Dropout
  - Gaussian Dropout
  - MNF

# Multiplicative Normalizing Flows (MNF)

Christos Louizos, Max Welling
ICML 2017

- Augment model with auxiliary variable

$$z \sim q(z) \quad W \sim q(W|z)$$

$$q(W) = \int q(W|z)q(z)dz$$

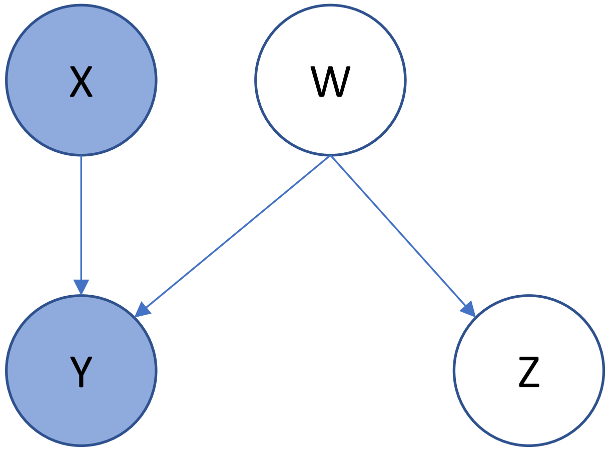
$$q(W|z) = \prod_{i=1}^{D_{in}} \prod_{j=1}^{D_{out}} N(z_i \mu_{ij}, \sigma_{ij}^2)$$

New lower bound

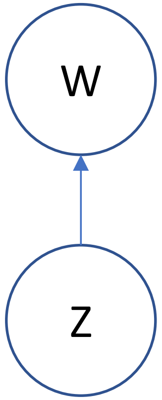
Normalizing Flows

$$\log p(Y|X) \geq E_{q(W)}[\log p(Y|X,W) + \log p(W) - \log q(W|z) + \log r(z|w) - \log q(z)]$$

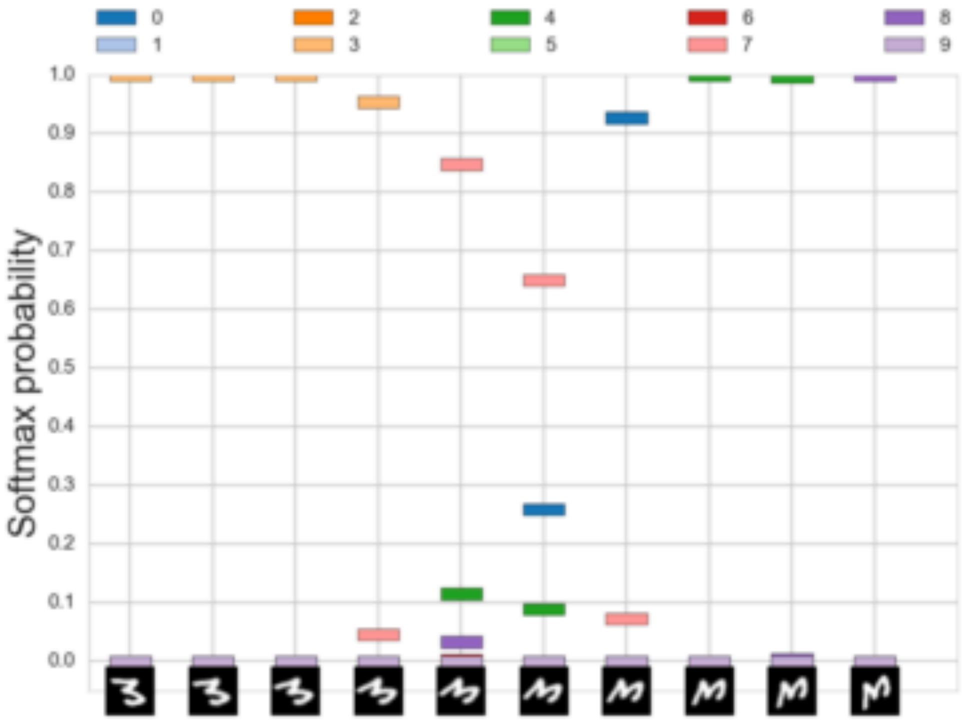
Generative Model



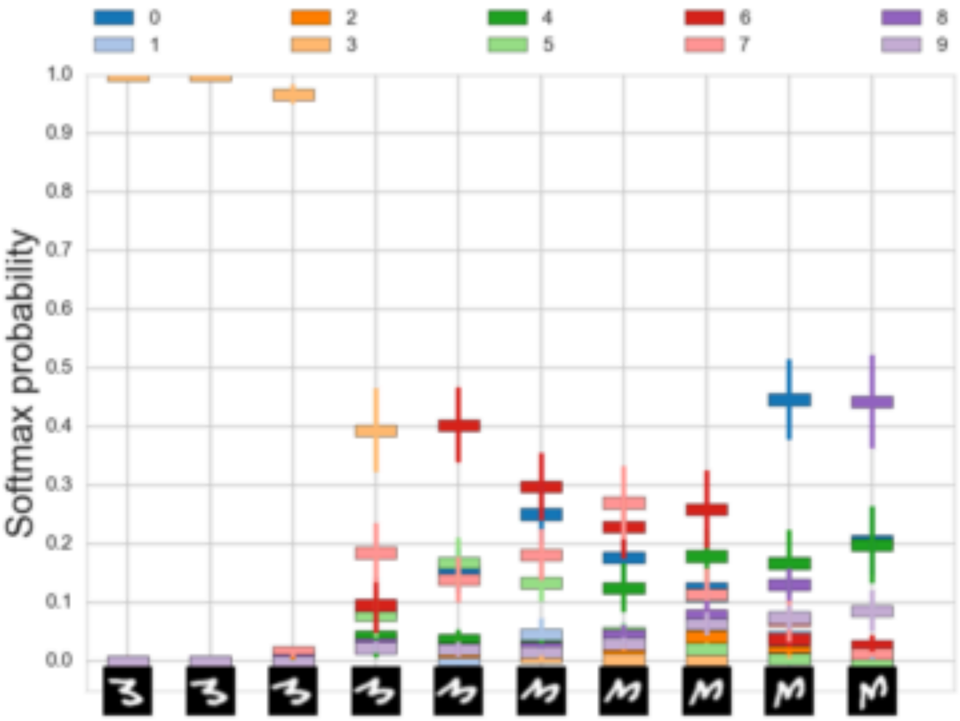

Inference Model



# Predictive Distributions




(a) LeNet with weight decay




(b) LeNet with multiplicative formalizing flows

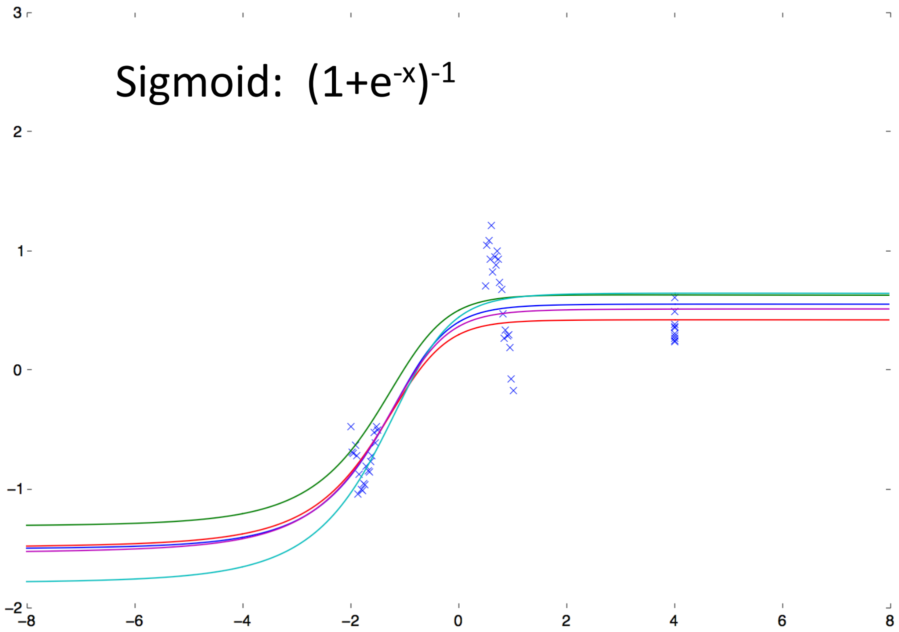
# Uncertainties

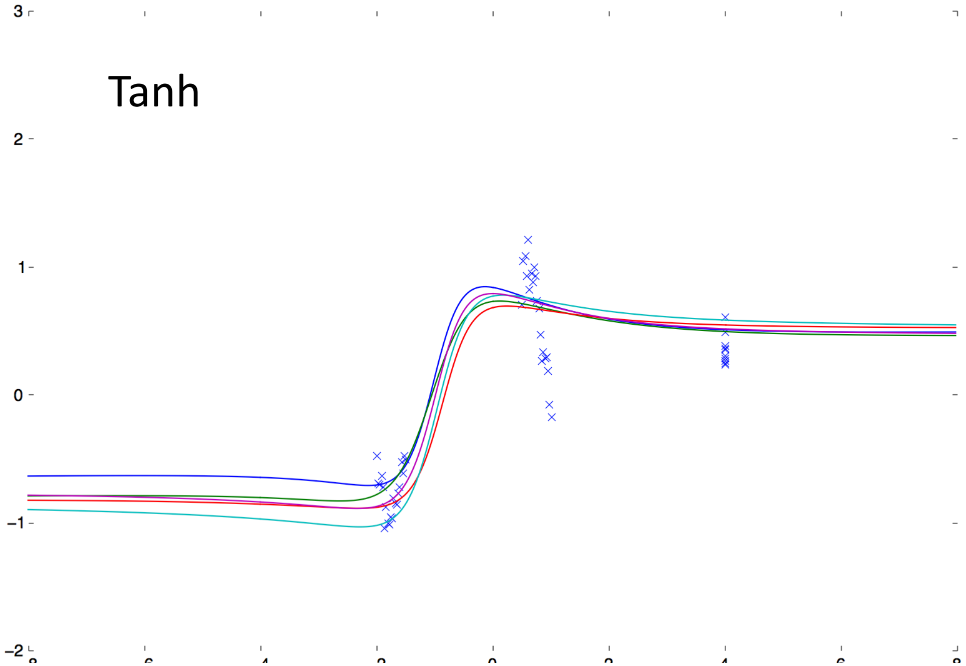
- Model uncertainty (Epistemic uncertainty)
  - Captures ignorance about the model that is most suitable to explain the data
  - Reduces as the amount of observed data increases
  - Summarized by generating function realizations from our distribution
- Measurement Noise (Aleatoric uncertainty)
  - Noise inherent in the environment, captured in likelihood function
- Predictive uncertainty
  - Entropy of prediction = $H[p(y|x)]$

# Visualization Experiments

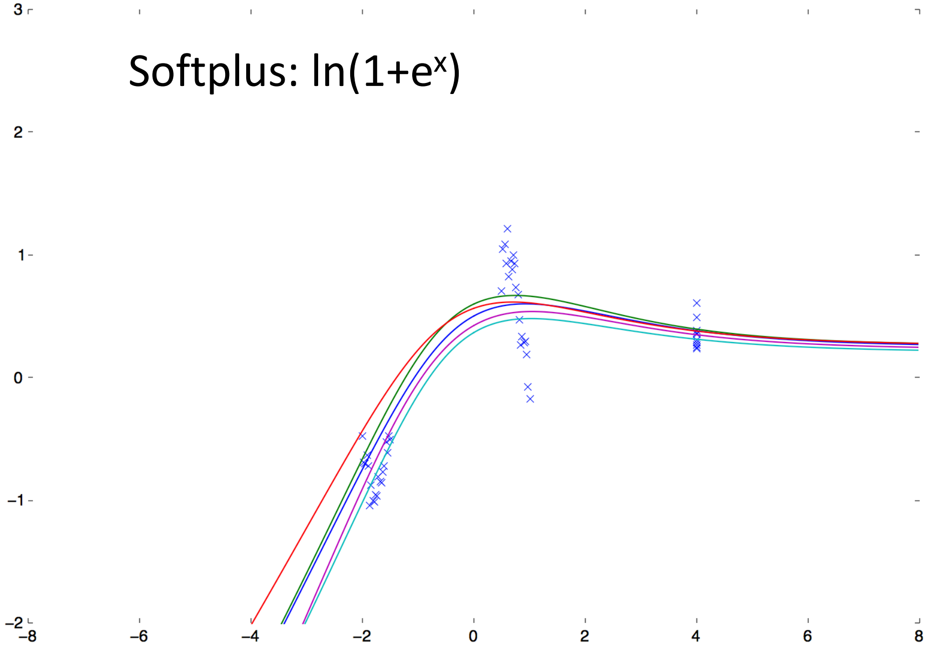
- 1D regression
- Classification of MNIST (visualize in 2D)

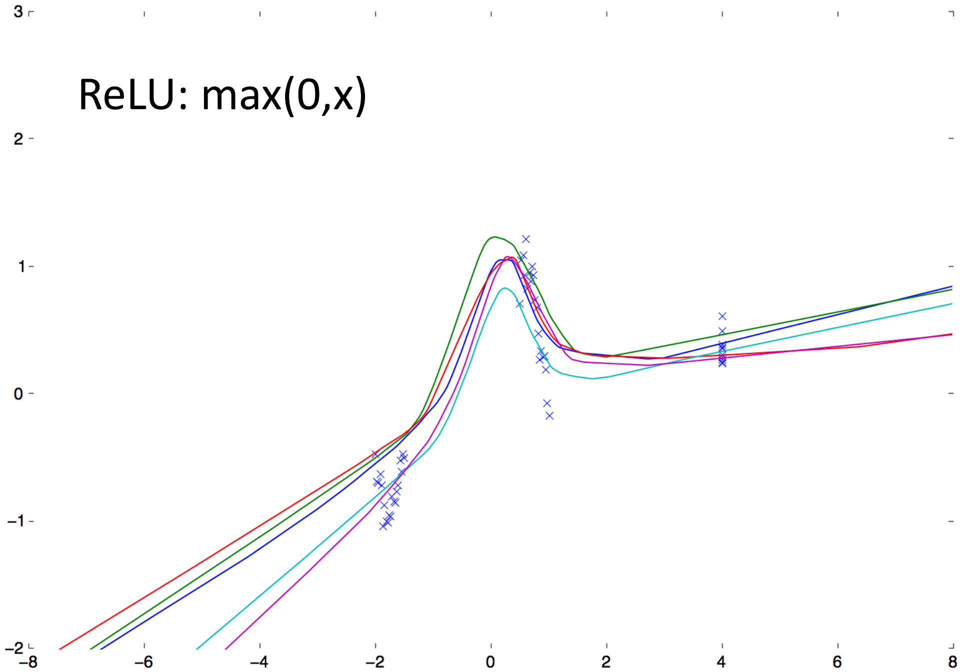
- Questions:
    - Activations
    - Number of samples
    - Held out classes
    - Type of uncertainties




# BNNs with Different Activation Functions



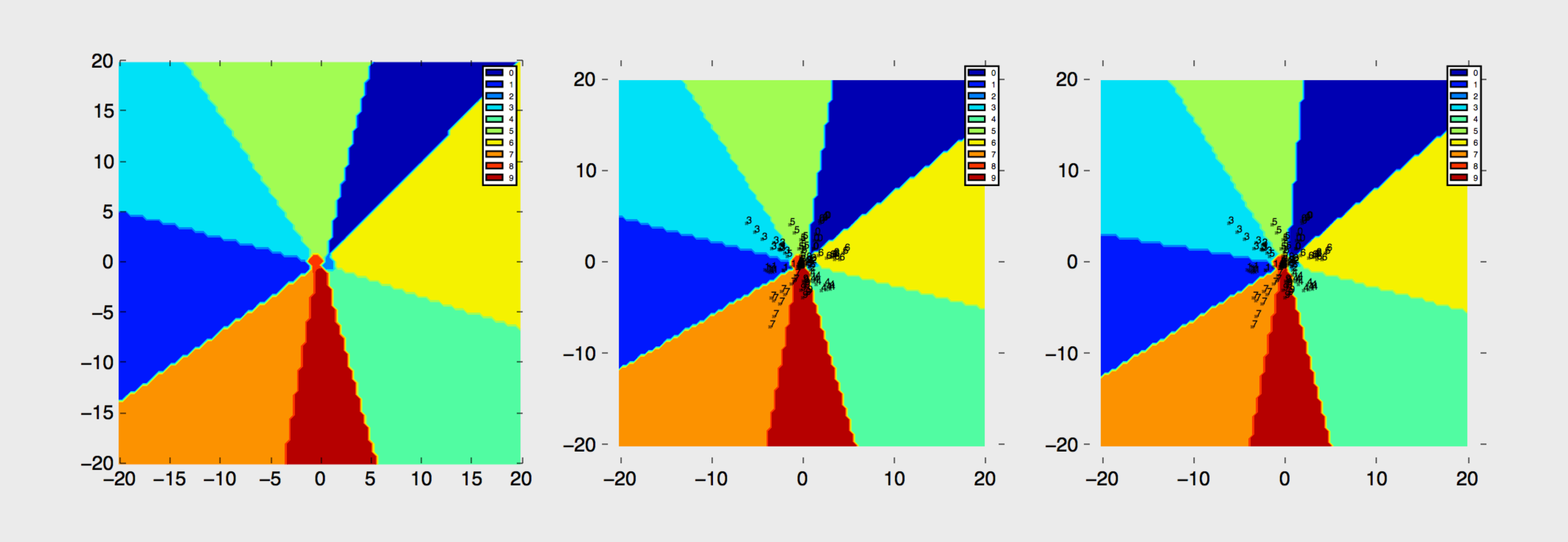
# Uncertainty of Decision Boundaries

- Setup:
  - Classification of MNIST
  - Train: 50000 Test: 10000

$$\underbrace{784\text{-}100\text{-}2}_{\text{NN}}\underbrace{\text{-}100\text{-}10}_{\text{BNN}}$$

BNN: FFG, N(0,1)
Activations: Softplus

# Decision Boundaries – 3 Samples



Plot of Argmax p(y|x) at each point

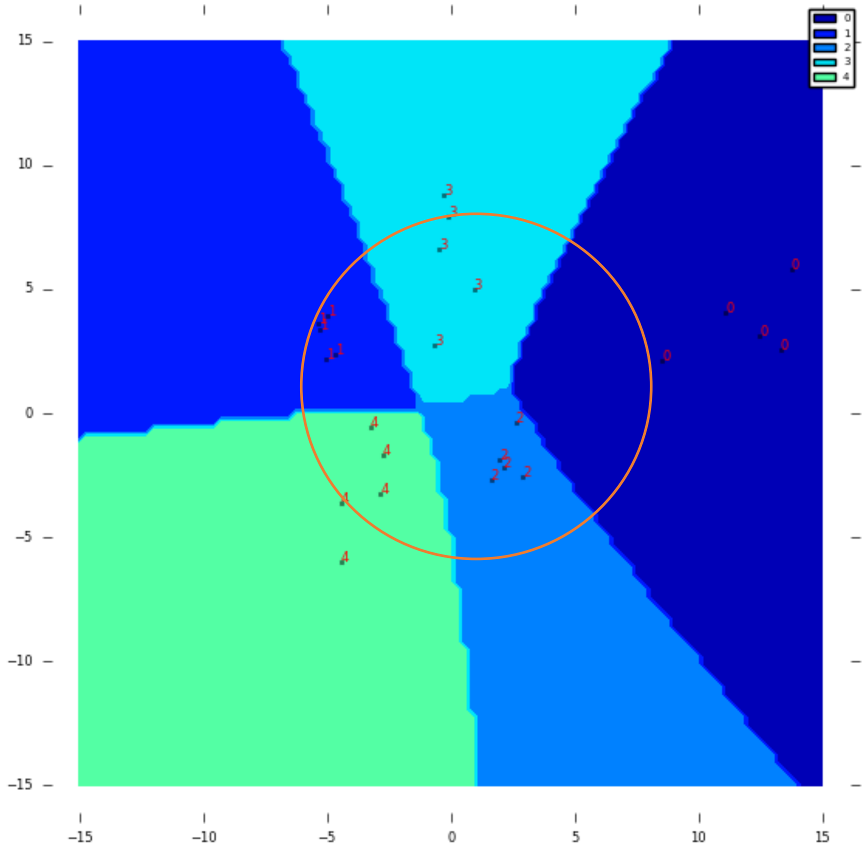
# Uncertainty of Decision Boundaries: Held Out Classes

- Setup:
  - Classification of digits 0 to 4 (5 to 9 held out)

$$\underbrace{784\text{-}100\text{-}100\text{-}2}_{\text{NN}}\text{-}\underbrace{100\text{-}100\text{-}10}_{\text{BNN}}$$

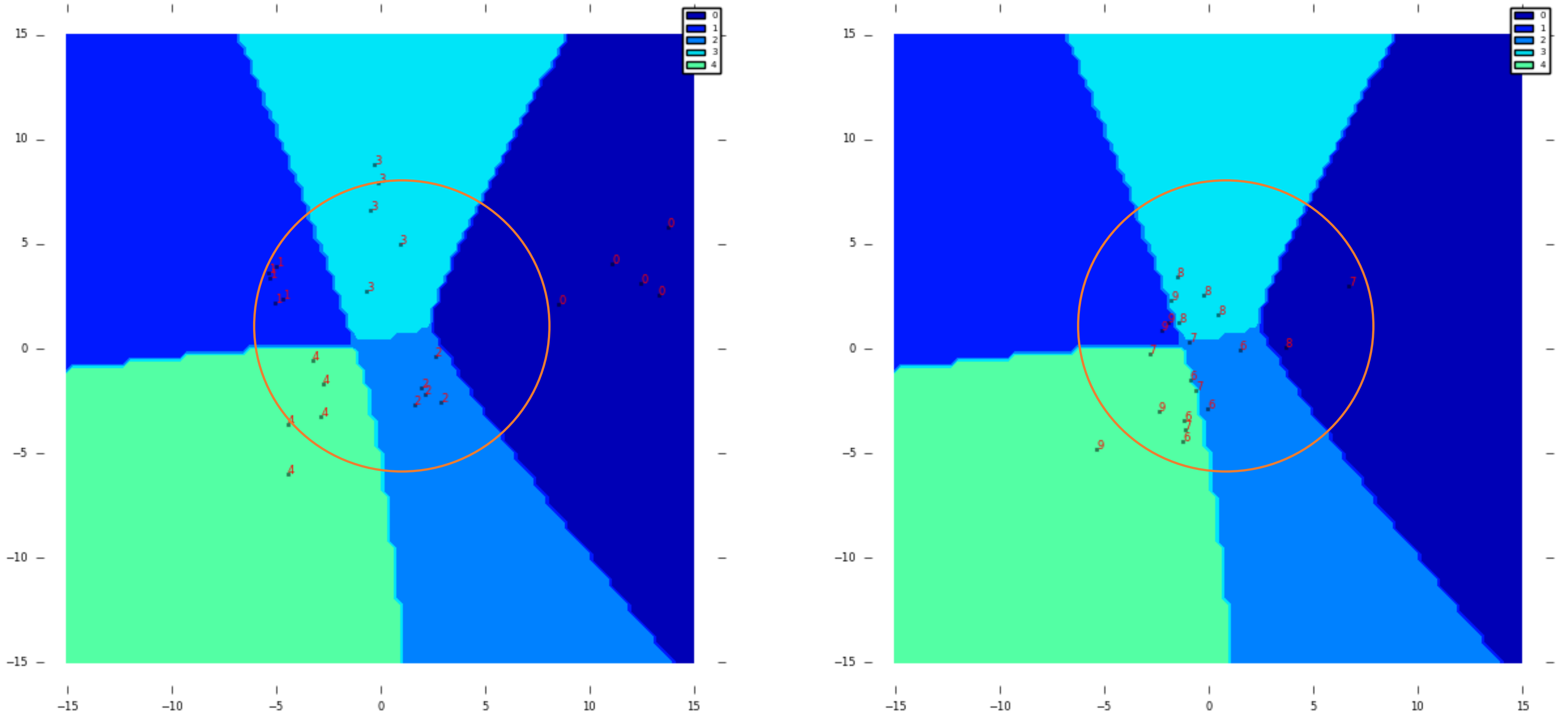
BNN: FFG, N(0,1)
Activations: Softplus

# Where do you think the held out classes will go?

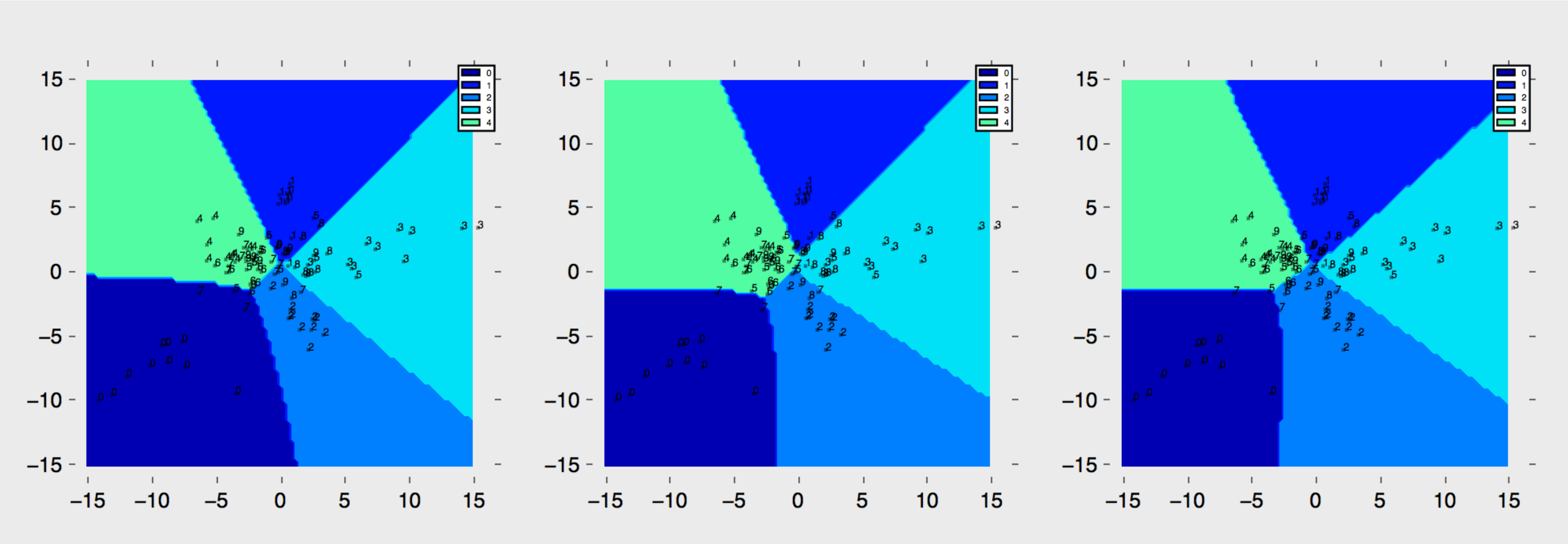



Inside or Outside the Circle?

# Where do you think the held out classes will go?



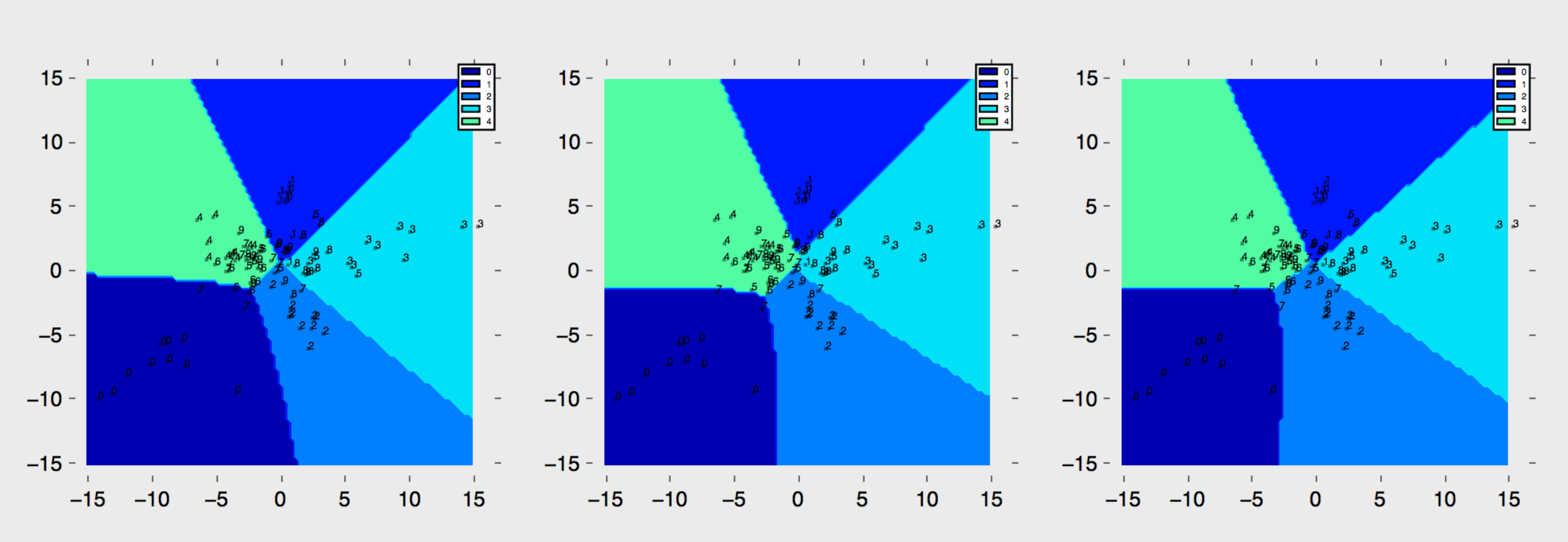
# Held Out Classes



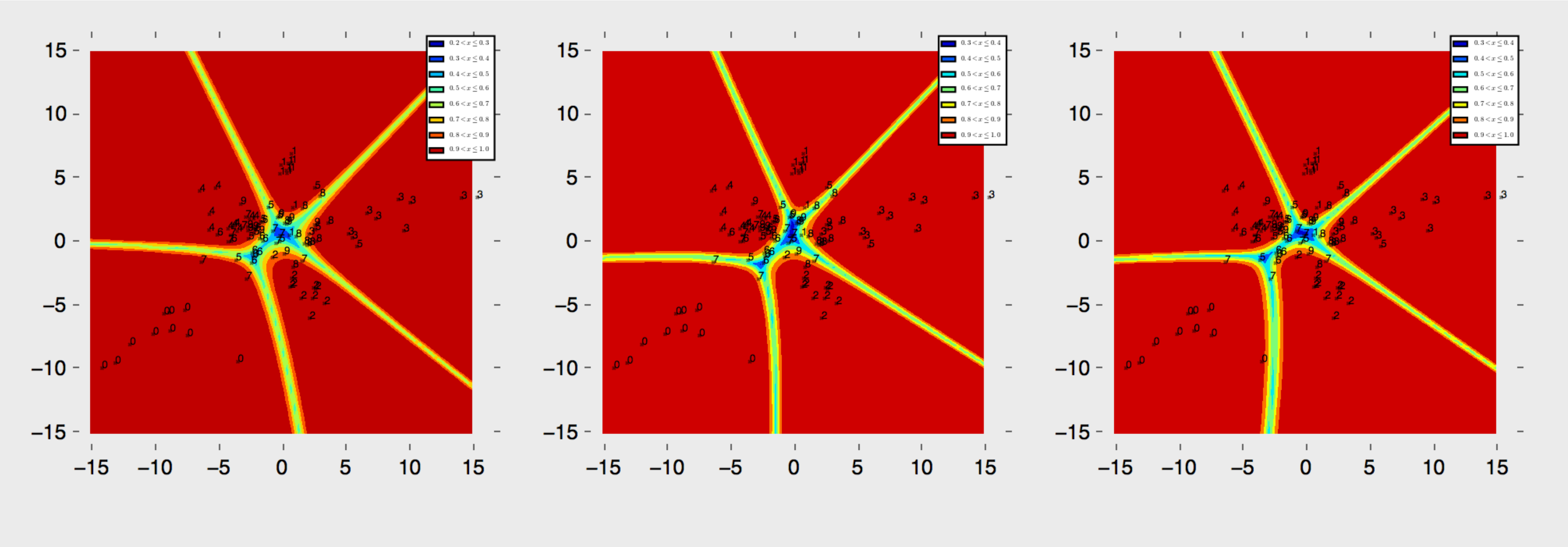

Unseen classes don’t get encoded as something far away, instead encoded near mean

# Confidence of Predictions?

Maybe large areas have high entropy
Argmax vs Max



# Class Boundaries - Confidences



Sharp transitions

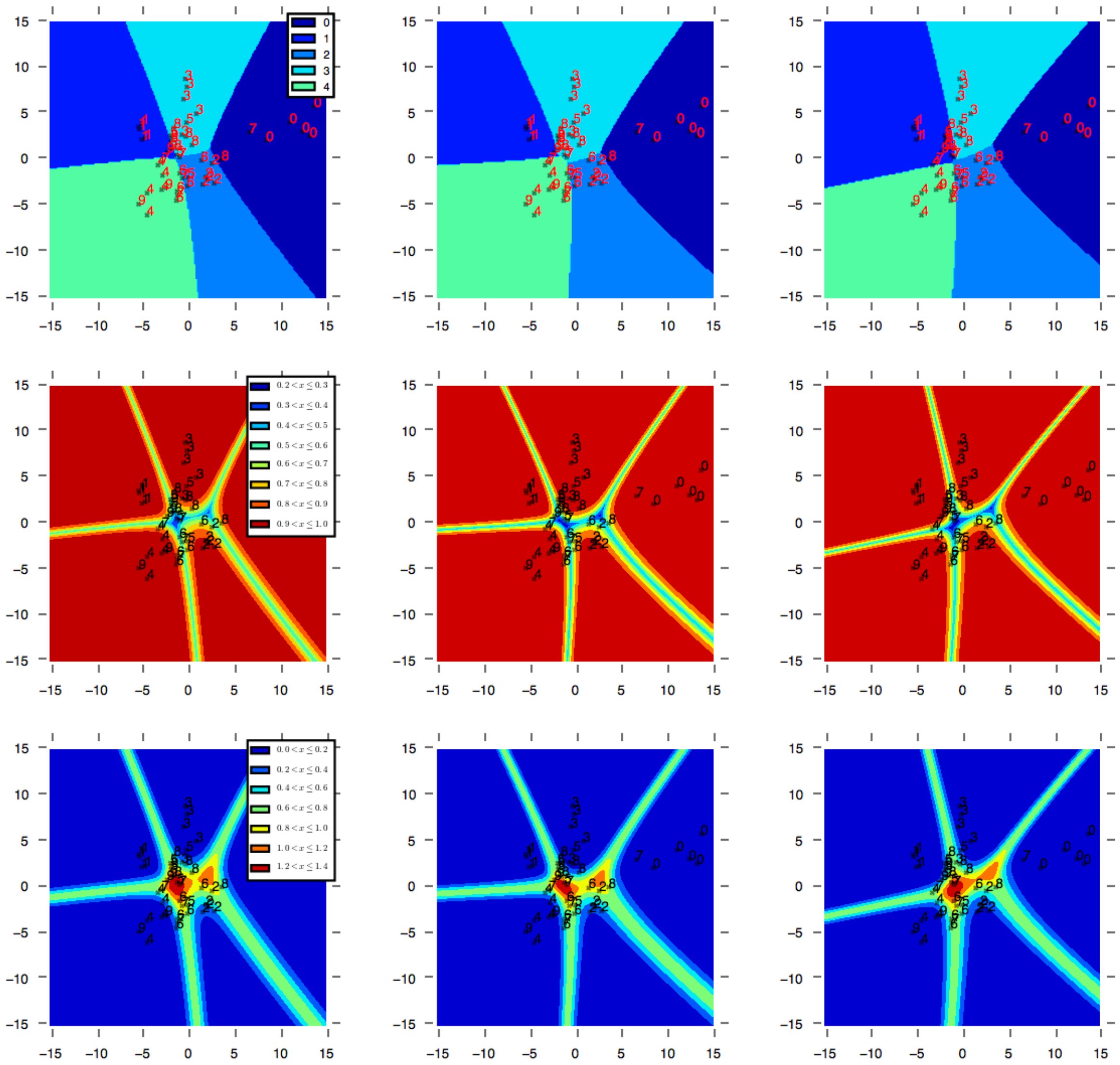
There isn’t much uncertain space: mostly uniform, high confidence

# Entropy

Argmax

Max

Entropy

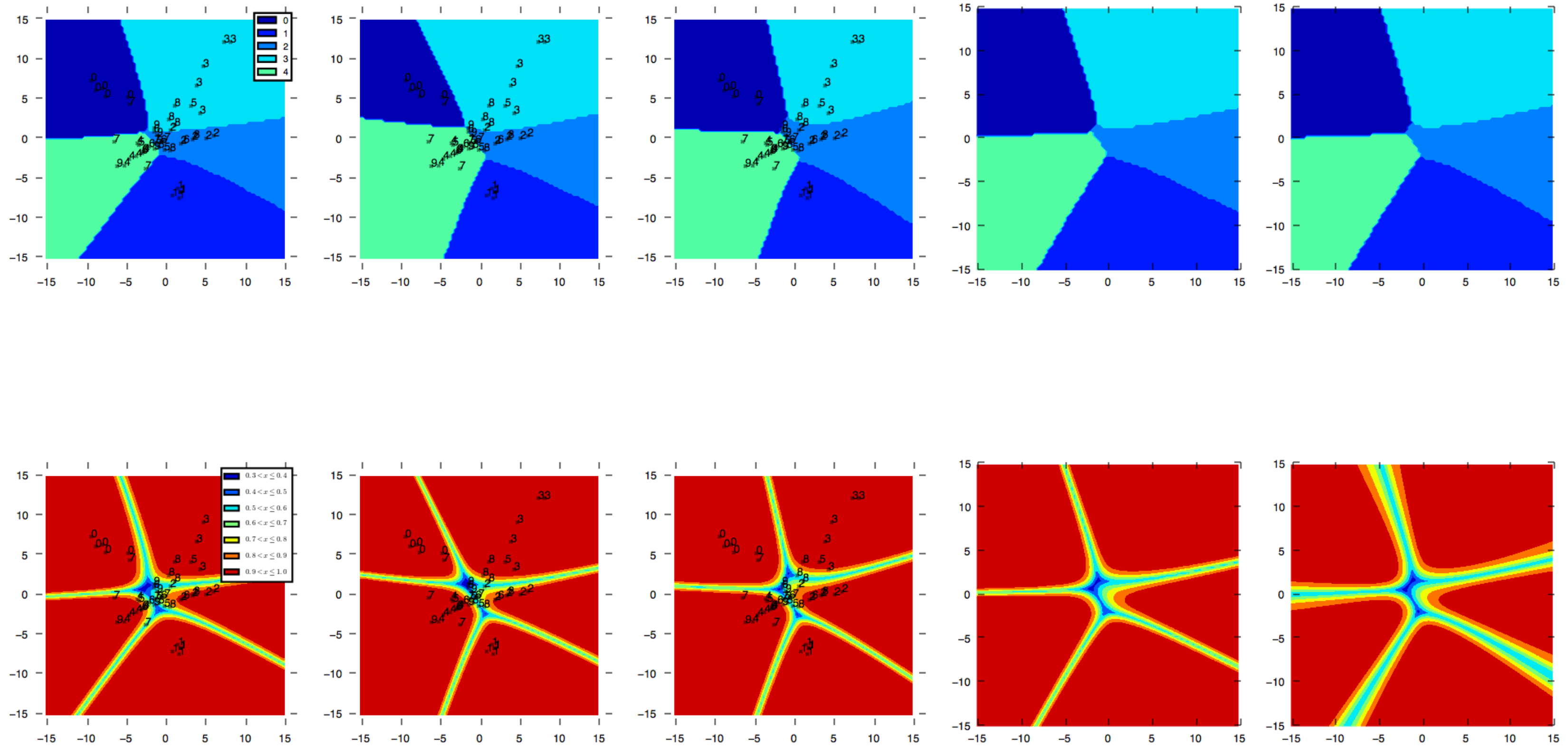


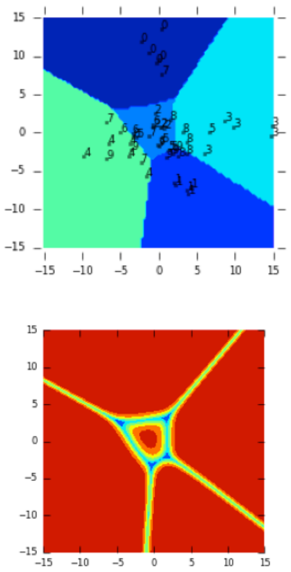
# Affect of Choice of Activation Function

- Softplus
- ReLU
- Tanh

# Softplus

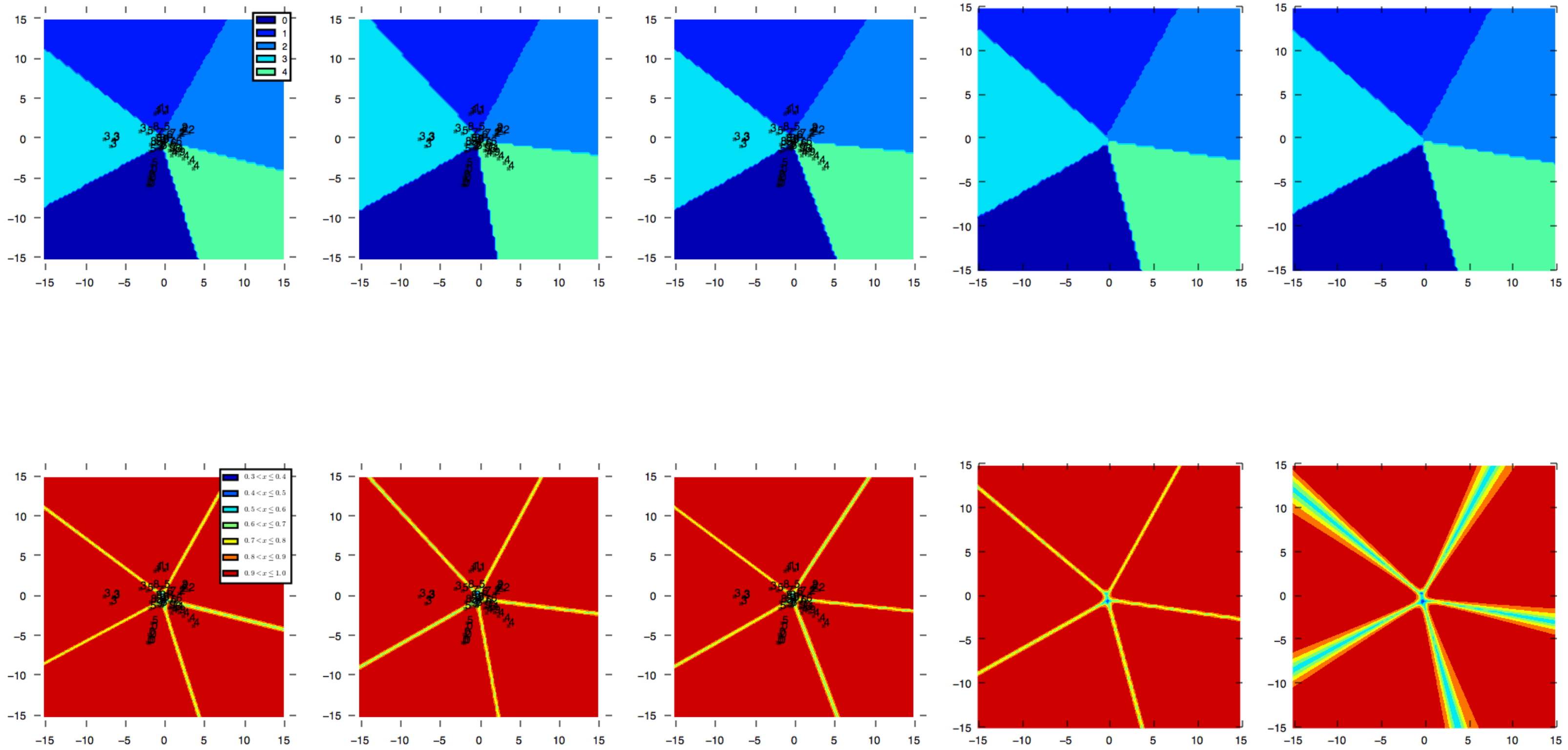
Sample 1 | Sample 2 | Sample 3 | Mean of q(W) | $E_{q(W)}[p(y|x,w)]$





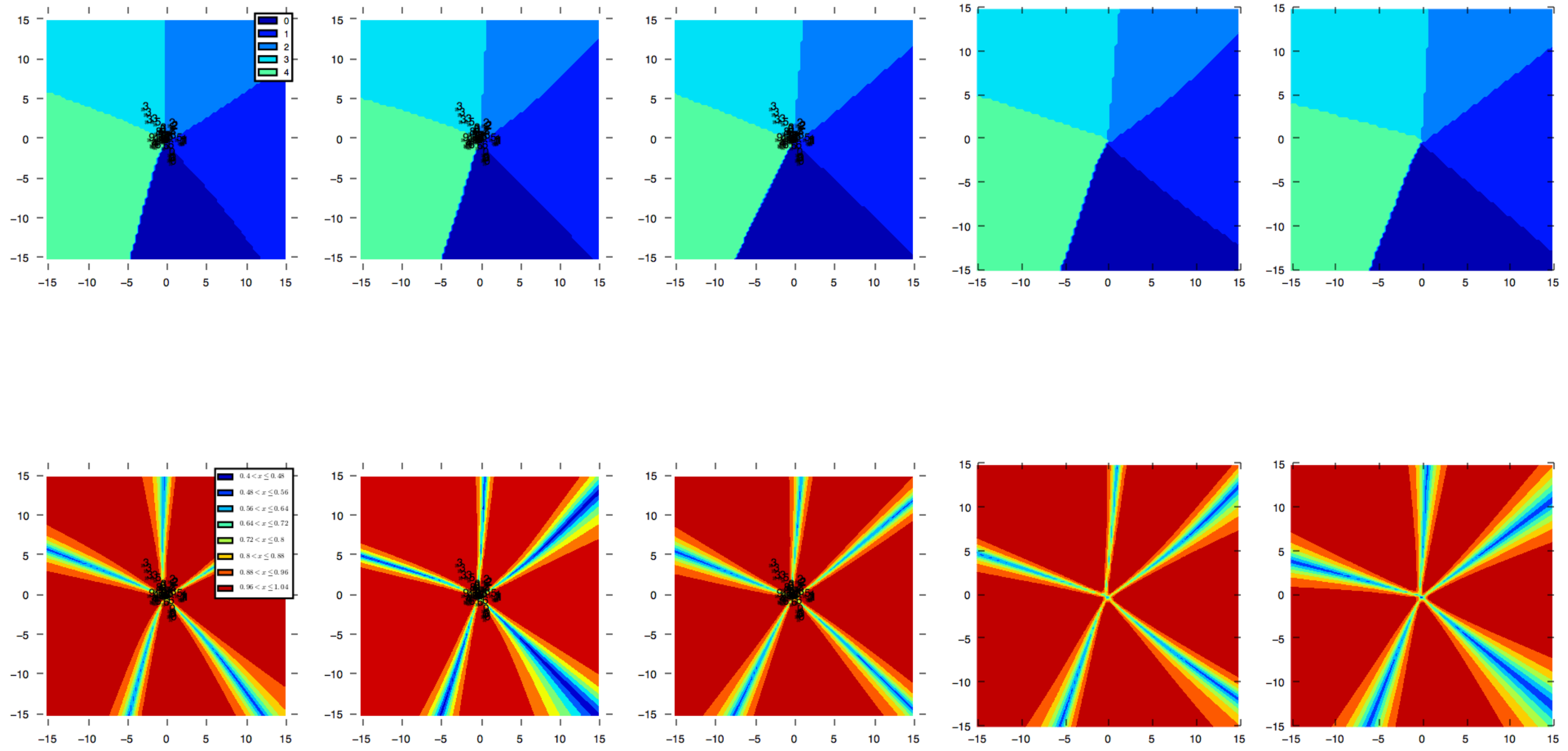
# ReLU

Sample 1 | Sample 2 | Sample 3 | Mean of q(W) | $E_{q(W)}[p(y|x,w)]$



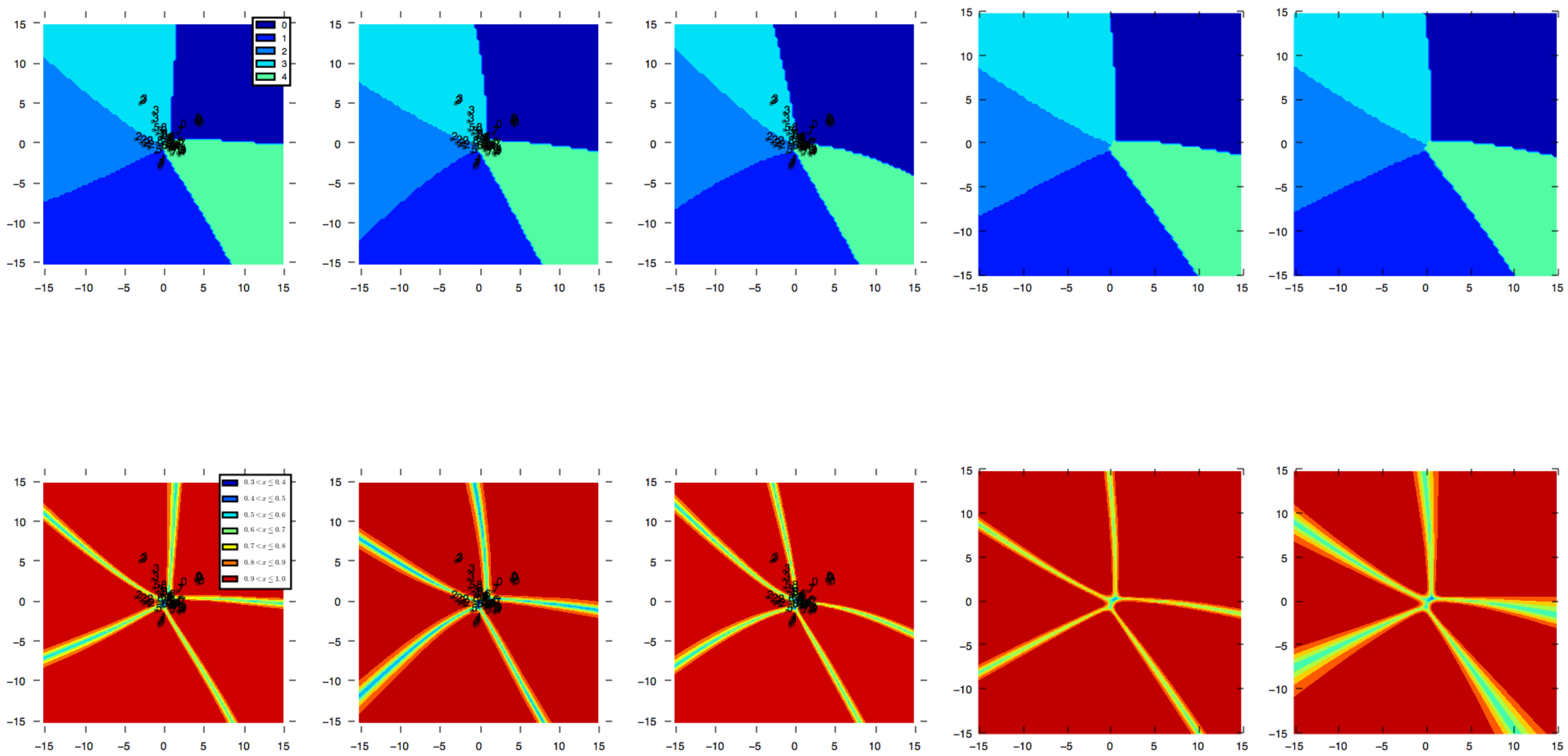
# Tanh

Sample 1 | Sample 2 | Sample 3 | Mean of q(W) | $E_{q(W)}[p(y|x,w)]$

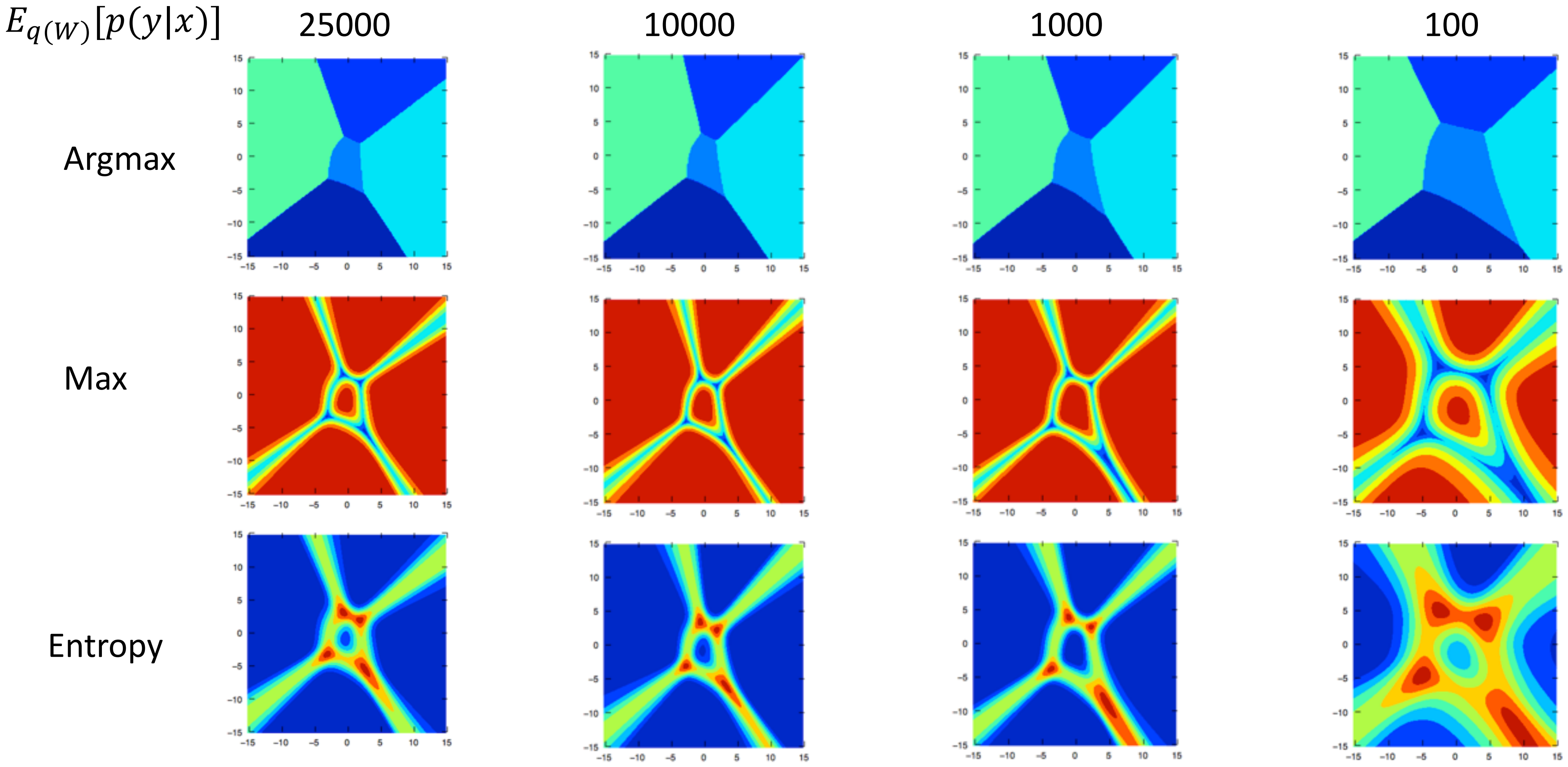


# Mix (Softplus, ReLu, Tanh)

Sample 1 | Sample 2 | Sample 3 | Mean of q(W) | $E_{q(W)}[p(y|x)]$



# Number of Datapoints



# Model vs Output Uncertainty

- Predictive Uncertainty = $H[p(y|x)]$

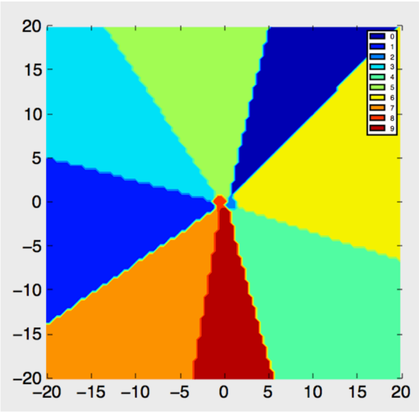
Output Uncertainty

Model Uncertainty

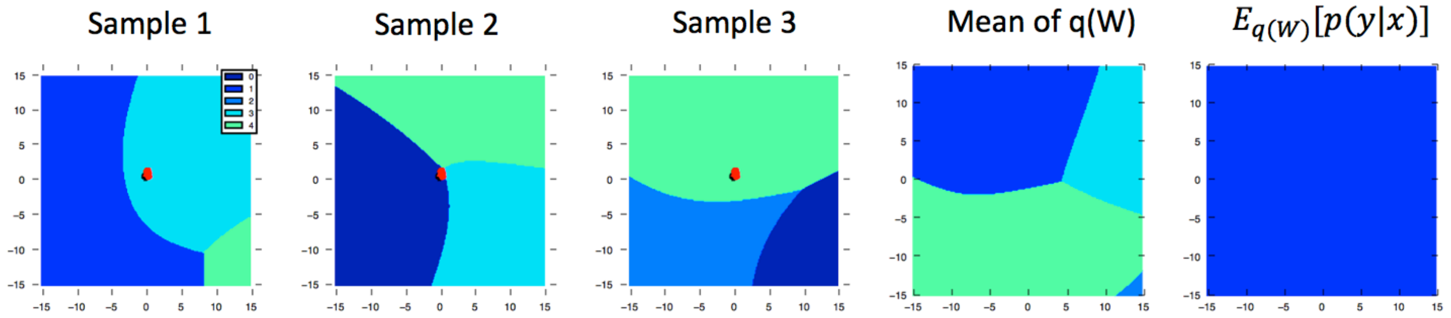
$H[p(y|x, w_m)]$
where $w_m$ = mean of q(w)

$H[E_{q(W)}[p(y|x, w)]]$

Output high entropy (on decision boundary)



High variance predictions



# Model vs Output Uncertainty

25000 training datapoints

| | Train | Test | Held Out |
|---|---|---|---|
| Model Uncertainty | .06 | .06 | .43 |
| Output Uncertainty | .05 | .05 | .36 |

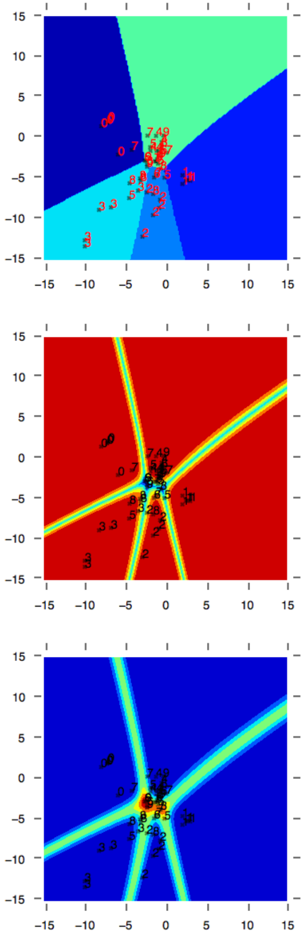
Large data: output uncertainty

100 training datapoints

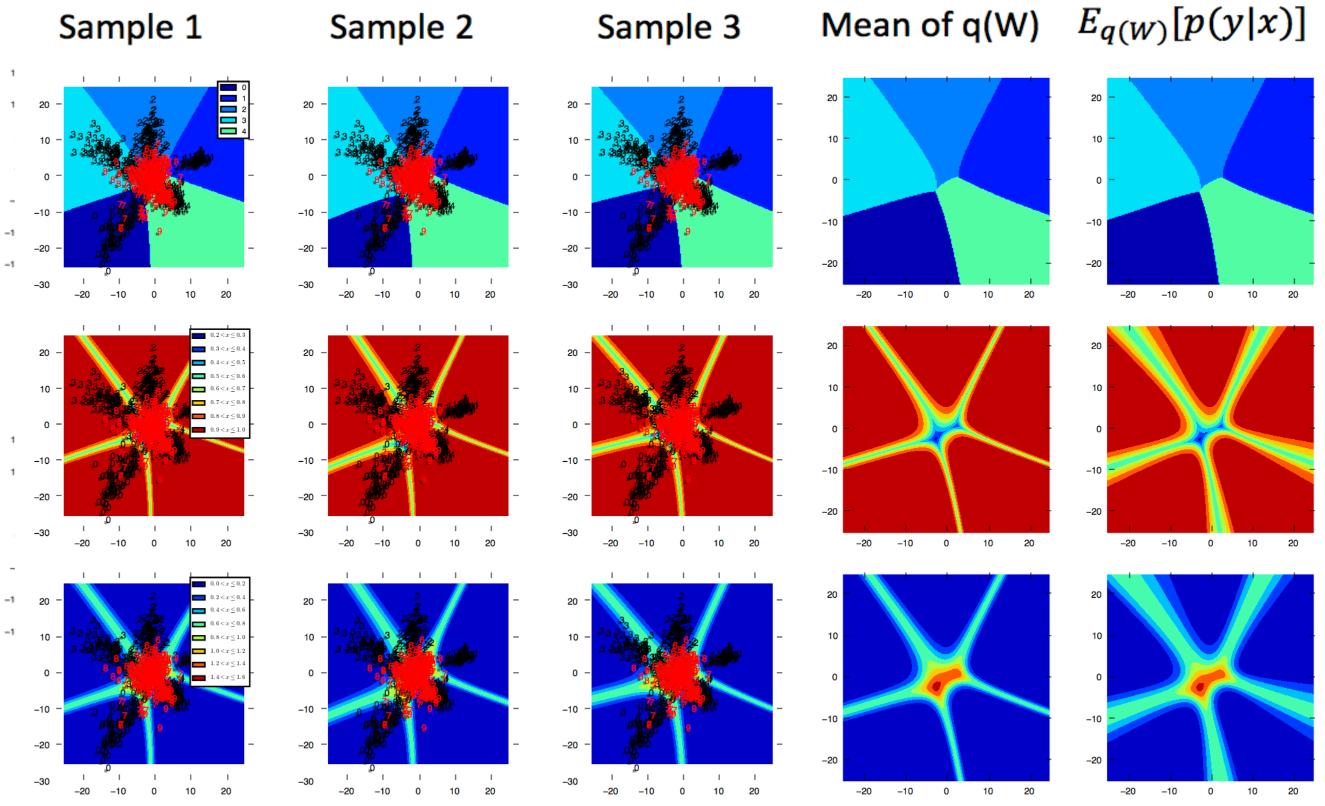
| | Train | Test | Held Out |
|---|---|---|---|
| Model Uncertainty | .07 | .26 | .43 |
| Output Uncertainty | .03 | .15 | .25 |

Small data: model uncertainty

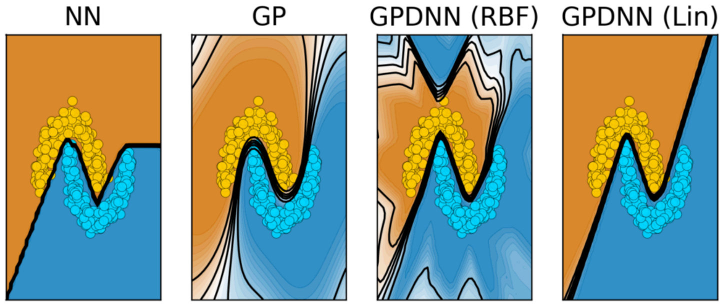
# NN



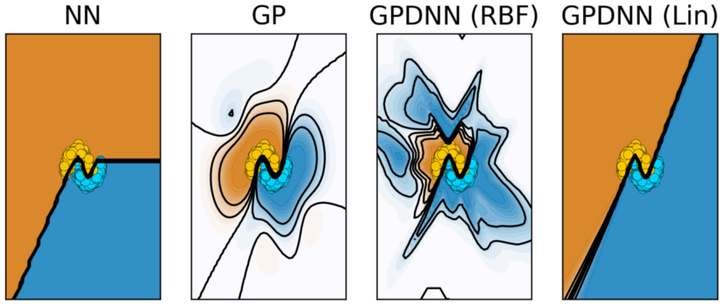
# BNN




# GP+NN



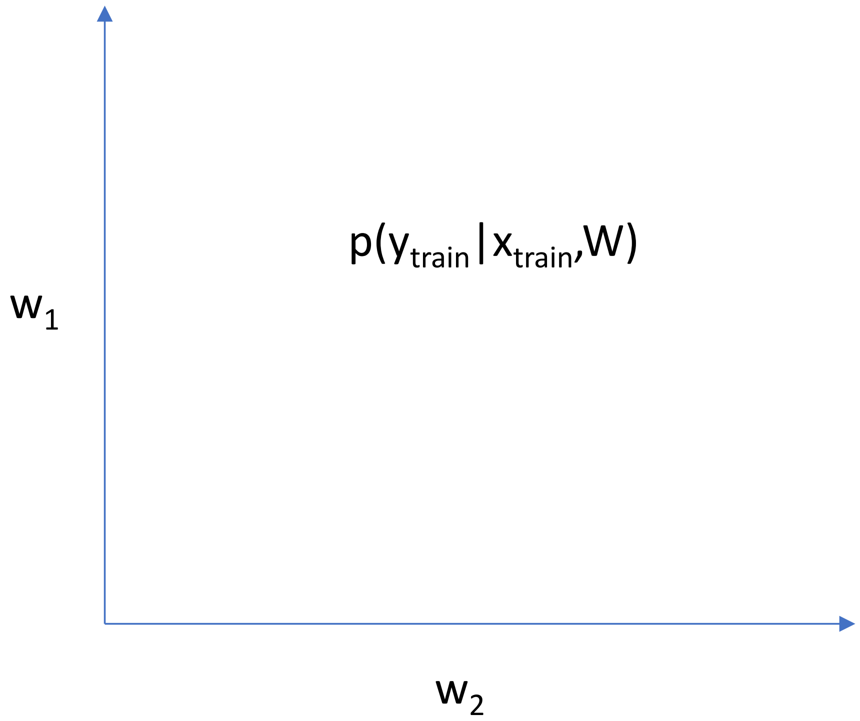

(a) zoomed in




(b) zoomed out

Adversarial Examples, Uncertainty, and Transfer Testing Robustness in Gaussian Process Hybrid Deep Networks (July 2017)

# Visualize landscape of likelihood




Dimension of W is large, so use an 2D auxiliary variable

# Visualize landscape of likelihood

- Auxiliary Variable Model

(2D)

$z \sim q(z)$

$W \sim q(W|z)$

hyper-network

r$(z|W)$

hypo-network

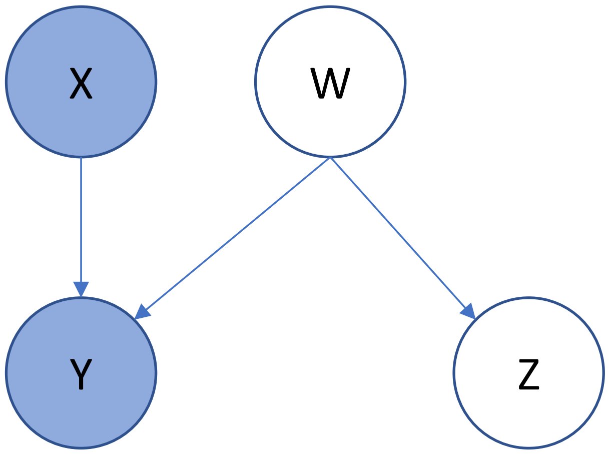
$\underbrace{\text{784-100-100-2}}_{\text{NN}}\underbrace{\text{-10-10-10}}_{\text{BNN}}$

$$q(W|z) = \delta(W|z)$$

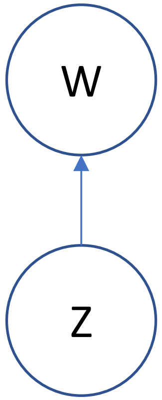
$$q(W) = \int \delta(W|z)q(z)dz$$

$$\log p(Y|X) \geq E_{q(W)}[\log p(Y|X,W) + \log p(W) - \log q(W|z) + \log r(z|w) - \log q(z)]$$

Generative Model

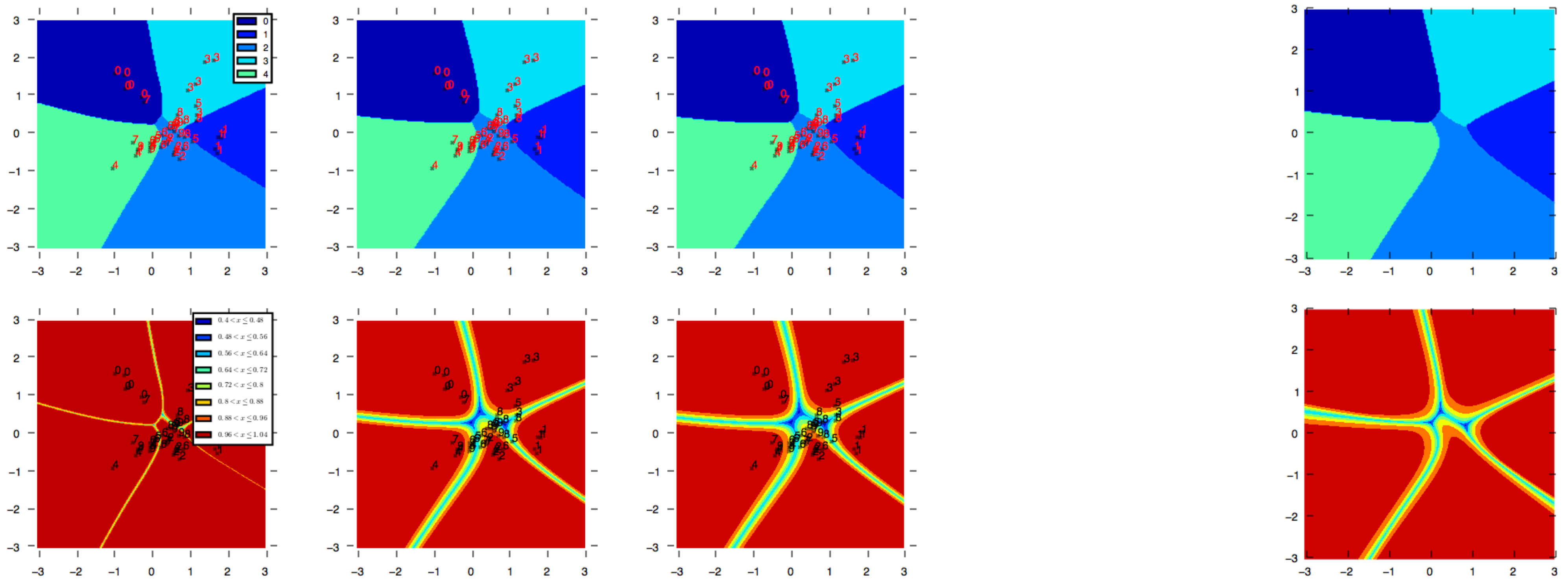



Inference Model

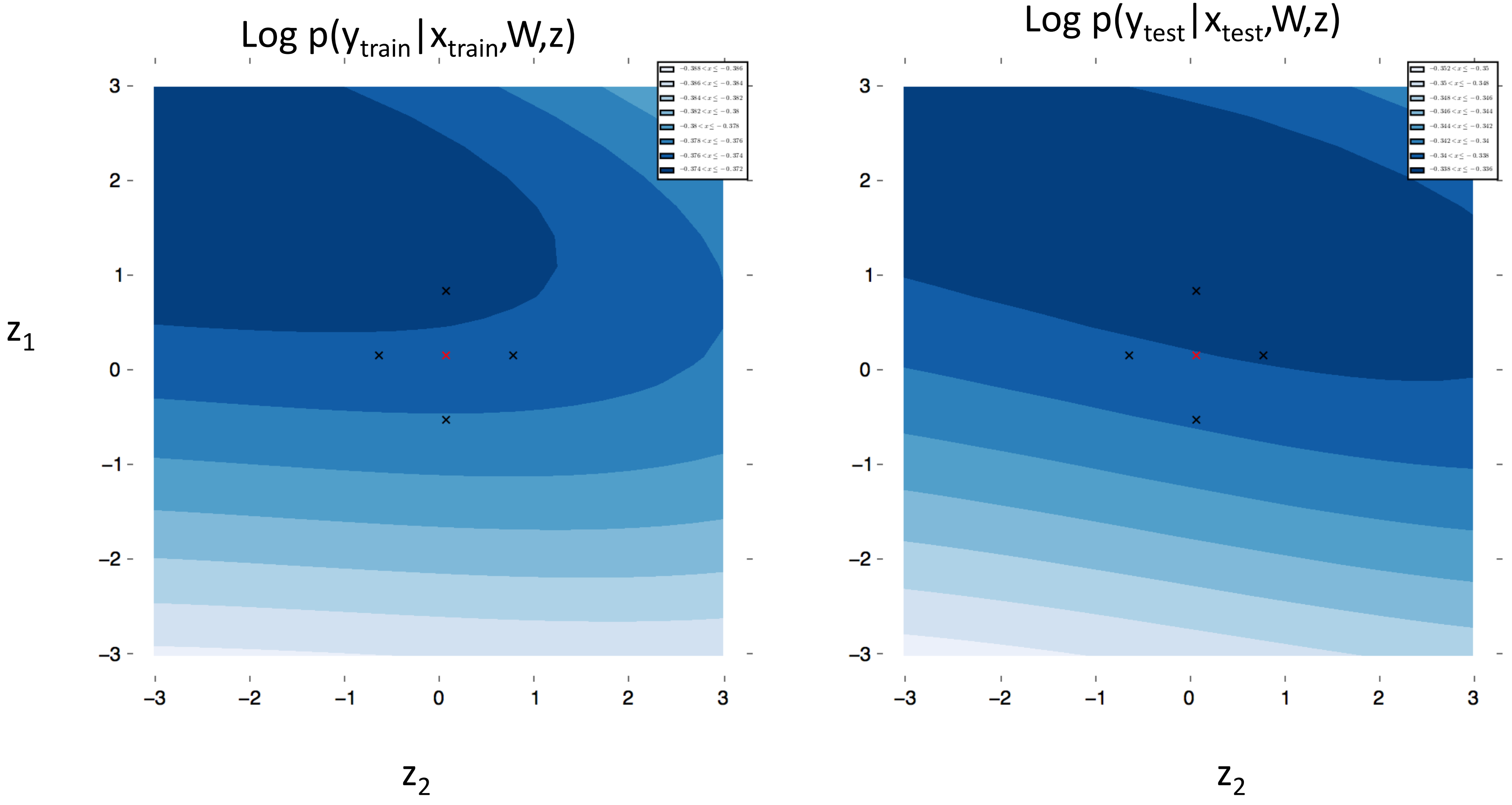


# Decision Boundaries

$z_1$ $z_2$ $z_3$ $E_{q(z)}[p(y|x,z)]$



# Likelihood Landscape

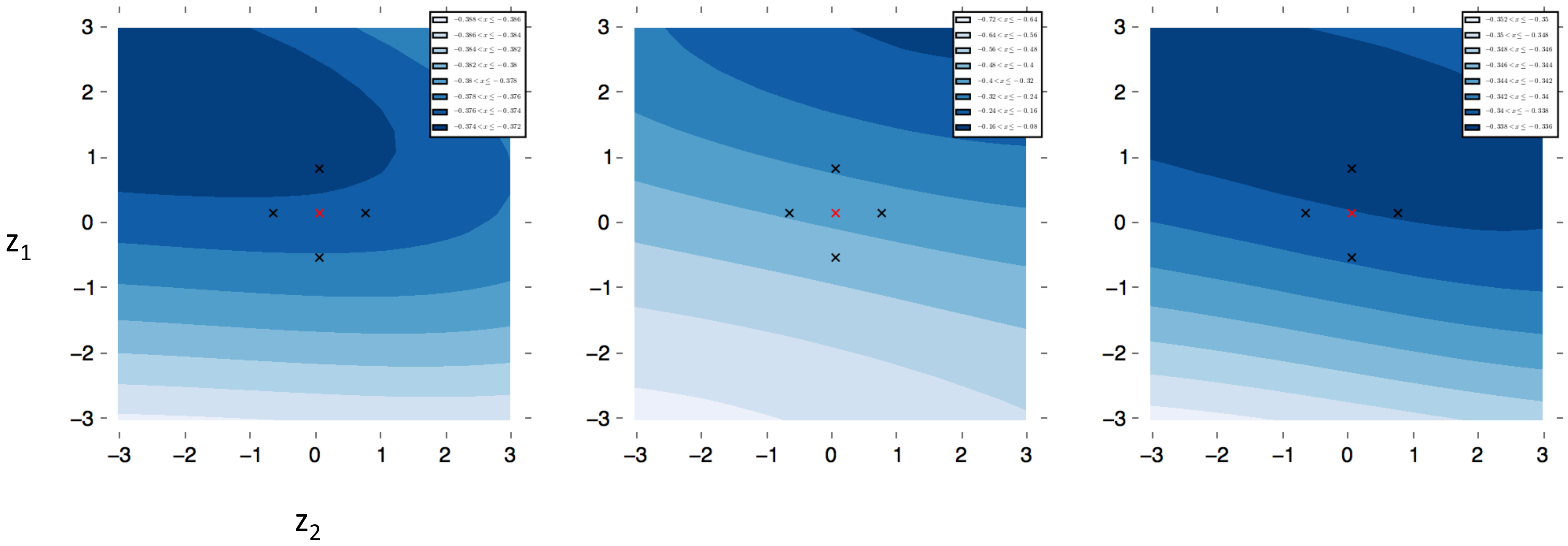


# Likelihood Landscape

$\log p(y_{train} | x_{train}, W, z)$

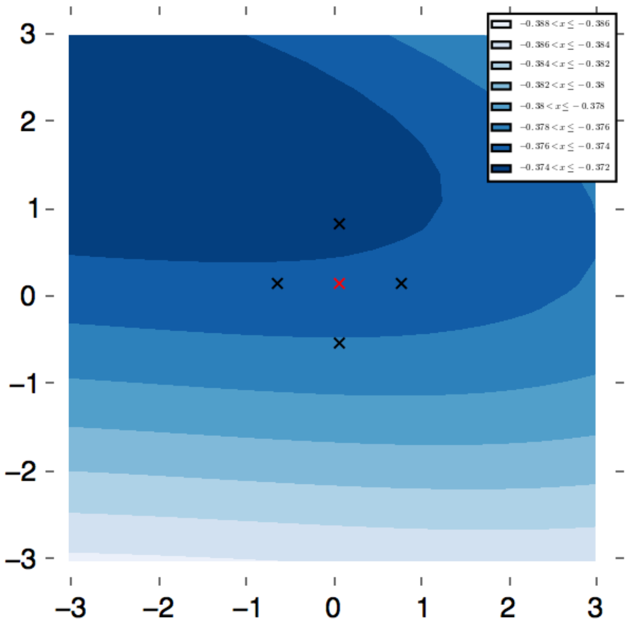
$\log p(y_{train} | x_{train}, W, z)$
$+ \log r(z | W)$
$- \log q(z)$

$\log p(y_{test} | x_{test}, W, z)$

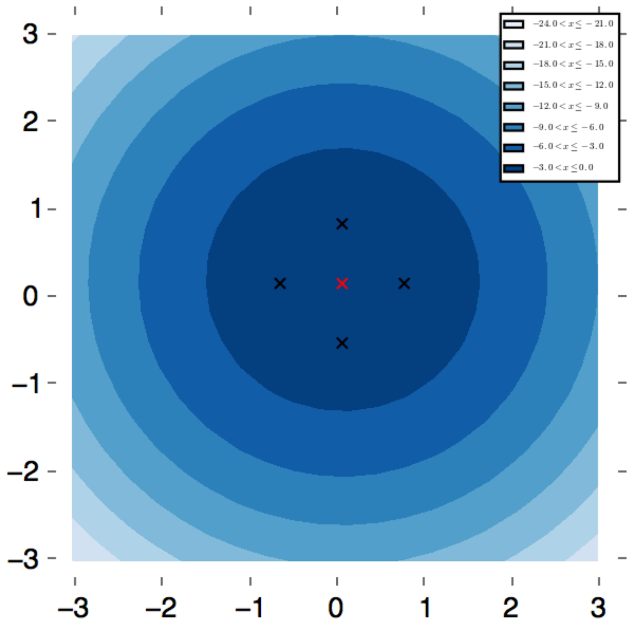


# Likelihood Landscape

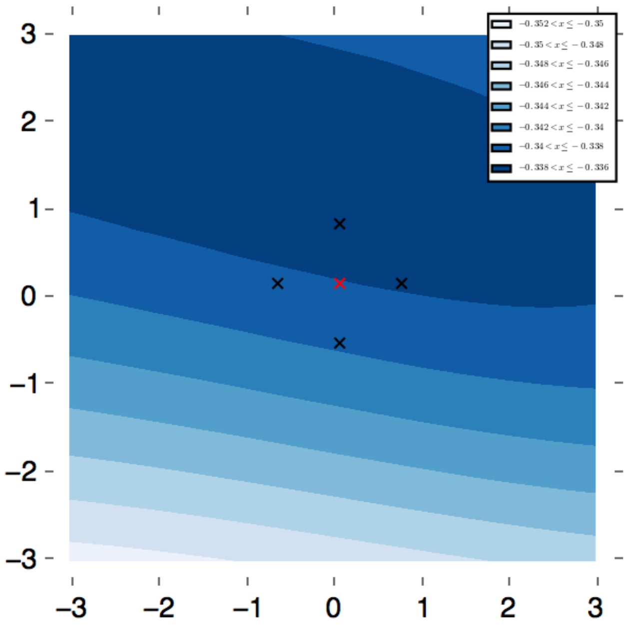
$\text{Log } p(y_{train} | x_{train}, W, z)$



$\log p(y_{train} | x_{train}, W, z)$
$+ \log r(z | W)$
$\cancel{- \log q(z)}$



$\text{Log } p(y_{test} | x_{test}, W, z)$



$z_1$

$z_2$

# Likelihood Landscape

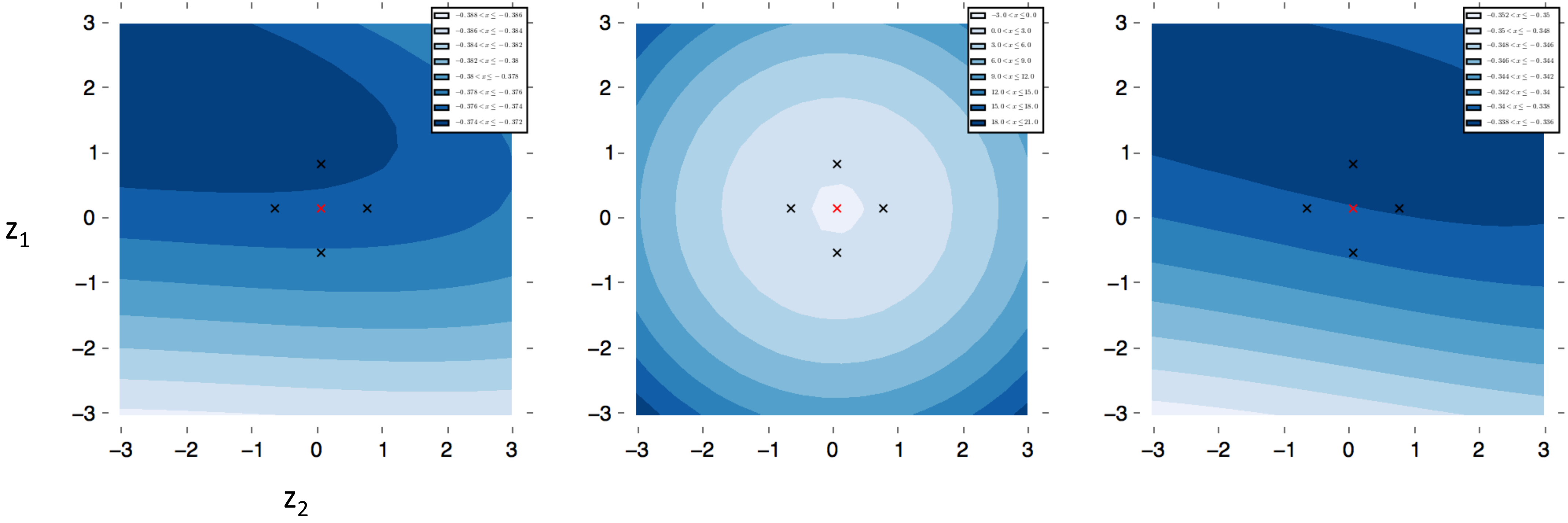
$\log p(y_{train}|x_{train},W,z)$

$\cancel{+ \log r(z|W)}$

$- \log q(z)$

Log $p(y_{train}|x_{train},W,z)$

Log $p(y_{test}|x_{test},W,z)$

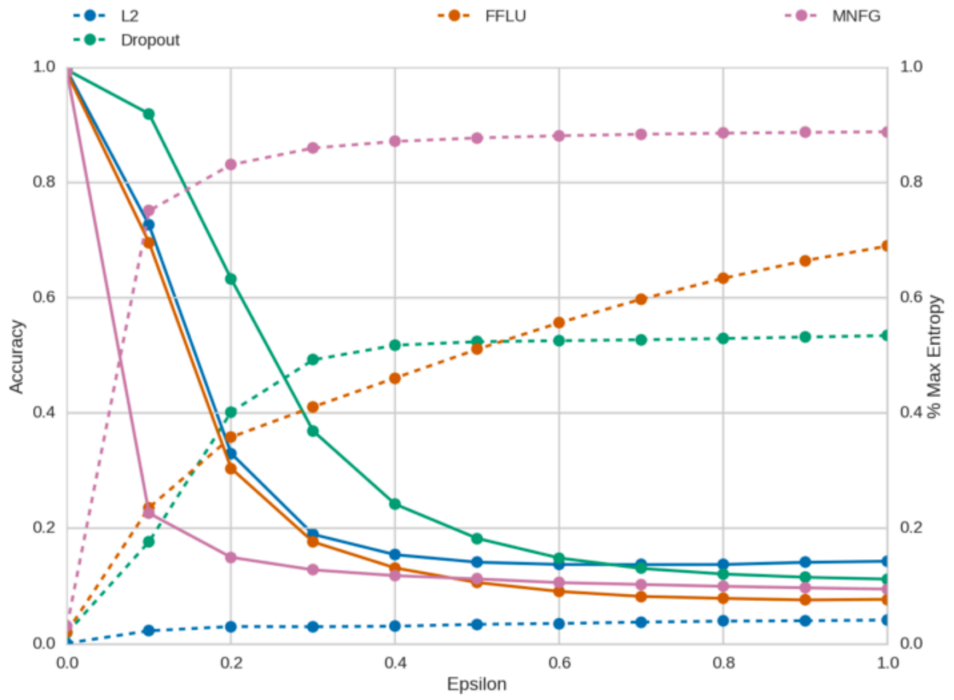


# Recent BNN Papers

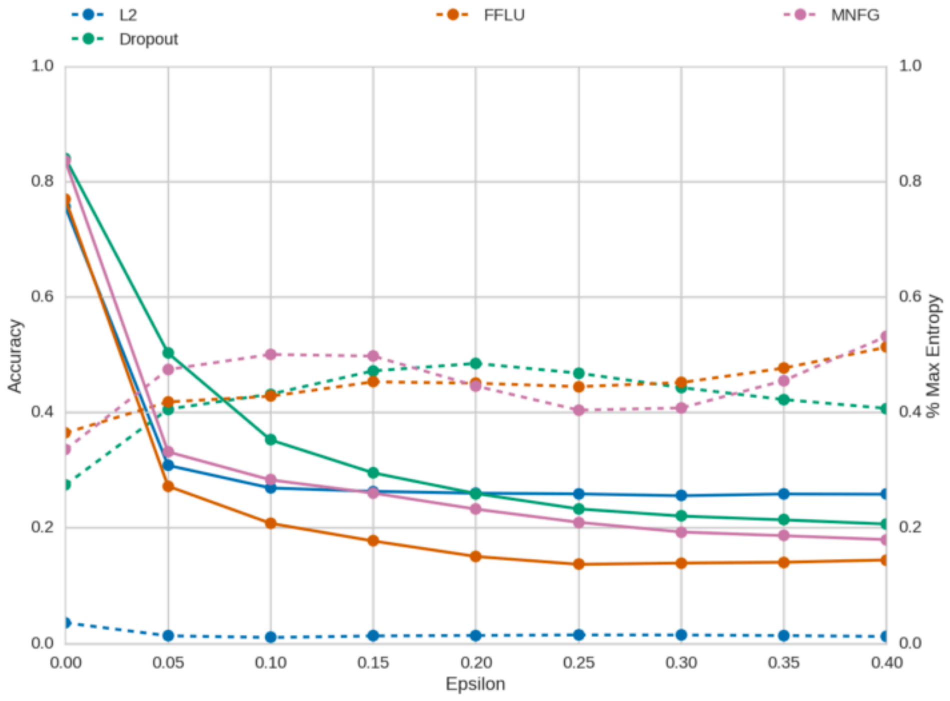
- Multiplicative Normalizing Flows for Variational Bayesian Neural Networks (2017)
- Variational Dropout Sparsifies Deep Neural Networks (2017)
- Bayesian Compression for Deep Learning (2017)

- Adversarial Perturbations
- Compression

# Adversarial perturbations

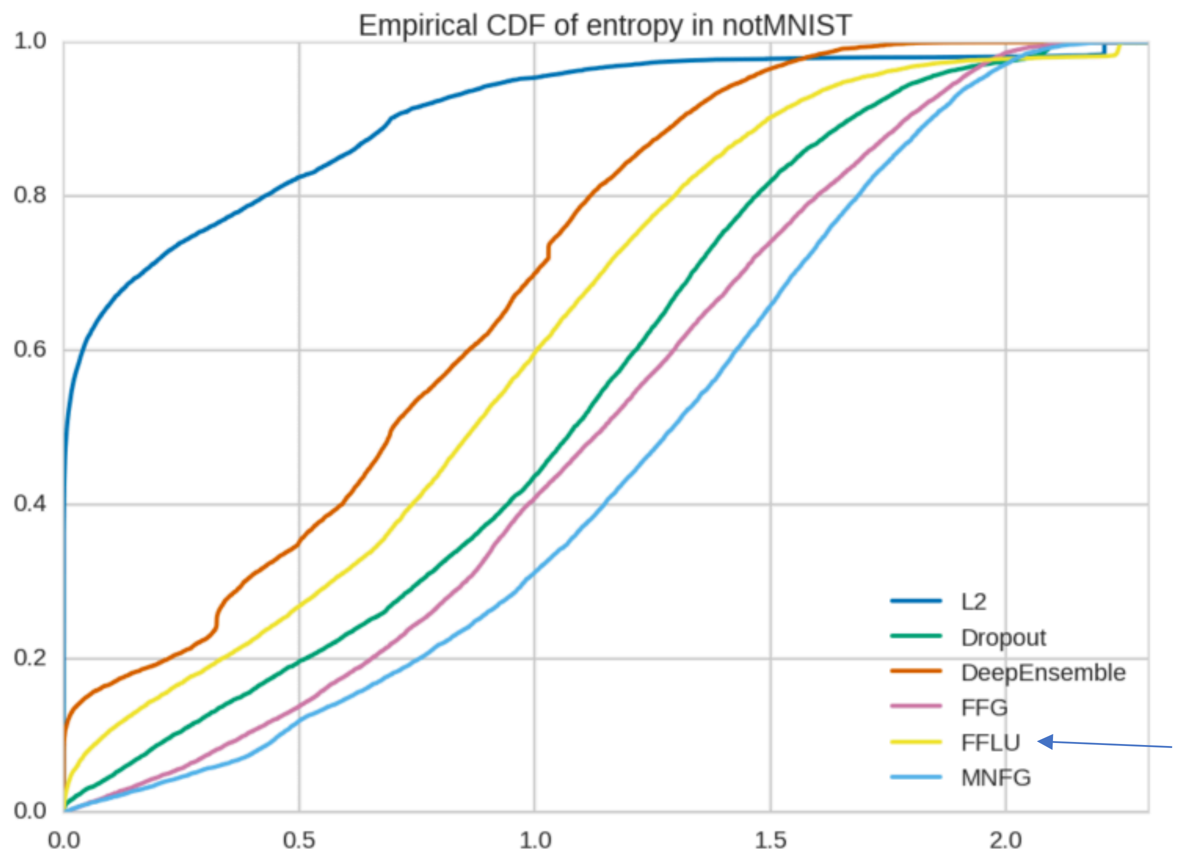



MNIST




CIFAR 10

# Compression vs Uncertainty




H[P]

# Conclusion

- Used visualizations to help understand uncertainty in BNNs
- Goal: improve uncertainty estimates and generalization

Applications

- Active learning
- Bayes Opt
- RL
  - Safety
  - Efficiency

# References


- Weight Uncertainty in Neural Networks (2015)
- Variational Dropout and the Local Reparameterization Trick (2015)
- Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning (2016)
- Variational Dropout Sparsifies Deep Neural Networks (2017)
- On Calibration of Modern Neural Networks (2017)
- Multiplicative Normalizing Flows for Variational Bayesian Neural Networks (2017)

# Thank You