What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision

Patryk Chrabąszcz

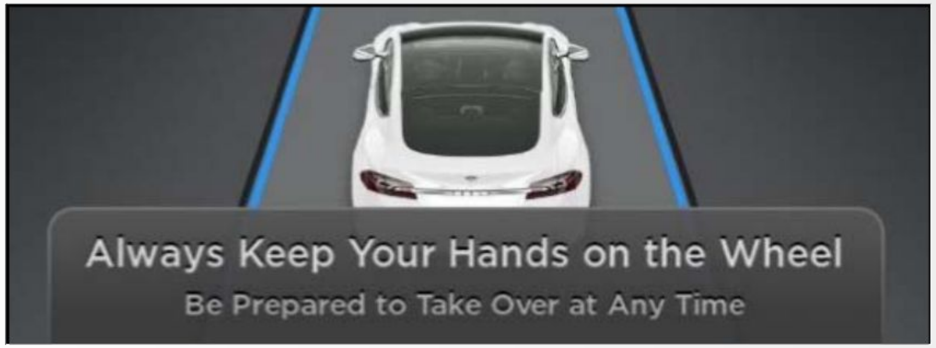
Outline:

- **Motivation**
- Types of Uncertainty
- Bayesian Neural Networks
- Dropout Variational Inference
- Modeling uncertainties
- Experiments
- Results Analysis
- Summary

Importance of modeling uncertainty

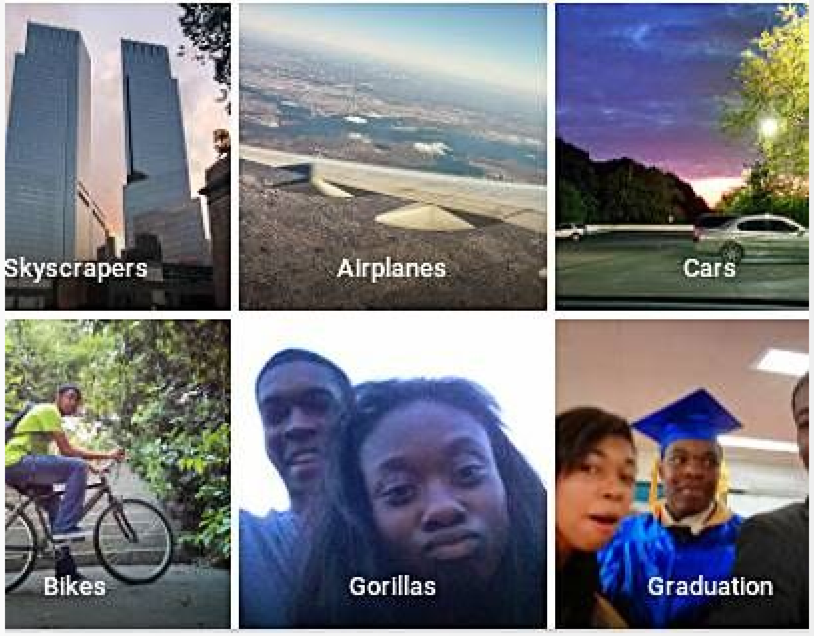
Importance of modeling uncertainty

- **Autonomous Car Accident**



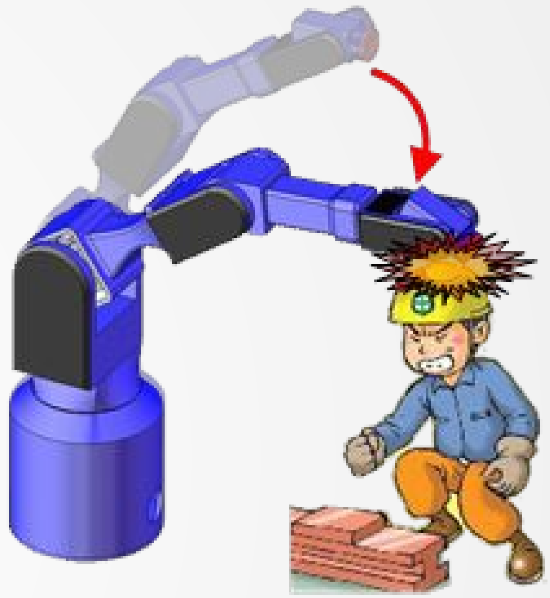
Importance of modeling uncertainty

- Autonomous Car Accident
- **Google app racial discrimination**



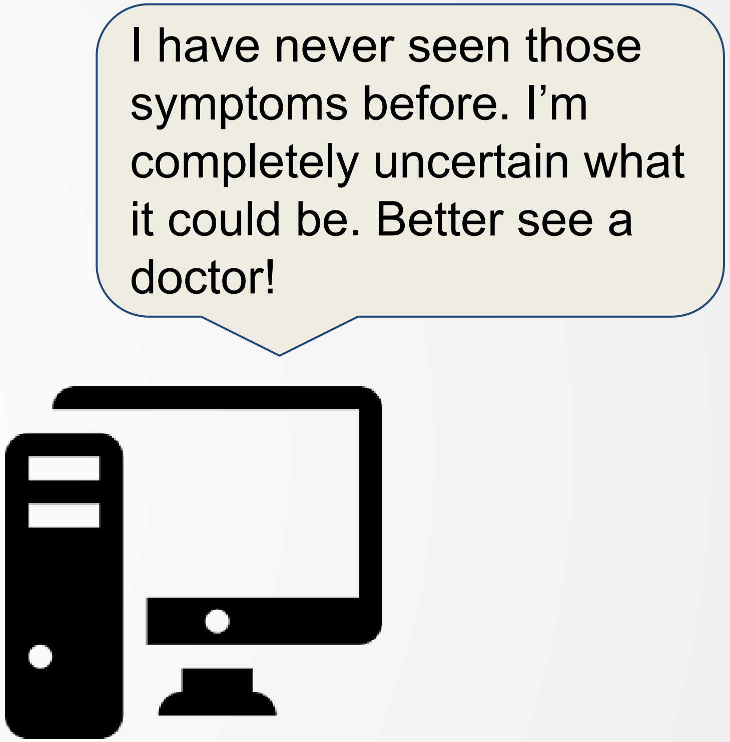
Importance of modeling uncertainty

- Autonomous Car Accident
- Google app racial discrimination
- **Safety Critical Systems**



Importance of modeling uncertainty

- Autonomous Car Accident
- Google app racial discrimination
- Safety Critical Systems
- **Medical Applications**



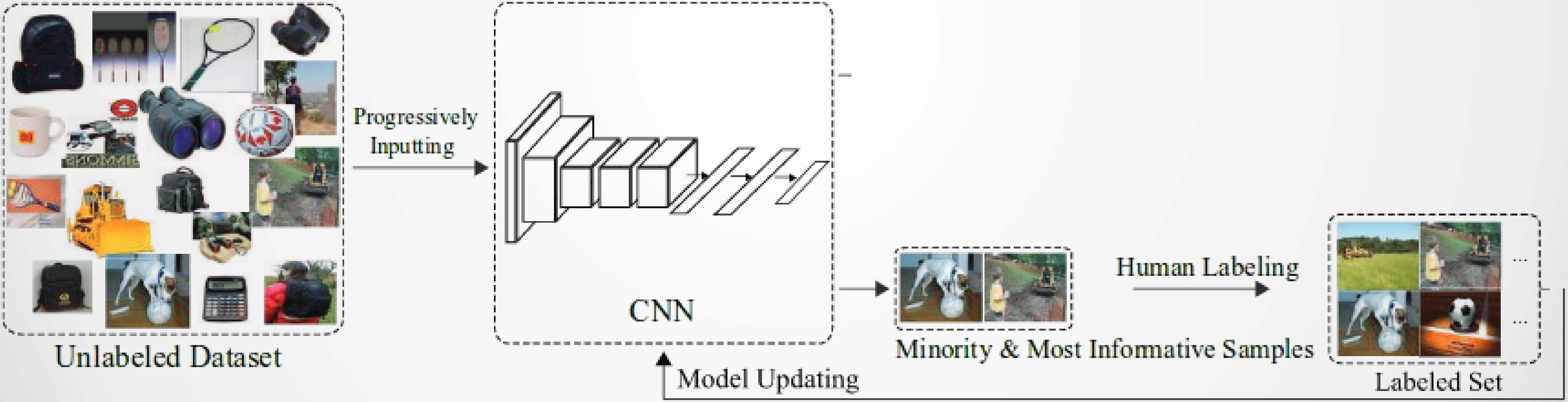
Example: Active Learning

Example: Active Learning

- **Model decides which data should be labeled**

Example: Active Learning

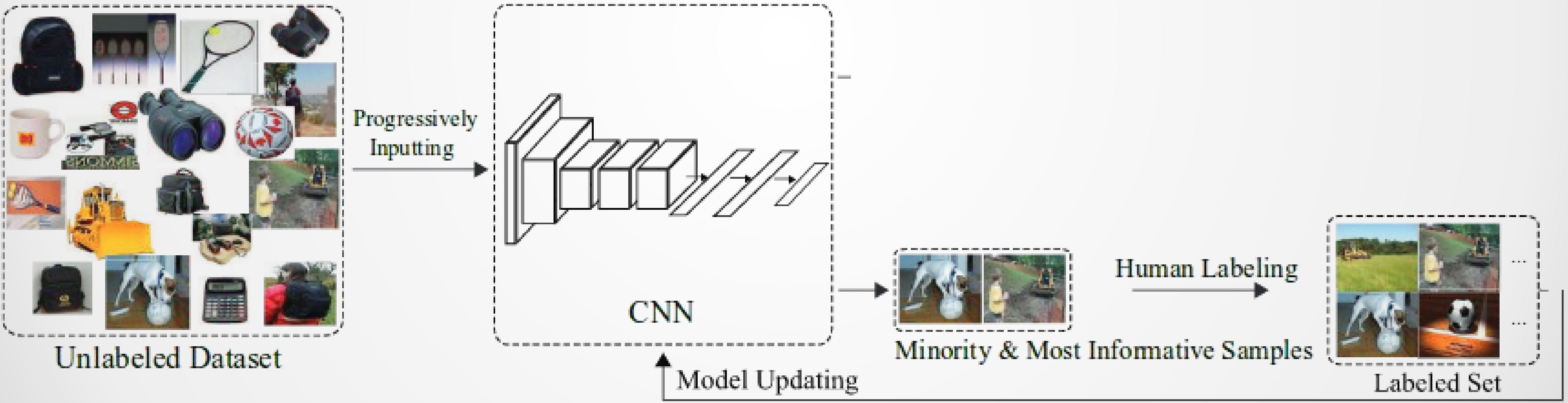
- **Model decides which data should be labeled**



Source: *“Cost-Effective Active Learning for Deep Image Classification”*

Example: Active Learning

- Model decides which data should be labeled
- **Collect the best data at low cost**

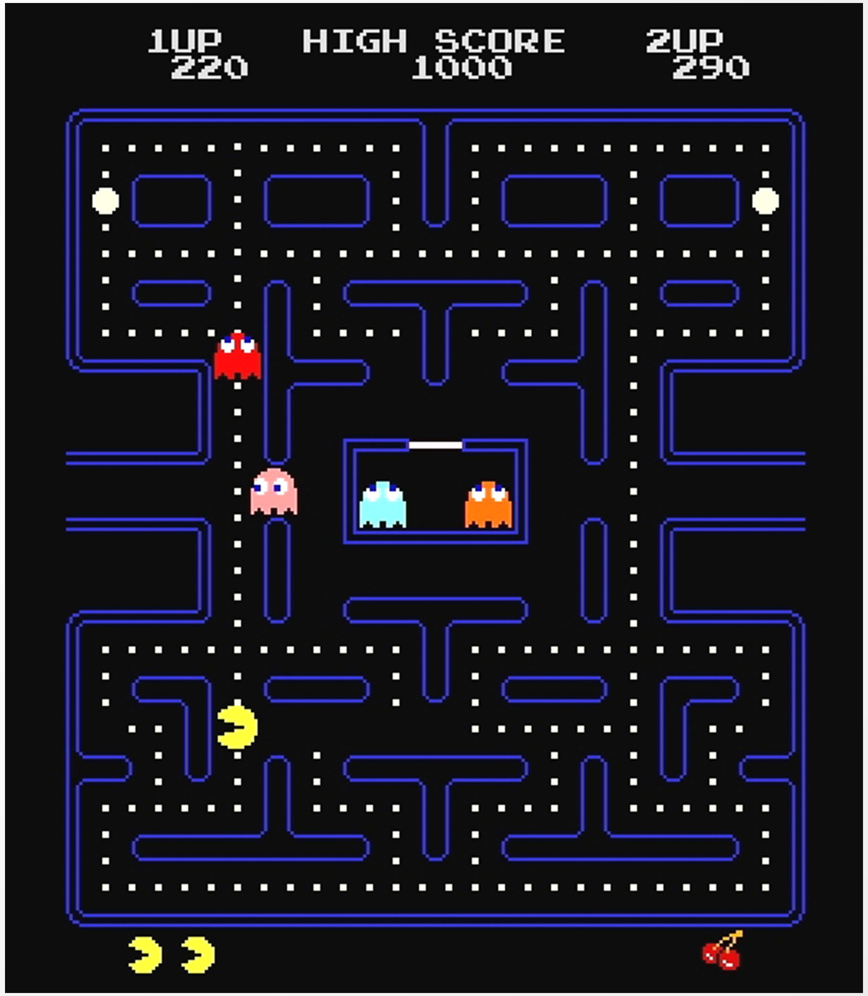


Source: *"Cost-Effective Active Learning for Deep Image Classification"*

Example: Reinforcement Learning

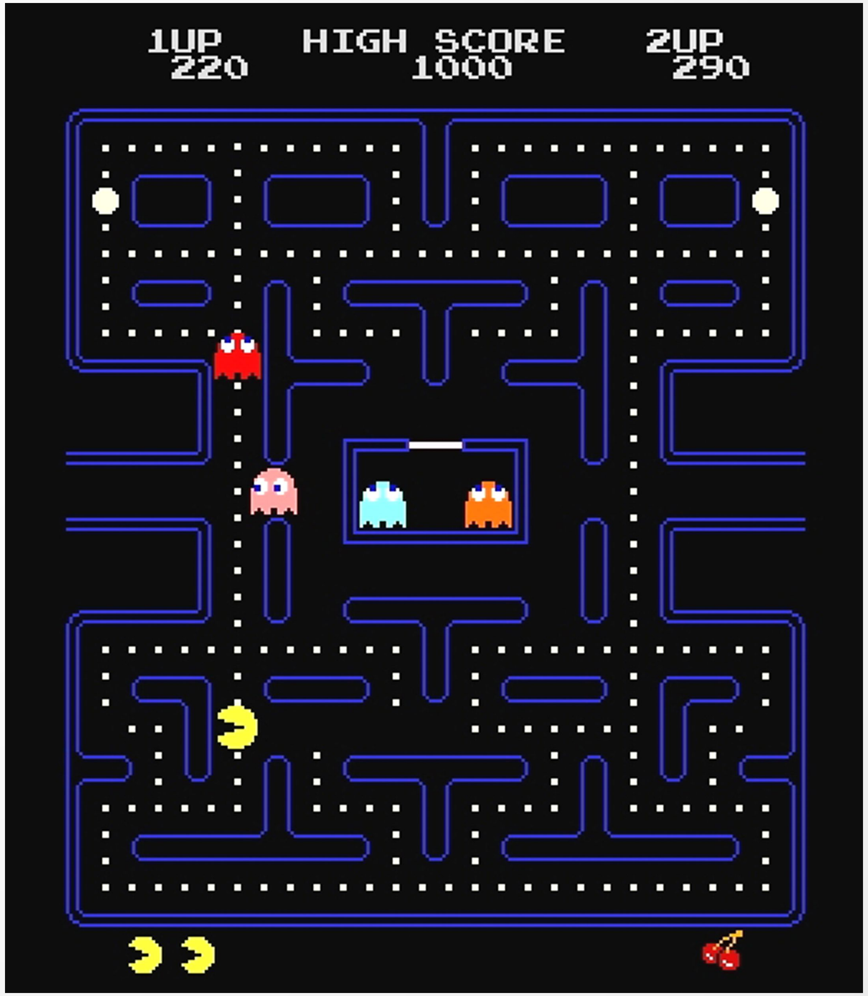
Example: Reinforcement Learning

- **No knowledge at the start**



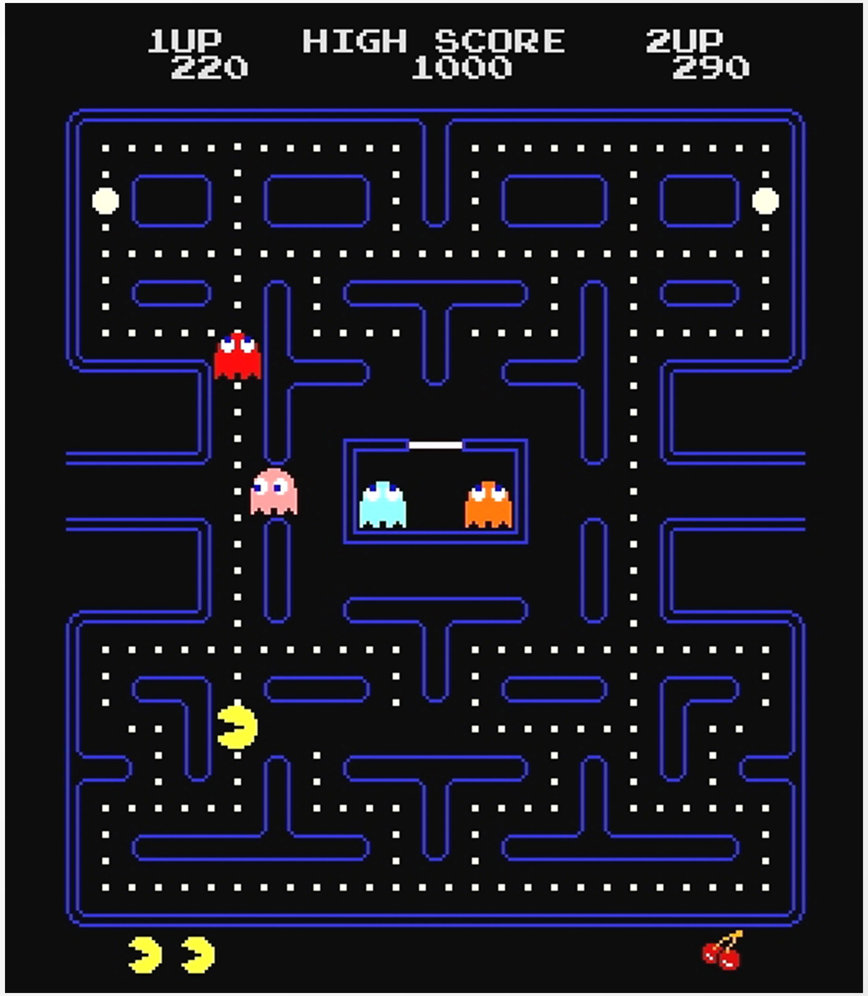
Example: Reinforcement Learning

- No knowledge at the start
- **Make decision at each step**
  - Explore ?
  - Exploit ?



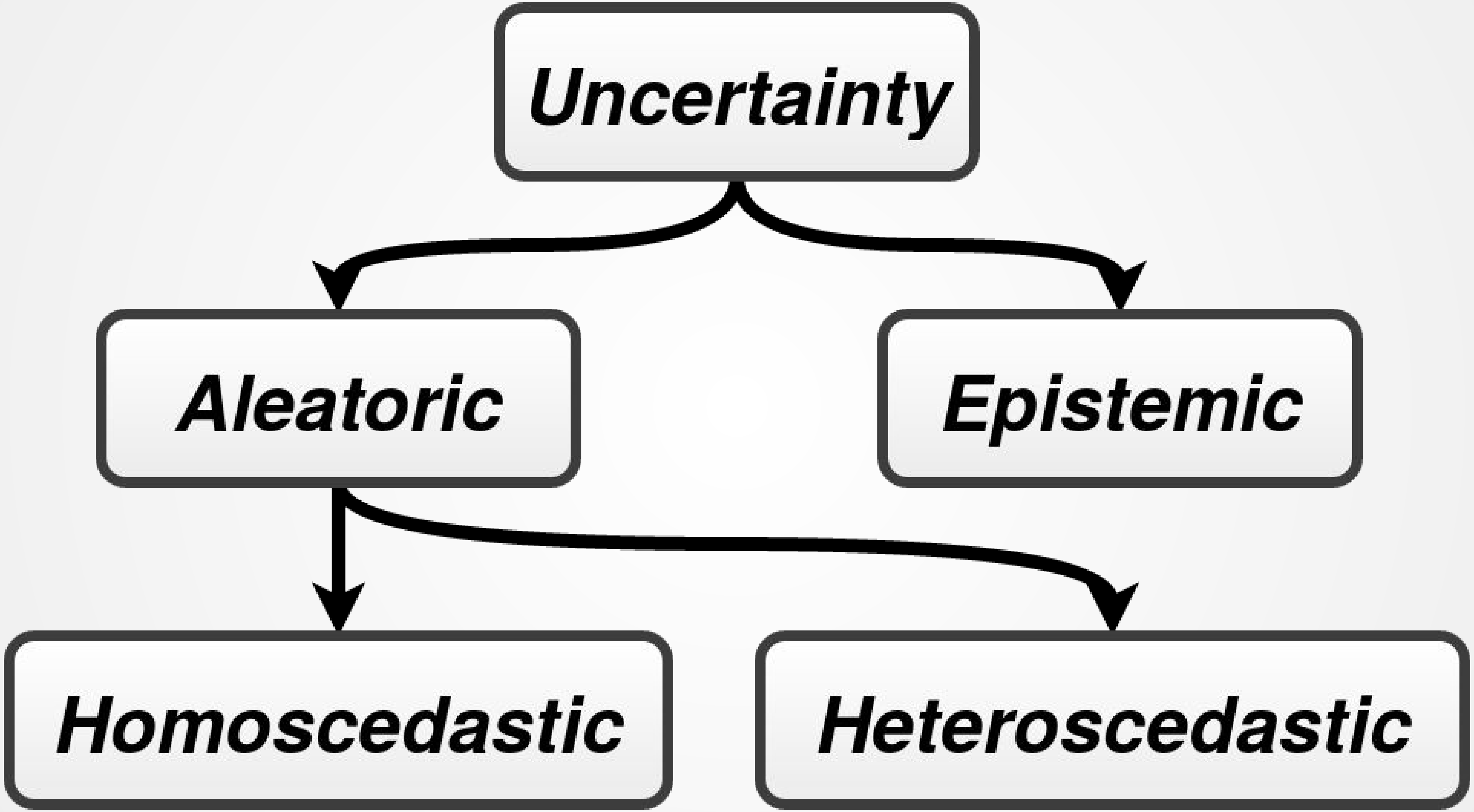
Example: Reinforcement Learning

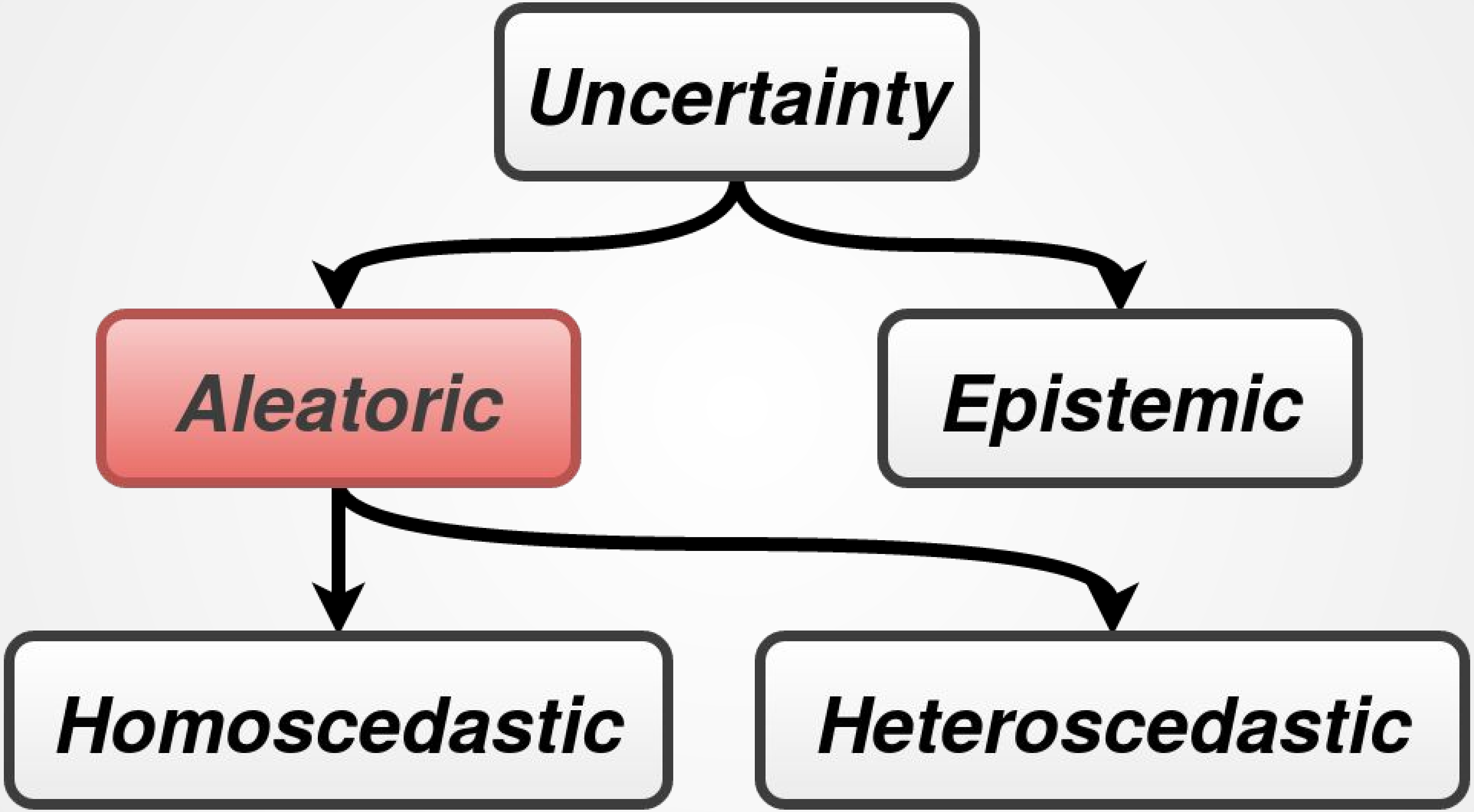
- No knowledge at the start
- Make decision each step
  - Explore ?
  - Exploit ?
- **Intelligent exploration**



Outline:

- Motivation
- **Types of Uncertainty**
- Bayesian Neural Networks
- Dropout Variational Inference
- Modeling uncertainties
- Experiments
- Results Analysis
- Summary


Uncertainty
Aleatoric
Epistemic
Homoscedastic
Heteroscedastic


Uncertainty
Aleatoric
Epistemic
Homoscedastic
Heteroscedastic

Aleatoric uncertainty

- Natural randomness

Modeling the result of dice throw.

In latin āleae: a die

Aleatoric uncertainty

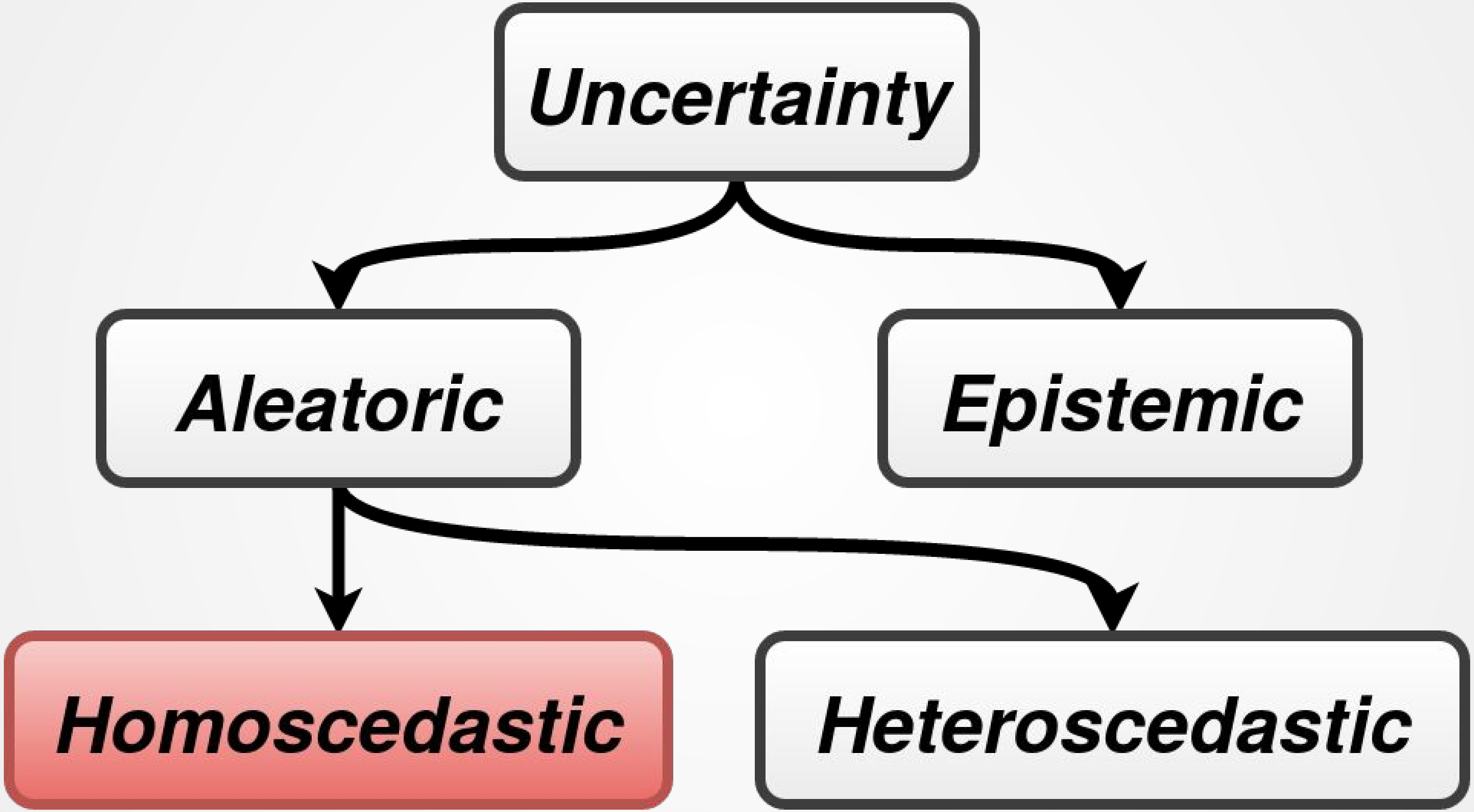
- Natural randomness
- Sensor quality

Aleatoric uncertainty

- Natural randomness
- Sensor quality
- Can’t be reduced

Aleatoric uncertainty

- Natural randomness
- Sensor quality
- Can’t be reduced
- But can be learned

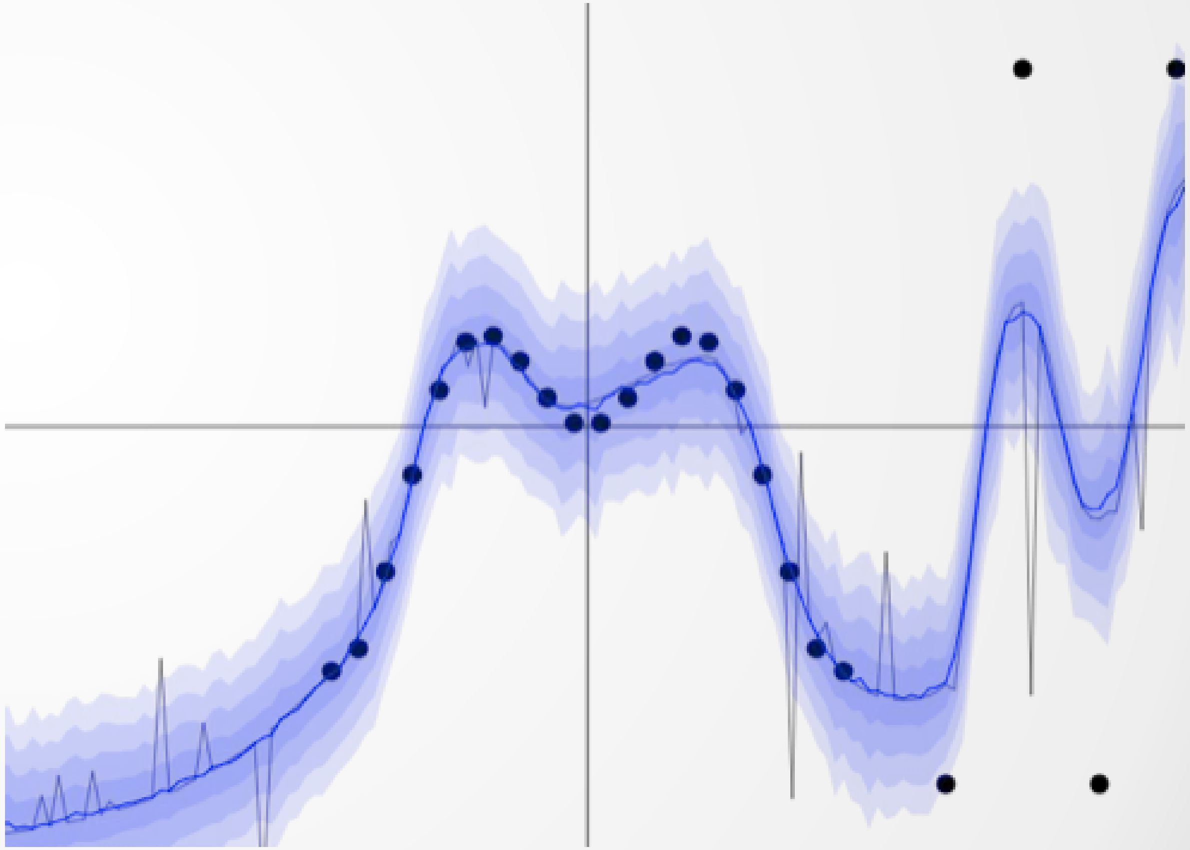


Uncertainty
Aleatoric
Epistemic
Homoscedastic
Heteroscedastic

Homoscedastic uncertainty

- **Stays constant for different**

input values



Source:Gal, Y. Uncertainty in Deep Learning. PhD thesis, University of Cambridge, 2016.

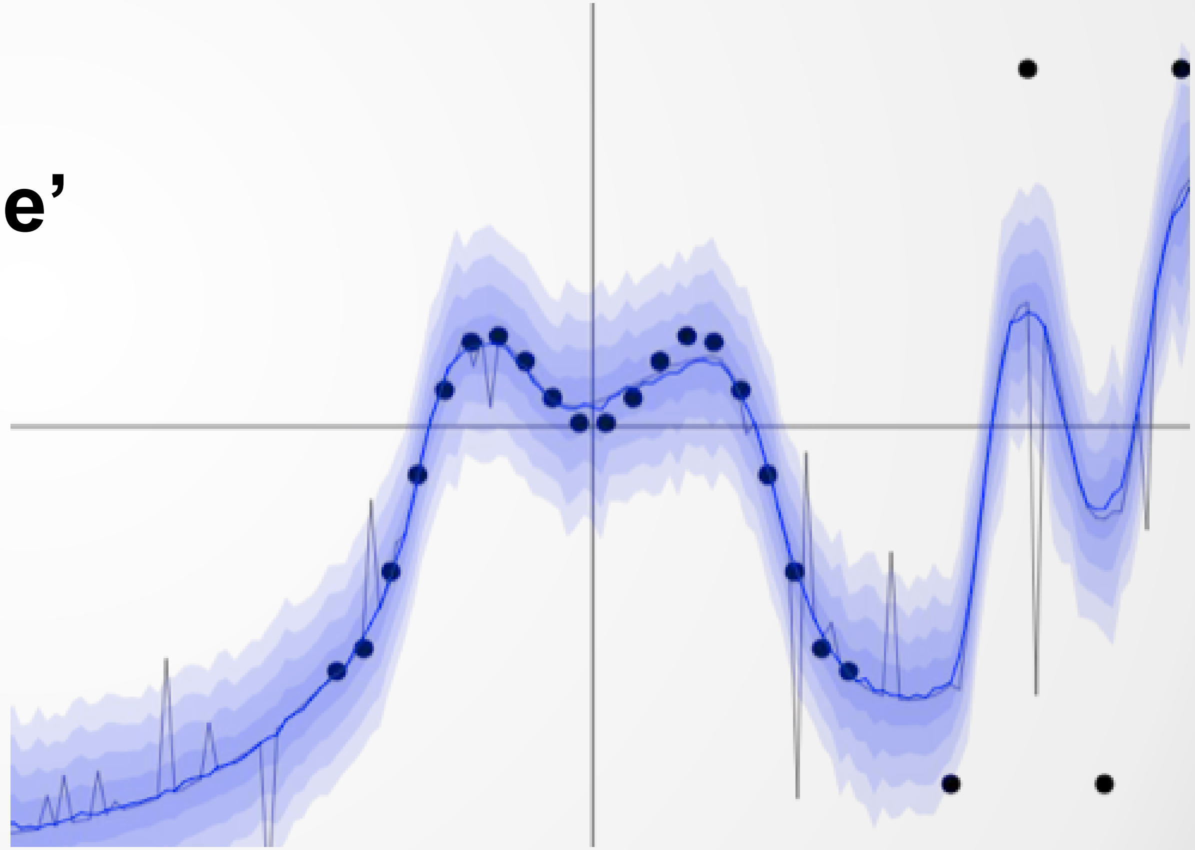
Homoscedastic uncertainty

- Stays constant for different

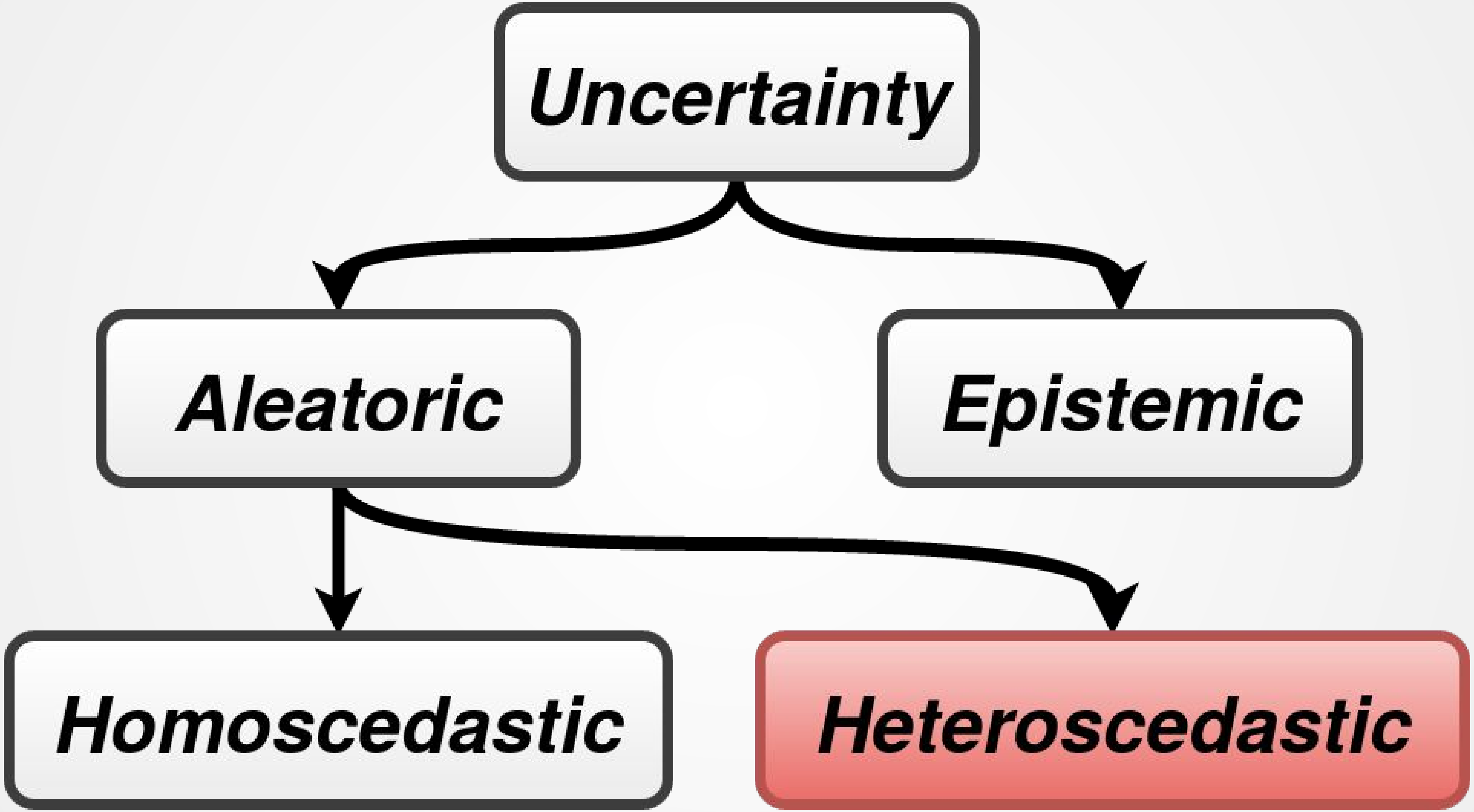
input values

- **Limited, captures ‘average’**

  uncertainty

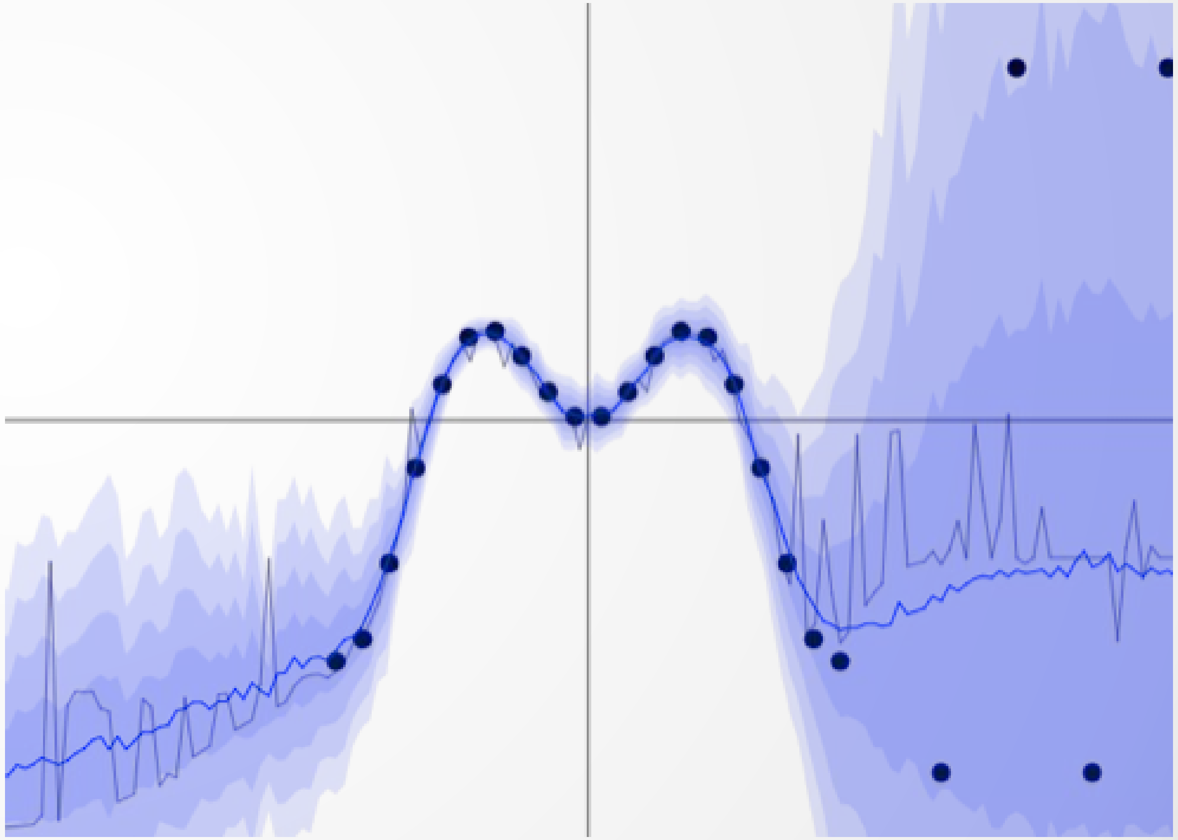


Source:Gal, Y. Uncertainty in Deep Learning. PhD thesis, University of Cambridge, 2016.


Uncertainty
Aleatoric
Epistemic
Homoscedastic
Heteroscedastic

Heteroscedastic uncertainty

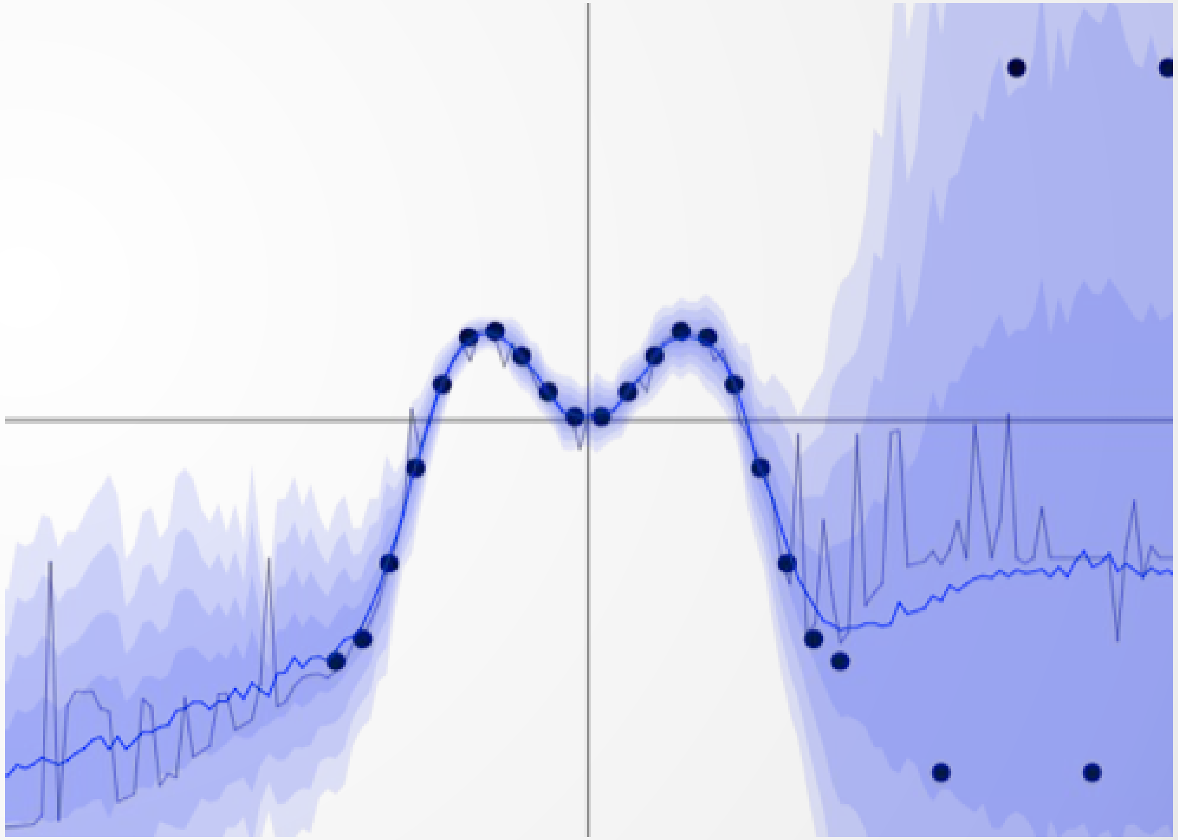
- **Depends on the input**



Source:Gal, Y. Uncertainty in Deep Learning. PhD thesis, University of Cambridge, 2016.

Heteroscedastic uncertainty

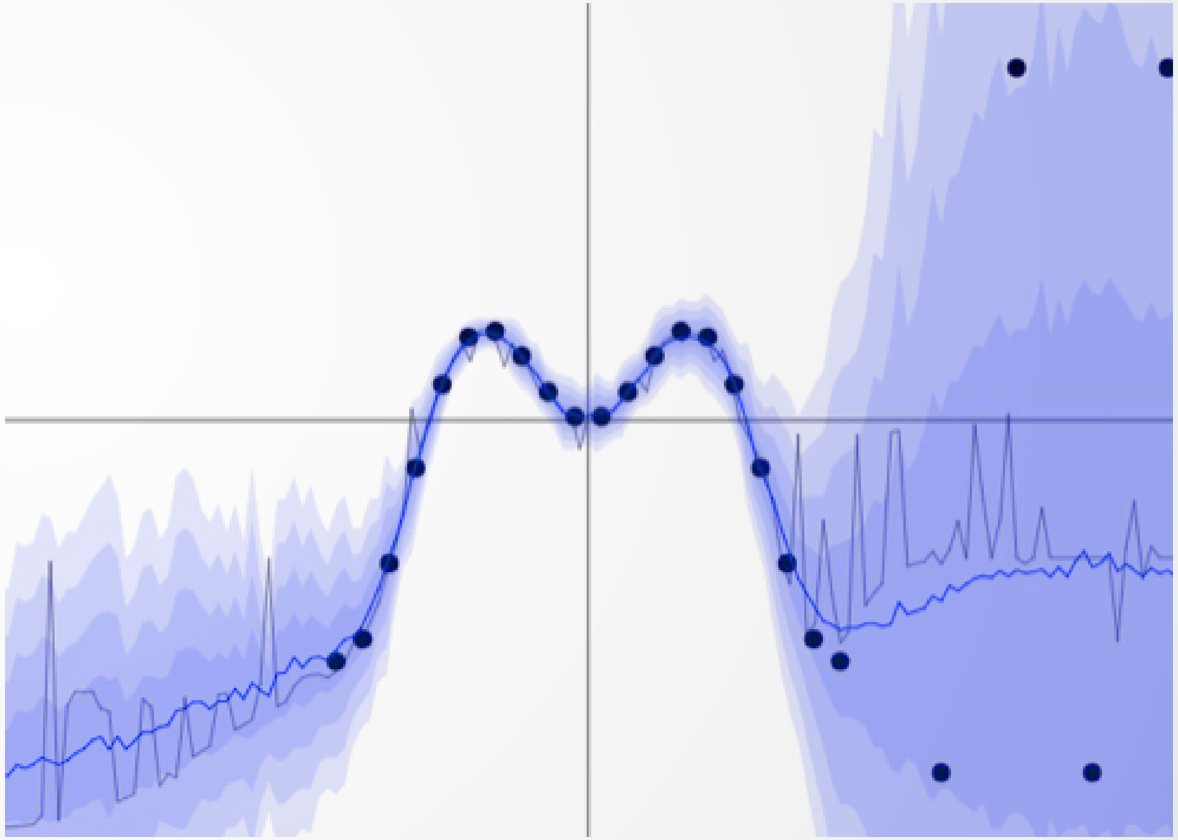
- Depends on the input
- **Important for CV tasks**



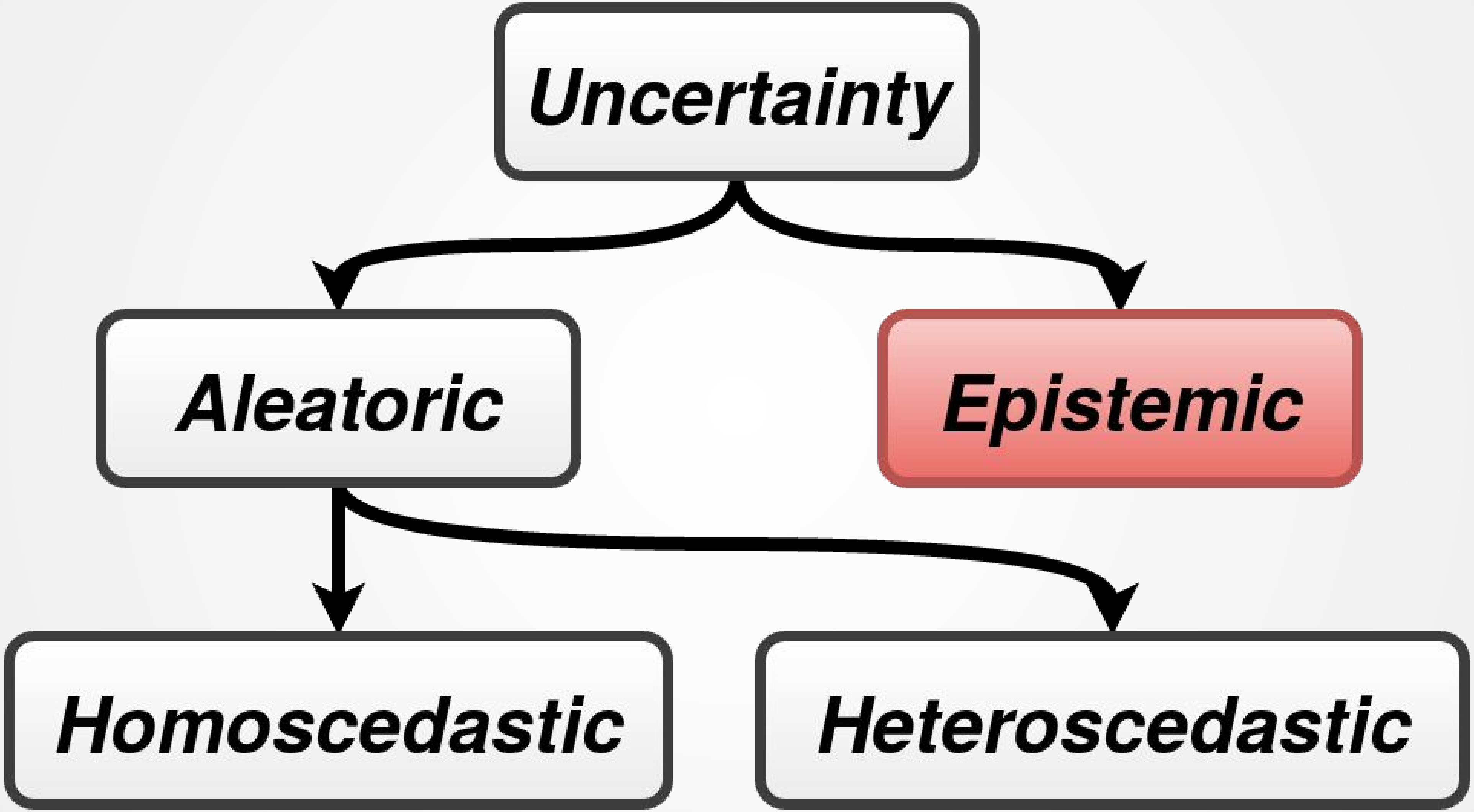
Source:Gal, Y. Uncertainty in Deep Learning. PhD thesis, University of Cambridge, 2016.

Heteroscedastic uncertainty

- Depends on the input
- Important for CV tasks
- **Learned from the data**



Source:Gal, Y. Uncertainty in Deep Learning. PhD thesis, University of Cambridge, 2016.


Uncertainty
Aleatoric
Epistemic
Homoscedastic
Heteroscedastic

Epistemic uncertainty

- **Lack of knowledge about the process**

Epistēmē Greek meaning: ***knowledge.***

Epistemic uncertainty

- Lack of knowledge about the process
- **Detects samples far from the training distribution**

Epistemic uncertainty

- Lack of knowledge about the process
- Detects samples far from training distribution
- **Disappears given enough data.**

Epistemic uncertainty

- Lack of knowledge about the process
- Detects samples far from training distribution
- Disappears given enough data.
- **Train many models, detect where models disagree**

Epistemic uncertainty

- Lack of knowledge about the process
- Detects samples far from training distribution
- Disappears given enough data.
- Train many models, detect where models disagree
- **Use distribution over model weights**

Outline:

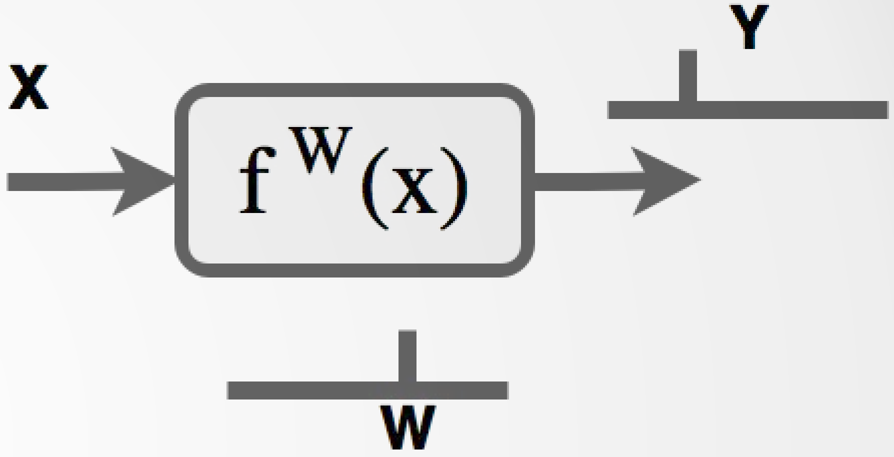
- Motivation
- Types of Uncertainty
- **Bayesian Neural Networks**
- Dropout Variational Inference
- Modeling uncertainties
- Experiments
- Results Analysis
- Summary

Bayesian Neural Networks

- **Neural Network**

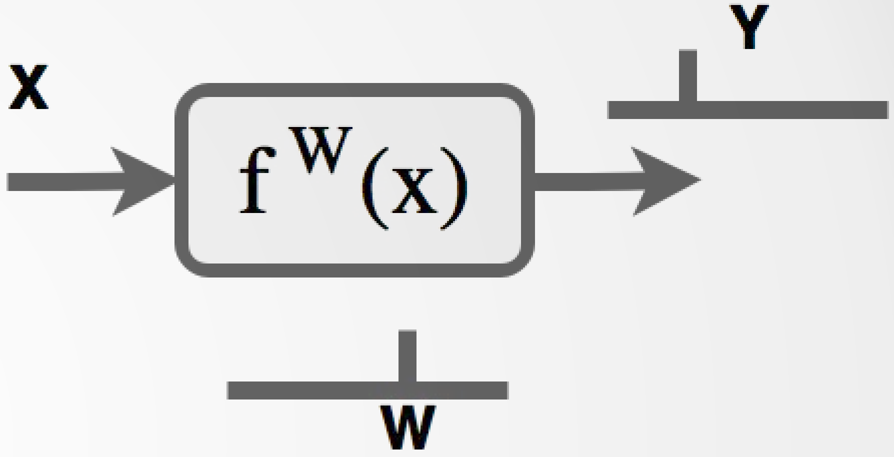
Bayesian Neural Networks

- **Neural Network**
  - Finds a function y = f(x)



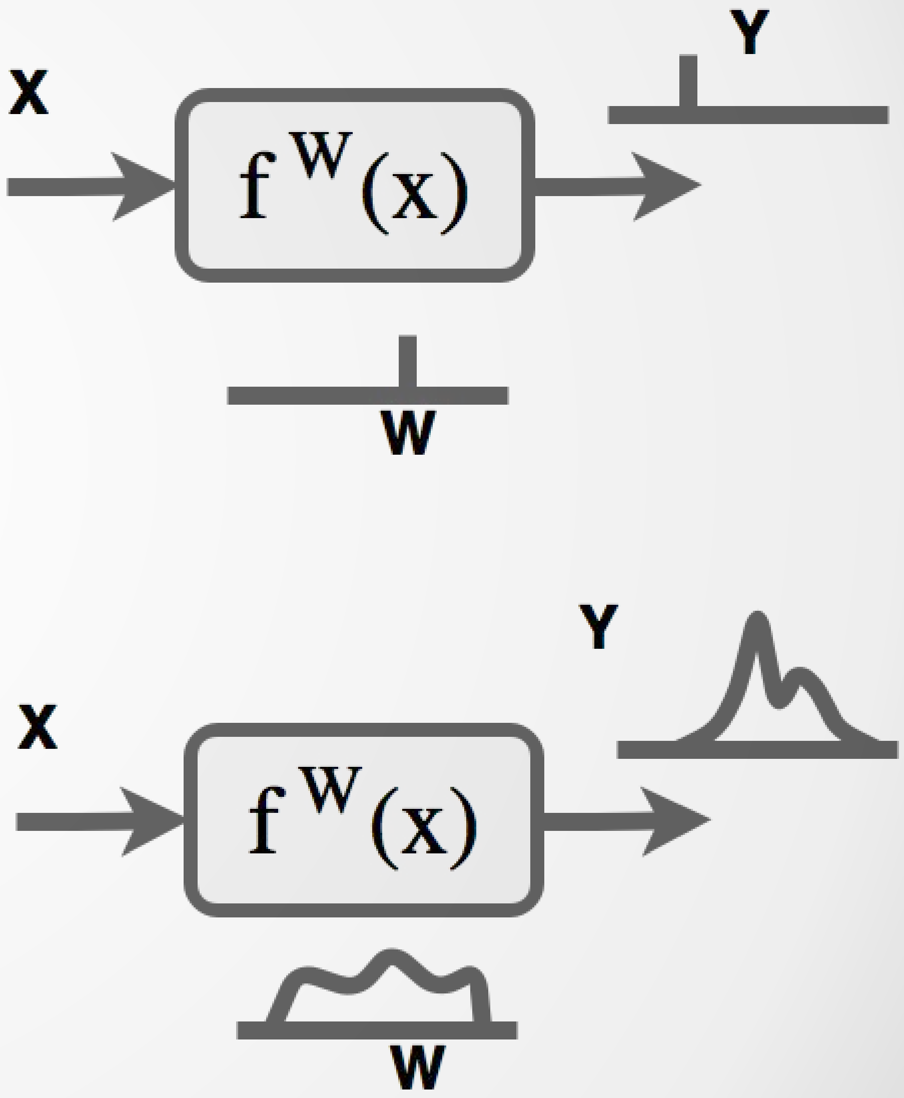
Bayesian Neural Networks

- **Neural Network**
  - Finds a function y = f(x)
  - One best model



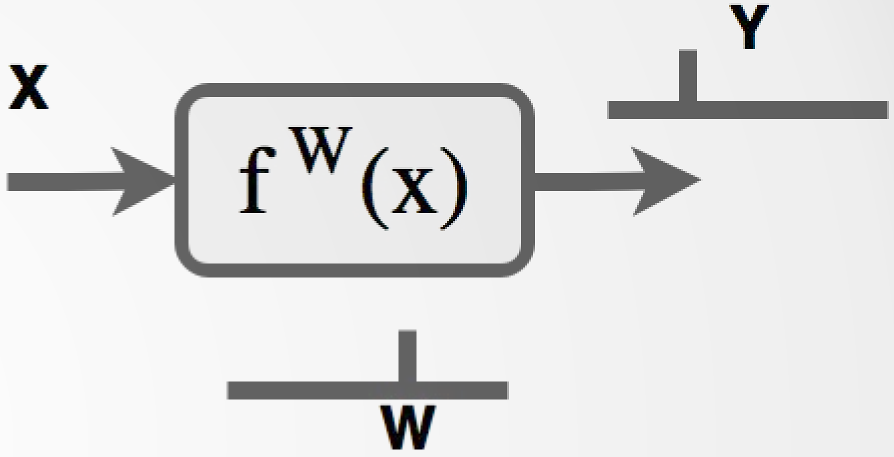
Bayesian Neural Networks

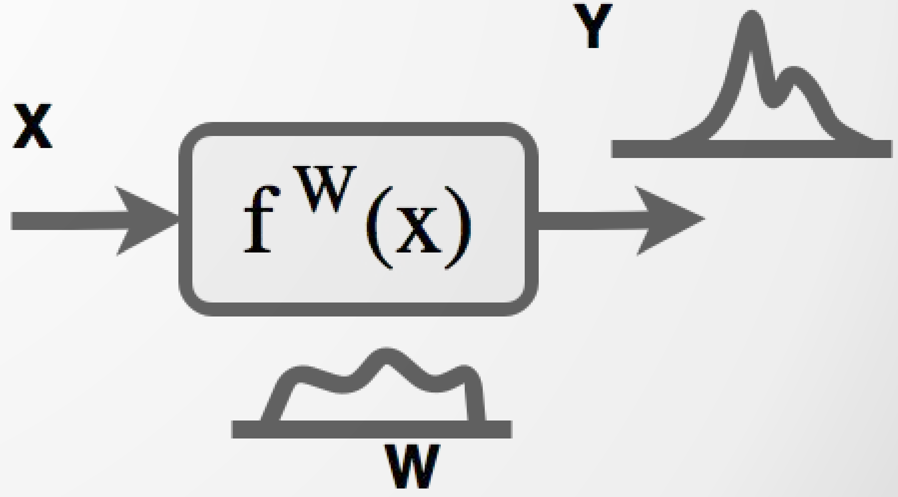
- Neural Network
  - Finds a function y = f(x)
  - One best model
- **Bayesian Neural Network**



Bayesian Neural Networks

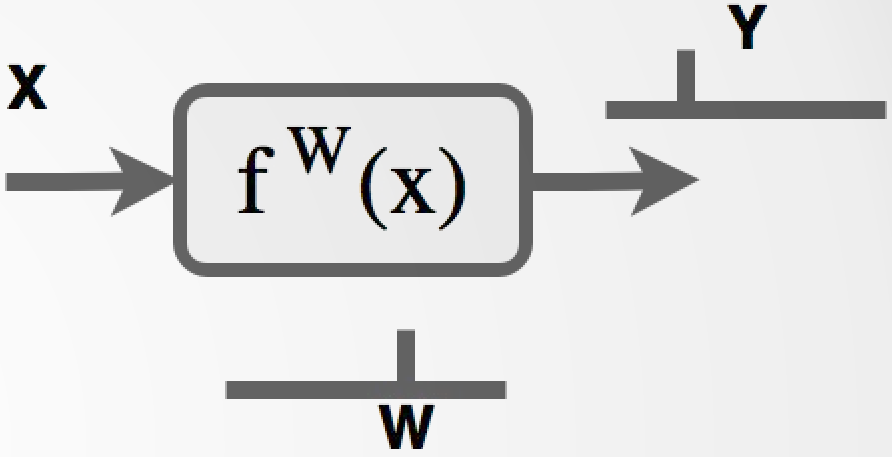
- Neural Network
  - Finds a function y = f(x)
  - One best model
- **Bayesian Neural Network**
  - Distribution over weights

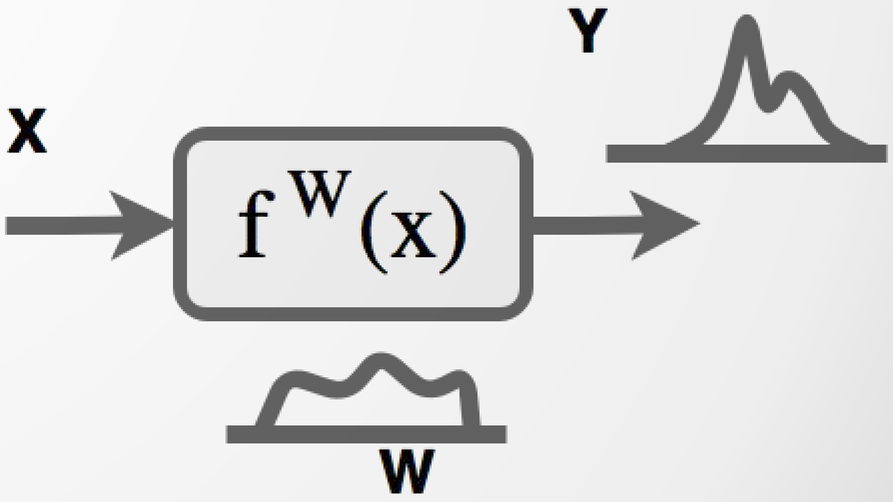




Bayesian Neural Networks

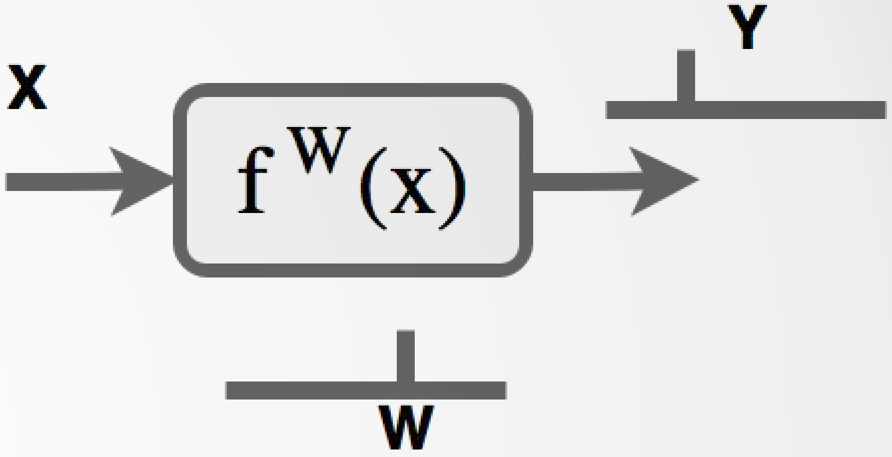
- Neural Network
  - Finds a function y = f(x)
  - One best model
- **Bayesian Neural Network**
  - Distribution over weights
  - Output is a distribution

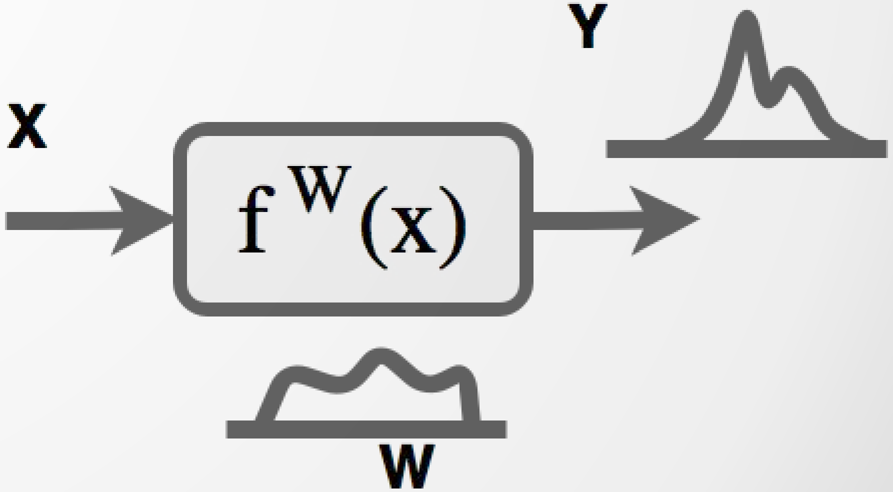




Bayesian Neural Networks

- Neural Network
  - Finds a function y = f(x)
  - One best model
- **Bayesian Neural Network**
  - Distribution over weights
  - Output is a distribution
  - Many models within a network

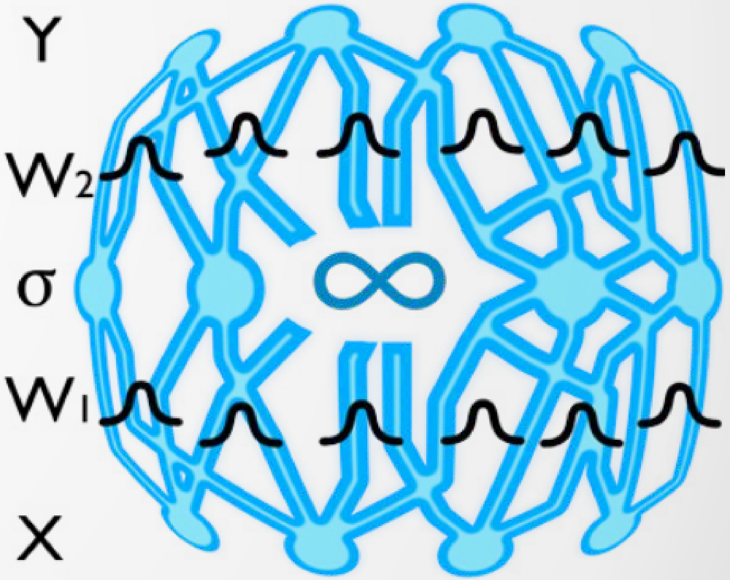




Bayesian Neural Networks

Bayesian Neural Networks

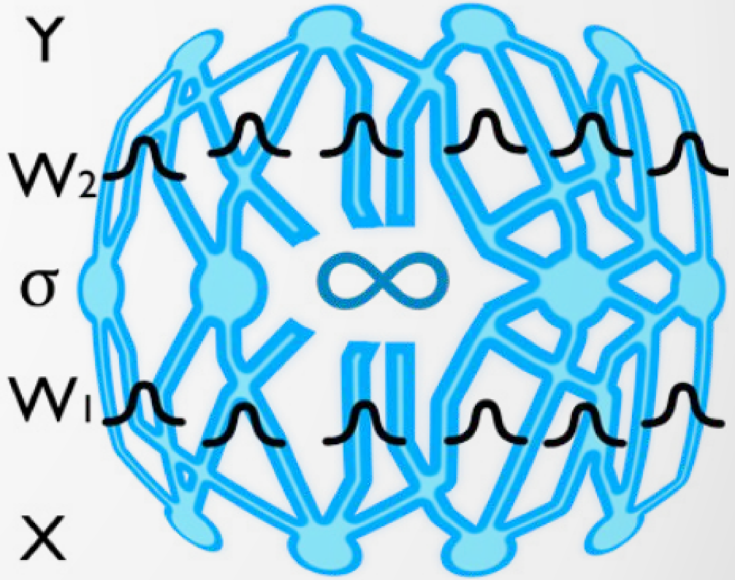
- **Ideas from 30 years ago**



Source:Gal, Y.
http://mlg.eng.cam.ac.uk/yarin/index.html

Bayesian Neural Networks

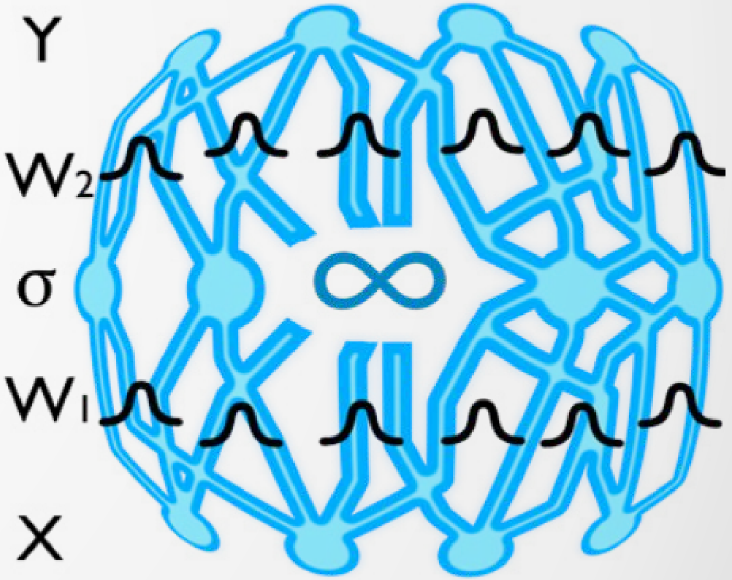
- Ideas from 30 years ago
- **BNN with ∞ many weights → Gaussian Process**



Source:Gal, Y.
http://mlg.eng.cam.ac.uk/yarin/index.html

Bayesian Neural Networks

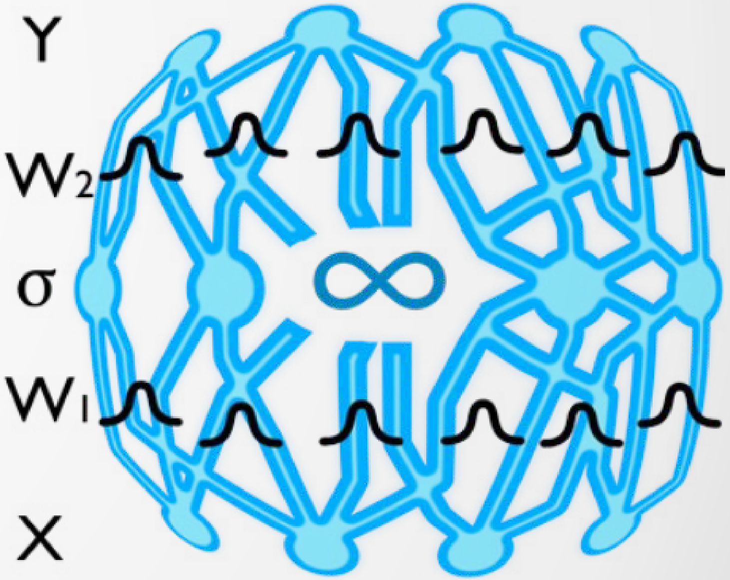
- Ideas from 30 years ago
- BNN with ∞ many weights → Gaussian Process
- **Difficult to make inference on:**
    - Multimodal correlated distribution
    - Nonlinearities



Source:Gal, Y.
http://mlg.eng.cam.ac.uk/yarin/index.html

Bayesian Neural Networks

- Ideas from 30 years ago
- BNN with ∞ many weights → Gaussian Process
- Difficult to make inference on:
  - Multimodal correlated distribution
  - Nonlinearities
- **Different approximation techniques**



Source:Gal, Y.
http://mlg.eng.cam.ac.uk/yarin/index.html

Bayesian Neural Networks

- **Find** $p(W|X, Y)$

Bayesian Neural Networks

- **Find** $p(W|X,Y)$
  - Find models that are likely to generate our training data

Bayesian Neural Networks

- **Find** $p(W|X,Y)$
  - Find models that are likely to generate our training data

$$p(W|X,Y) = \frac{p(Y|X,W)p(W)}{p(Y|X)}$$

Bayesian Neural Networks

- **Likelihood (Gaussian, Laplace)**
  - Measures how likely model with weights W generated Y

$$p(W|X,Y) = \frac{\boxed{p(Y|X,W)}\,p(W)}{p(Y|X)}$$

Bayesian Neural Networks

- **Likelihood (Gaussian, Laplace)**
  - Measures how likely model with weights W generated Y

$$p(y|f^W(x)) = \mathcal{N}(f^W(x), \sigma^2)$$

$$p(W|X,Y) = \frac{\boxed{p(Y|X,W)}p(W)}{p(Y|X)}$$

Bayesian Neural Networks

- **Prior**
  - Usually a Gaussian distribution with mean at 0

$$p(W|X,Y) = \frac{p(Y|X,W)\boxed{p(W)}}{p(Y|X)}$$

Bayesian Neural Networks

- **Prior**
  - Usually a Gaussian distribution with mean at 0
  - Acts as a regularizer

$$p(W|X,Y) = \frac{p(Y|X,W)\boxed{p(W)}}{p(Y|X)}$$

Bayesian Neural Networks

- **Marginal Probability**
  - Normalizes probability

$$p(W|X,Y) = \frac{p(Y|X,W)p(W)}{\boxed{p(Y|X)}}$$

Bayesian Neural Networks

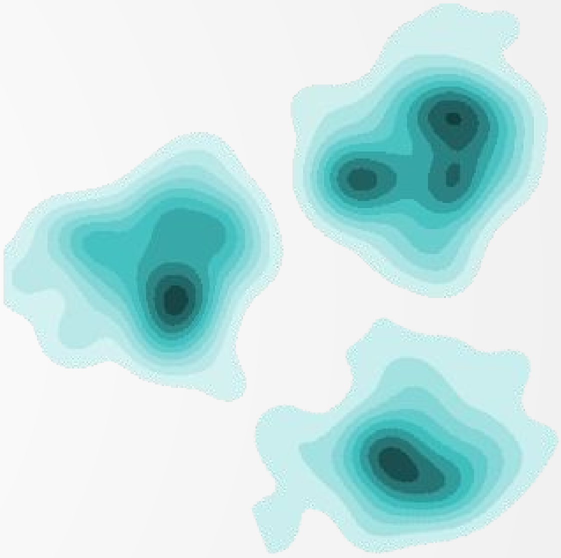
- **Marginal Probability**
  - Normalizes probability
  - Can not be evaluated

$$p(W|X,Y) = \frac{p(Y|X,W)p(W)}{\boxed{p(Y|X)}}$$

$$\int p(Y|X,W)p(W)dW$$

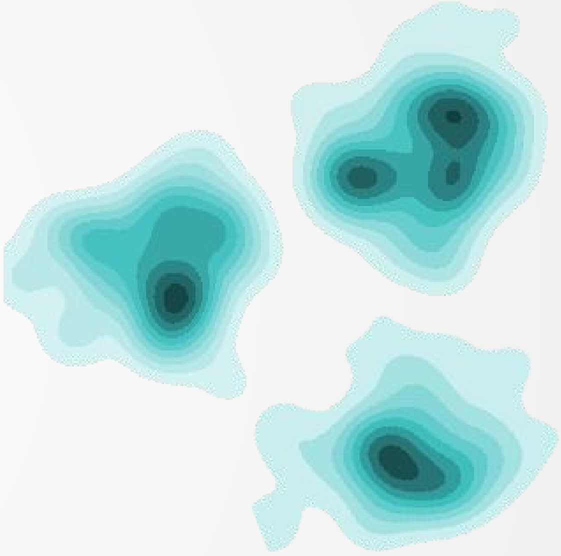
Bayesian Neural Networks

- **In general** $p(W|X,Y)$ **can be a complex distribution**



Bayesian Neural Networks

- In general $p(W|X,Y)$ can be a complex distribution
- **We need an approximation**

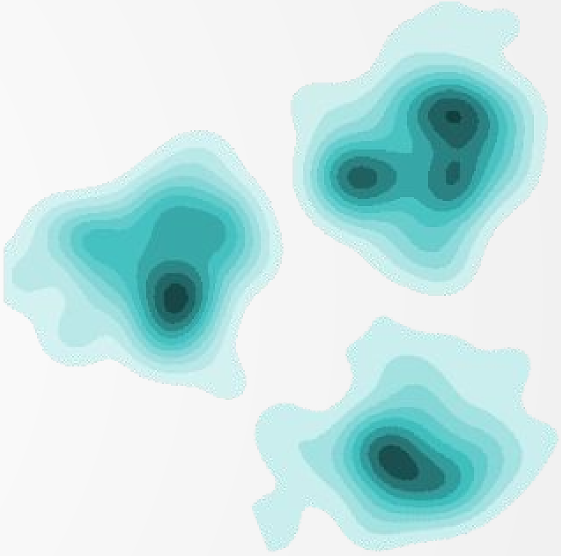


Bayesian Neural Networks

- In general $p(W|X,Y)$ can be a complex distribution
- We need an approximation
- **Replace complex $p(W|X,Y)$**

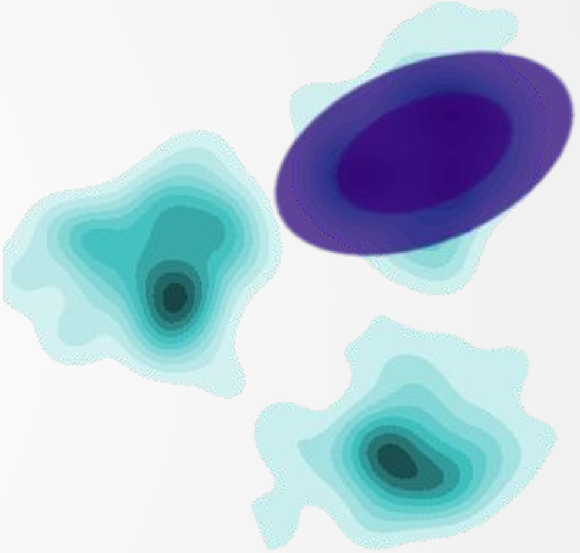
with $q^*_\theta(W)$ from a tractable family

(Gaussian, Bernoulli)



Bayesian Neural Networks

- In general $p(W|X,Y)$ can be a complex distribution
- We need an approximation
- Approximate complex $p(W|X,Y)$ with $q^*_\theta(W)$ from a tractable family (Gaussian, Bernoulli)



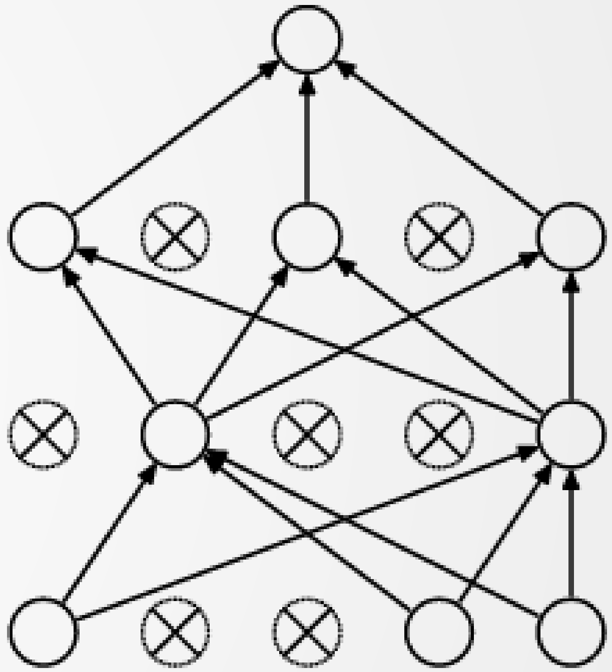
- **Minimize the distance between them (KL Divergence)**

Outline:

- Motivation
- Types of Uncertainty
- Bayesian Neural Networks
- **Dropout Variational Inference**
- Modeling uncertainties
- Experiments
- Results Analysis
- Summary

Dropout Variational Inference

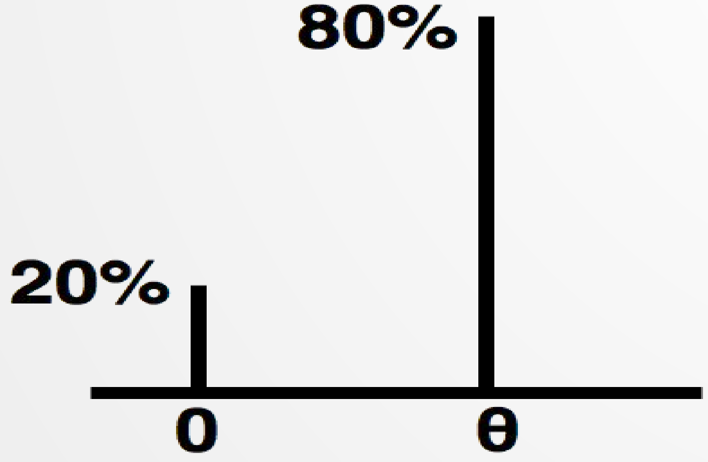
- **Randomly drop network units**

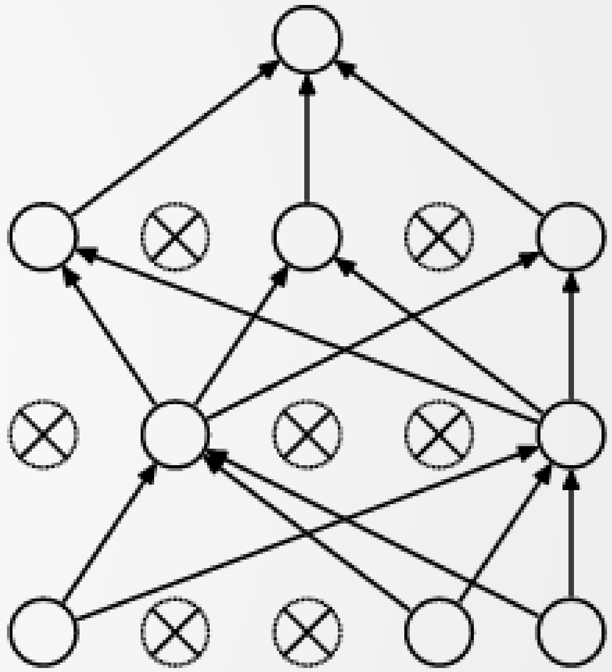


Source: “Dropout: A Simple Way to Prevent Neural Networks from Overfitting” Srivastava et. al.

Dropout Variational Inference

- Randomly drop network units
- **Bernoulli approximation** $q_\theta^*(W)$

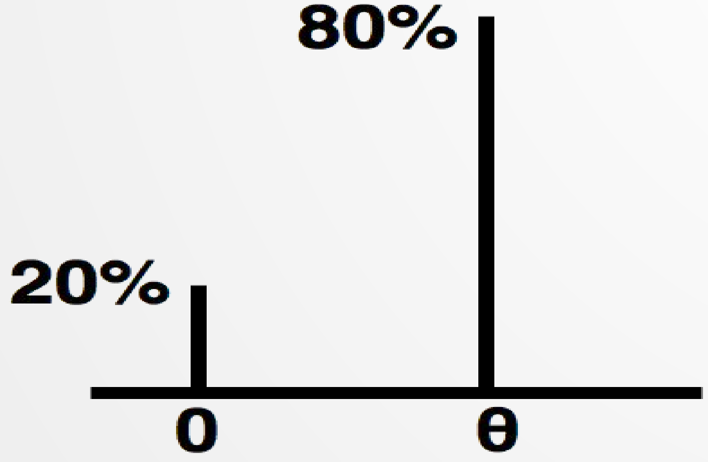


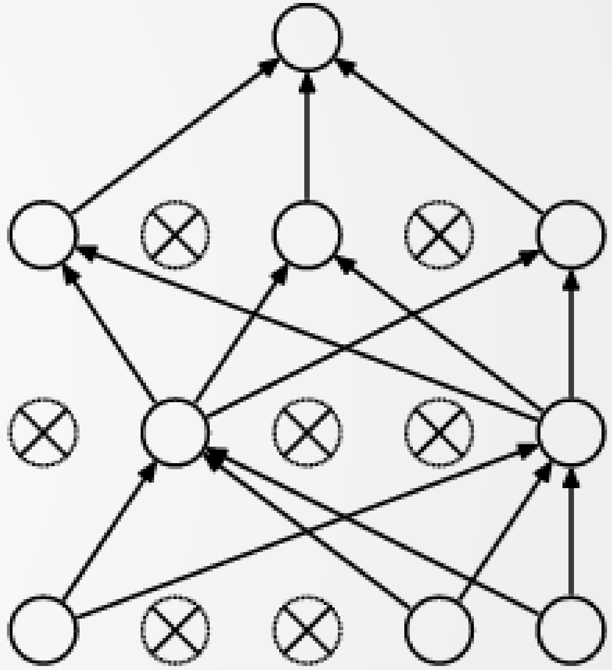


Source: “Dropout: A Simple Way to Prevent Neural Networks from Overfitting” Srivastava et. al.

Dropout Variational Inference

- Randomly drop network units
- Bernoulli approximation $q^*_\theta(W)$
- **One of the simplest possible**

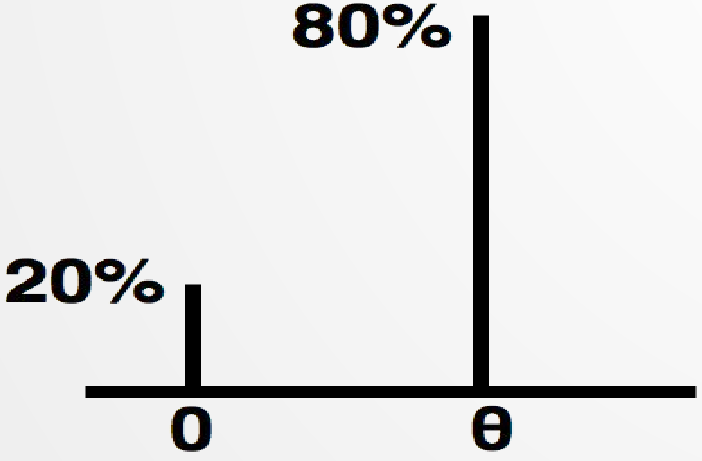


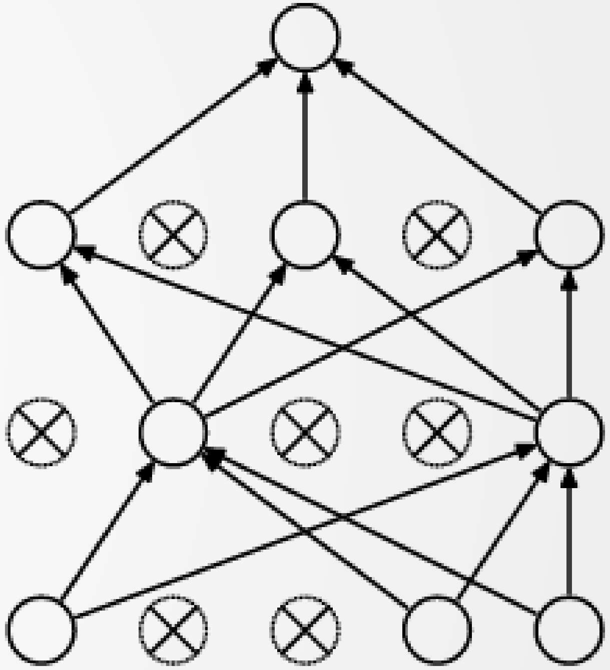


Source: “Dropout: A Simple Way to Prevent Neural Networks from Overfitting” Srivastava et. al.

Dropout Variational Inference

- Randomly drop network units
- Bernoulli approximation $q_{\theta}^{*}(W)$
- One of the simplest possible
- **We learn θ for each weight**





Source: "Dropout: A Simple Way to Prevent Neural Networks from Overfitting" Srivastava et. al.

Training with dropout

- **Neural network trained with dropout is already a BNN**

Training with dropout

- **Neural network trained with dropout is already a BNN**
    - Because we have a distribution over its weights

Training with dropout

- **Neural network trained with dropout is already a BNN**

$$L(\theta, p) = -\frac{1}{N}\sum_{i=1}^{N} \log p(y_i | f^{\hat{W}_i}(x_i)) + \frac{1-p}{2N}||\theta||^2$$

Training with dropout

- Neural network trained with dropout is already a BNN

$$L(\theta, p) = -\frac{1}{N}\sum_{i=1}^{N} \log p(y_i | \boxed{f^{\hat{W}_i}(x_i)}) + \frac{1-p}{2N}||\theta||^2$$

For each training point

draw a new dropout mask

Training with dropout

- Neural network trained with dropout is already a BNN

$$L(\theta, p) = \boxed{-\frac{1}{N}\sum_{i=1}^{N}\log p(y_i|f^{\hat{W}_i}(x_i))} + \frac{1-p}{2N}||\theta||^2$$

Minimize negative log likelihood

Training with dropout

- Neural network trained with dropout is already a BNN

$$L(\theta, p) = -\frac{1}{N}\sum_{i=1}^{N} \log p(y_i | f^{\hat{W}_i}(x_i)) + \boxed{\frac{1-p}{2N}||\theta||^2}$$

Regularization term (prior)

Training with dropout

- Neural network trained with dropout is already a BNN

$$L(\theta, p) = -\frac{1}{N}\sum_{i=1}^{N} \log p(y_i | f^{\hat{W}_i}(x_i)) + \frac{1-p}{2N}||\theta||^2$$

- **In this work dropout probability *p* is set to 0.2**

Training with dropout

- Neural network trained with dropout is already a BNN

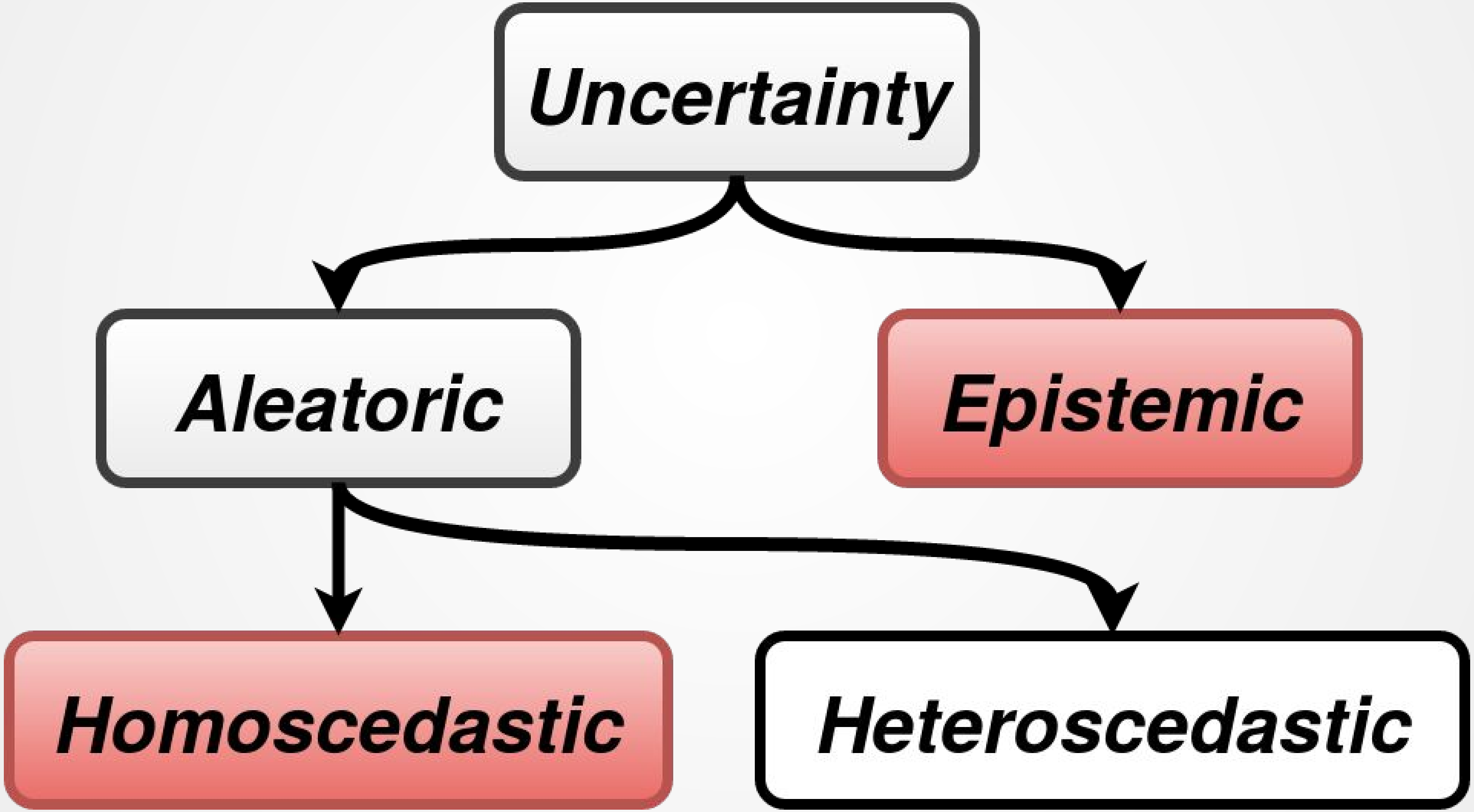
$$L(\theta, p) = -\frac{1}{N}\sum_{i=1}^{N}\log p(y_i | f^{\hat{W}_i}(x_i)) + \frac{1-p}{2N}||\theta||^2$$

- In this work dropout probability ***p*** is set to 0.2
- **Minimizing this loss we also minimize KL divergence between** $p(W|X,Y)$ **and** $q_\theta^*(W)$

Outline:

- Motivation
- Types of Uncertainty
- Bayesian Neural Networks
- Dropout Variational Inference
- **Modeling uncertainties**
- Experiments
- Results Analysis
- Summary

Modeling uncertainties



Homoscedastic uncertainty

- **Regression Problem**

Homoscedastic uncertainty

- **Regression Problem**

$$L(\theta, p) = -\frac{1}{N}\sum_{i=1}^{N}\log p(y_i|f^{\hat{W}_i}(x_i)) + \frac{1-p}{2N}||\theta||^2$$

Homoscedastic uncertainty

- Regression Problem

$$L(\theta, p) = \boxed{-} \frac{1}{N} \sum_{i=1}^{N} \boxed{\log p(y_i | f^{\hat{W}_i}(x_i))} + \frac{1-p}{2N} ||\theta||^2$$

- **If Laplace Likelihood:** $\boxed{\frac{1}{\sigma^2} ||y_i - f^{\hat{W}_i}(x_i)|| + log\sigma^2}$

Homoscedastic uncertainty

- Regression Problem

$$L(\theta, p) = \boxed{-}\frac{1}{N}\sum_{i=1}^{N}\boxed{\log p(y_i|f^{\hat{W}_i}(x_i))} + \frac{1-p}{2N}||\theta||^2$$

- **If Laplace Likelihood:** $\boxed{\frac{1}{\sigma^2}||y_i - f^{\hat{W}_i}(x_i)|| + log\sigma^2}$
    - Minimize the distance between model predictions and the training data

Homoscedastic uncertainty

- Regression Problem

$$L(\theta, p) = \boxed{-}\frac{1}{N}\sum_{i=1}^{N}\boxed{\log p(y_i | f^{\hat{W}_i}(x_i))} + \frac{1-p}{2N}||\theta||^2$$

- If Laplace Likelihood

$$\frac{1}{\boxed{\sigma^2}}||y_i - \boxed{f^{\hat{W}_i}(x_i)}|| + log\boxed{\sigma^2}$$

- **Use sigma and dropout samples to estimate uncertainty**

Estimated variance

- **Sum of aleatoric and epistemic variance**

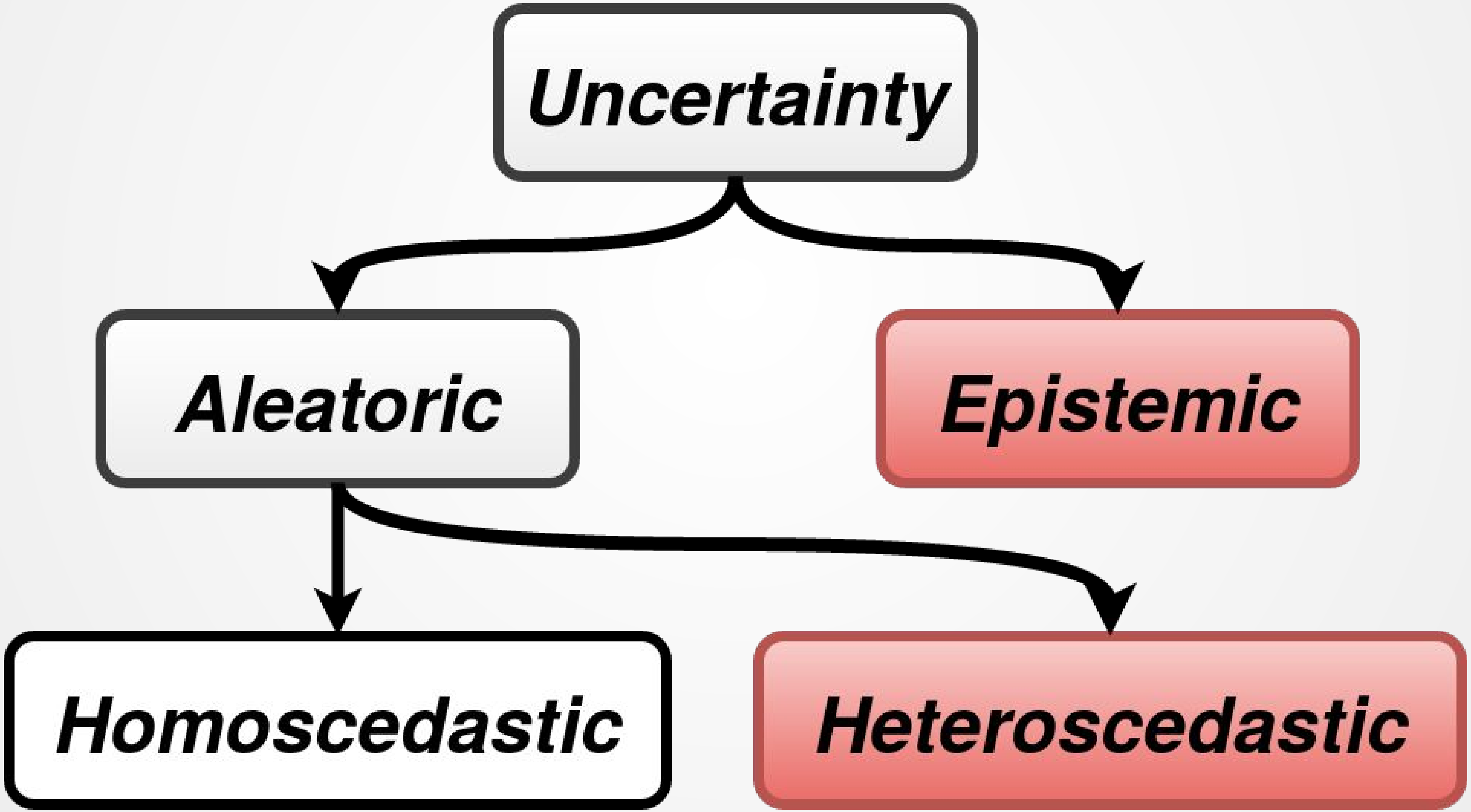
Estimated variance

- **Sum of aleatoric and epistemic variance**
    - Epistemic variance: variance over multiple dropout draws

$$Var(y) \approx \sigma^2 + \frac{1}{T}\sum_{t=1}^{T}(f^{\hat{W}_t}(x) - E(y))^2$$

$$E(y) \approx \frac{1}{T}\sum_{t=1}^{T} f^{\hat{W}_t}(x)$$

Heteroscedastic uncertainty



Heteroscedastic uncertainty

- **Start as before**

$$\frac{1}{\sigma^2}||y_i - f^{\hat{W}_i}(x_i)|| + log\sigma^2$$

Heteroscedastic uncertainty

- Start as before

$$\frac{1}{\sigma^2}||y_i - f^{\hat{W}_i}(x_i)|| + log\sigma^2$$

- **Uncertainty as a function of the input**

$$\frac{1}{\sigma(x_i)^2}||y_i - f^{\hat{W}_i}(x_i)|| + log\sigma(x_i)^2$$

Heteroscedastic uncertainty

- Start as before

$$\frac{1}{\boxed{\sigma^2}}||y_i - f^{\hat{W}_i}(x_i)|| + log\boxed{\sigma^2}$$

- **Uncertainty as a function of the input**

$$\frac{1}{\boxed{\sigma(x_i)^2}}||y_i - f^{\hat{W}_i}(x_i)|| + log\boxed{\sigma(x_i)^2}$$

  - If it is hard to predict correct output, increase uncertainty to reduce loss
  - Uncertainty acts as a loss attenuation
  - Robust to outliers

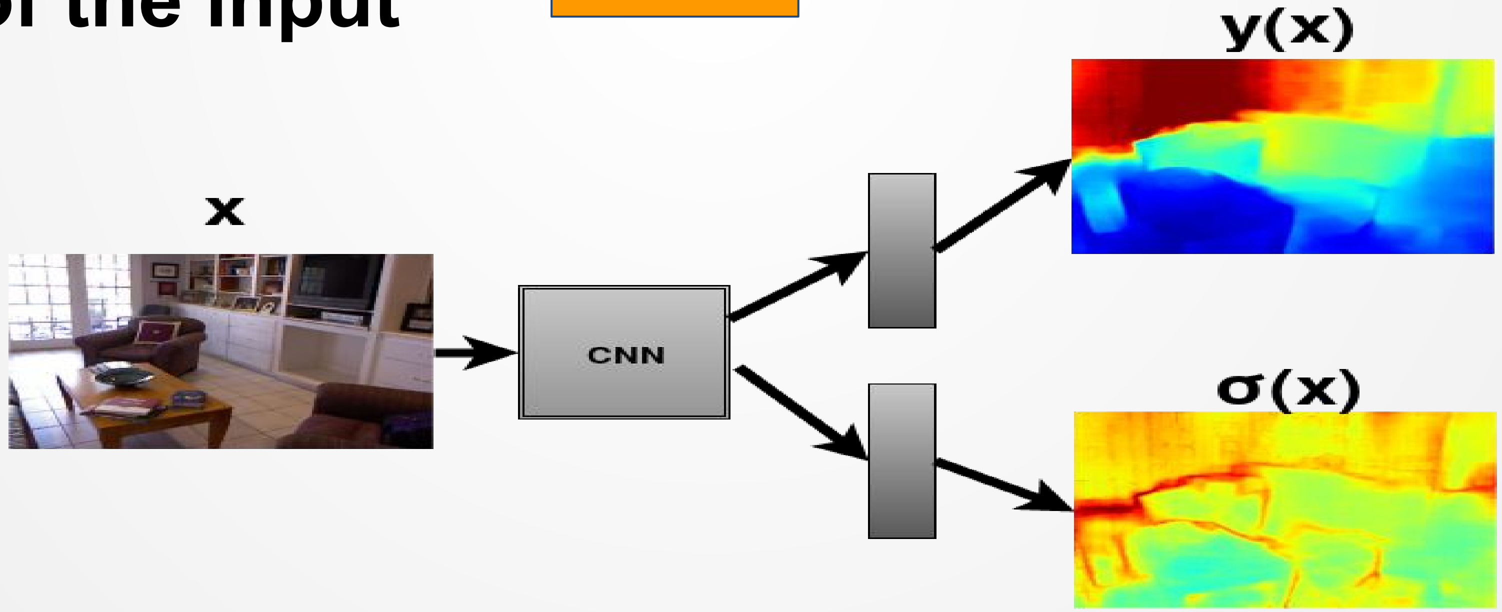
Heteroscedastic uncertainty

- Start as before

$$\frac{1}{\boxed{\sigma^2}}||y_i - f^{\hat{W}_i}(x_i)|| + log\boxed{\sigma^2}$$

- **Uncertainty as a function of the input**

$$\frac{1}{\boxed{\sigma(x_i)^2}}||y_i - f^{\hat{W}_i}(x_i)|| + log\boxed{\sigma(x_i)^2}$$



Classification with Uncertainty

- **Modeling uncertainty for classification**

Classification with Uncertainty

- **Modeling uncertainty for classification**
  - Add noise to the output of the network (logits)

Classification with Uncertainty

- **Modeling uncertainty for classification**
  - Add noise to the output of the network (logits)
  - Variance of this noise depends on the input

Classification with Uncertainty

- **Modeling uncertainty for classification**
  - Add noise to the output of the network (logits)
  - Variance of this noise depends on the input

$$\hat{\mathbf{x}}_{i,t} = \boxed{\mathbf{f}_i^{\mathbf{W}}} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \boxed{(\sigma_i^{\mathbf{W}}}\,)^2)$$

$$\mathcal{L}_x = \frac{1}{T} \sum_{i,t} (-\hat{x}_{i,t,c} + \log \sum_{c'} \exp \hat{x}_{i,t,c'})$$

Classification with Uncertainty

- Modeling uncertainty for classification
  - Add noise to the output of the network (logits)
  - Variance of this noise depends on the input

$$\hat{\mathbf{x}}_{i,t} = \mathbf{f}_i^{\mathbf{W}} + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, (\sigma_i^{\mathbf{W}})^2)$$

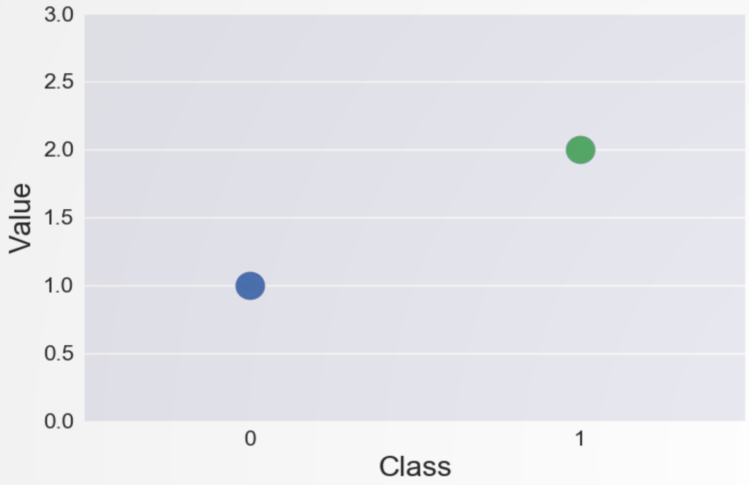
$$\mathcal{L}_x = \frac{1}{T} \sum_{i,t} (-\hat{x}_{i,t,c} + \log \sum_{c'} \exp \hat{x}_{i,t,c'})$$

- **If model is wrong, bigger uncertainty results in a lower loss**

Classification with Uncertainty

- **Example**

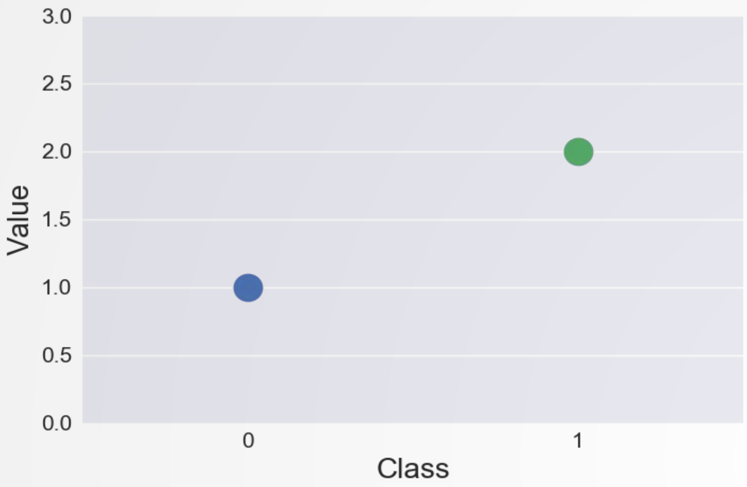
Classification with Uncertainty

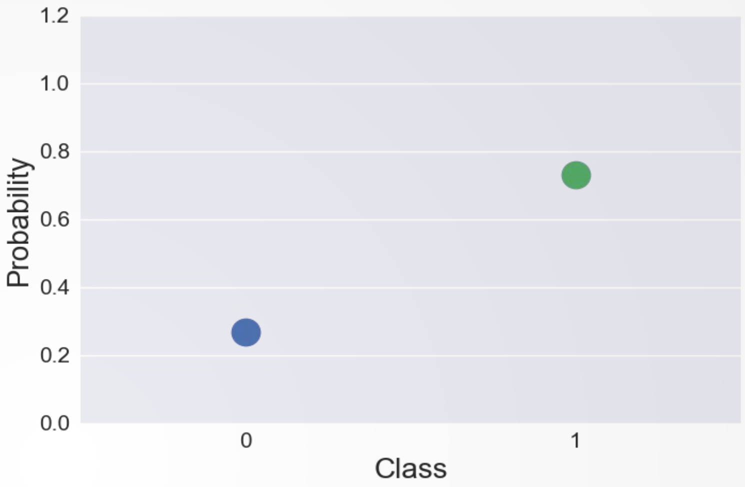


Network outputs:

- First class: 1
- Second class: 2

Classification with Uncertainty



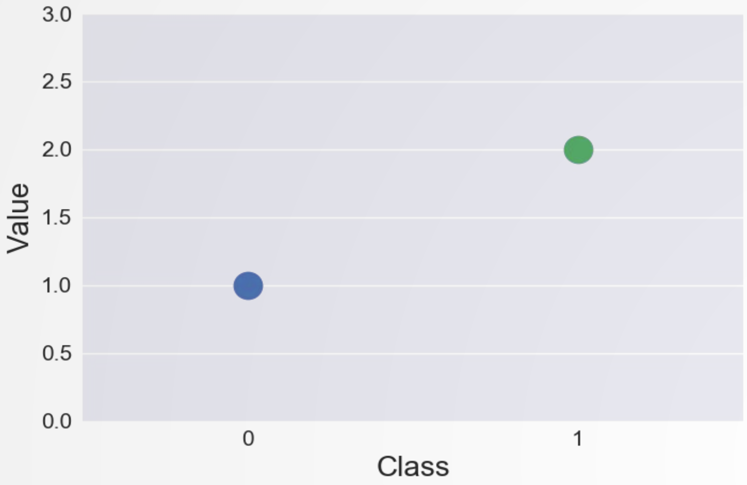


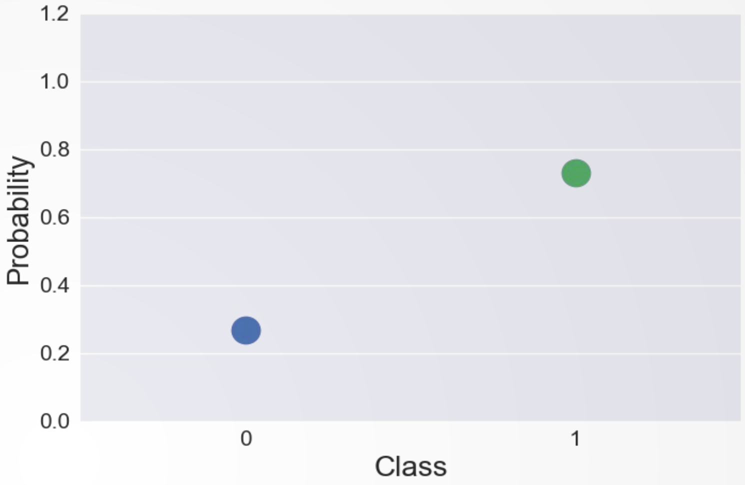
After softmax:

First class: 27%

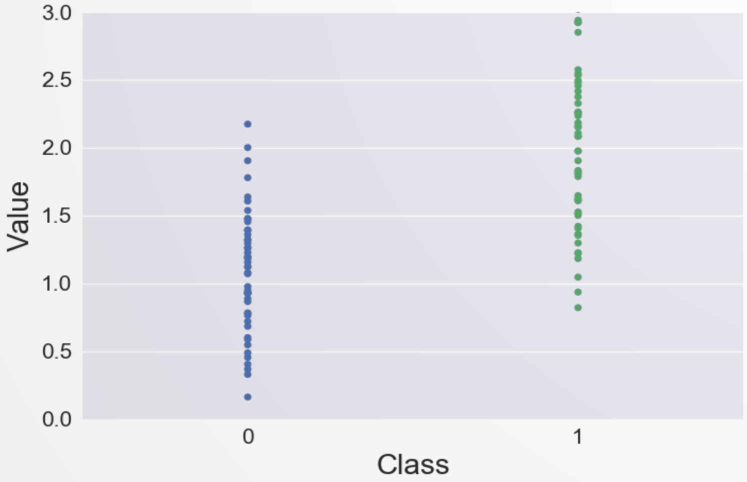
Second class: 73%

Classification with Uncertainty



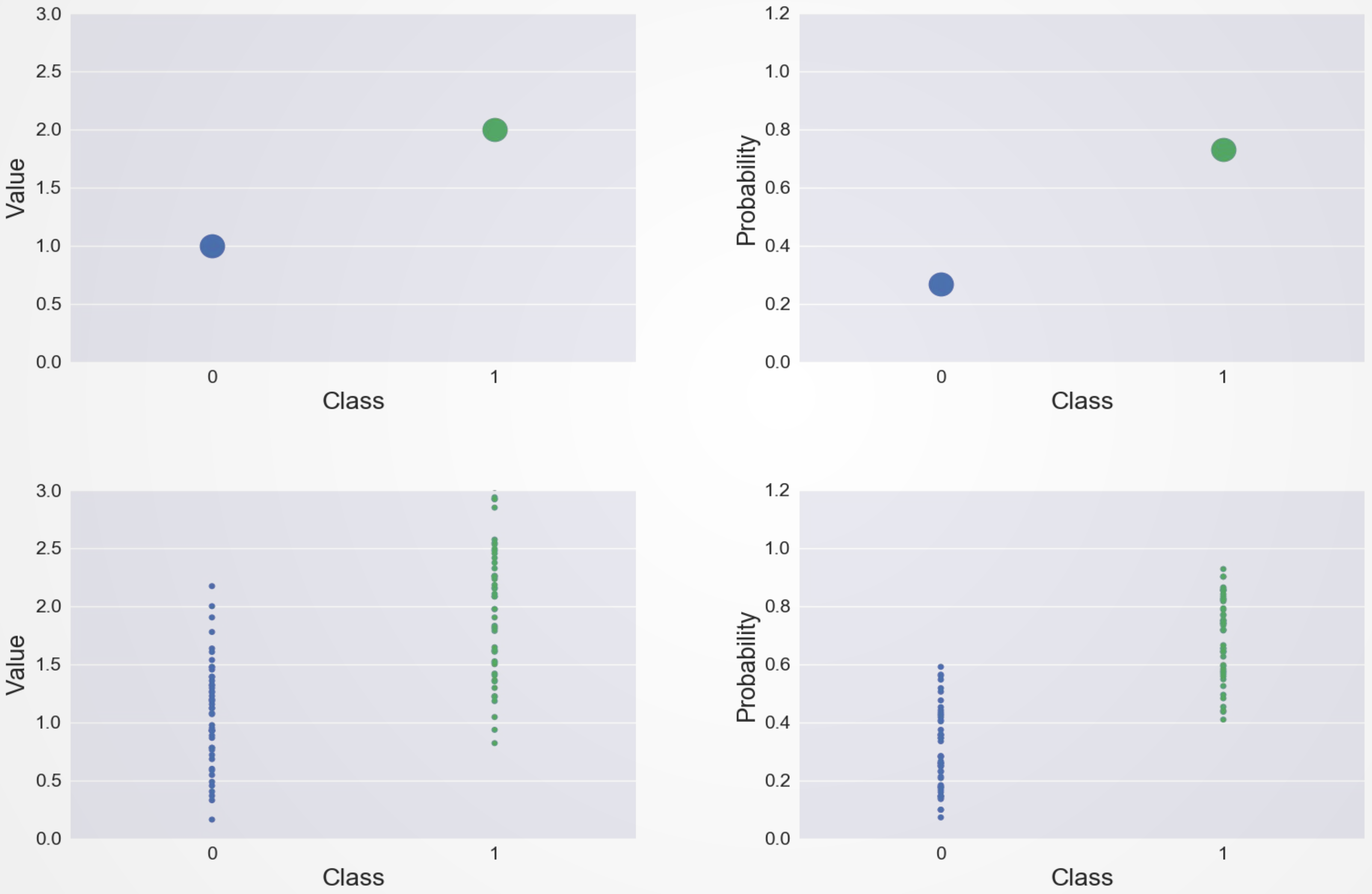


27%
73%



Sample 50 times logits with noise (variance = 0.5)

Classification with Uncertainty



27%
73%

30%
70%

Outline:

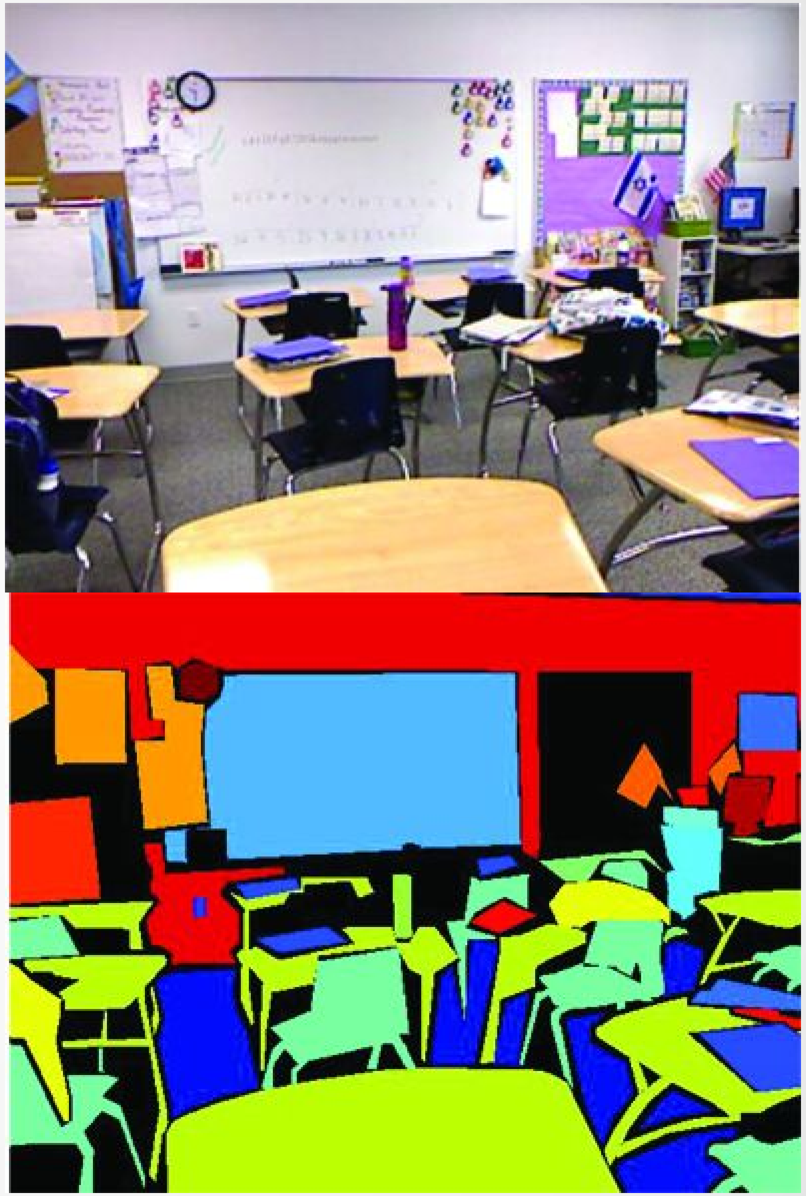
- Motivation
- Types of Uncertainty
- Bayesian Neural Networks
- Dropout Variational Inference
- Modeling uncertainties
- **Experiments**
- Results Analysis
- Summary

CamVid

- Road scene understanding dataset
- 367 train, 233 test
- Day and dusk scenes
- 11 classes
- Resized to 360×480

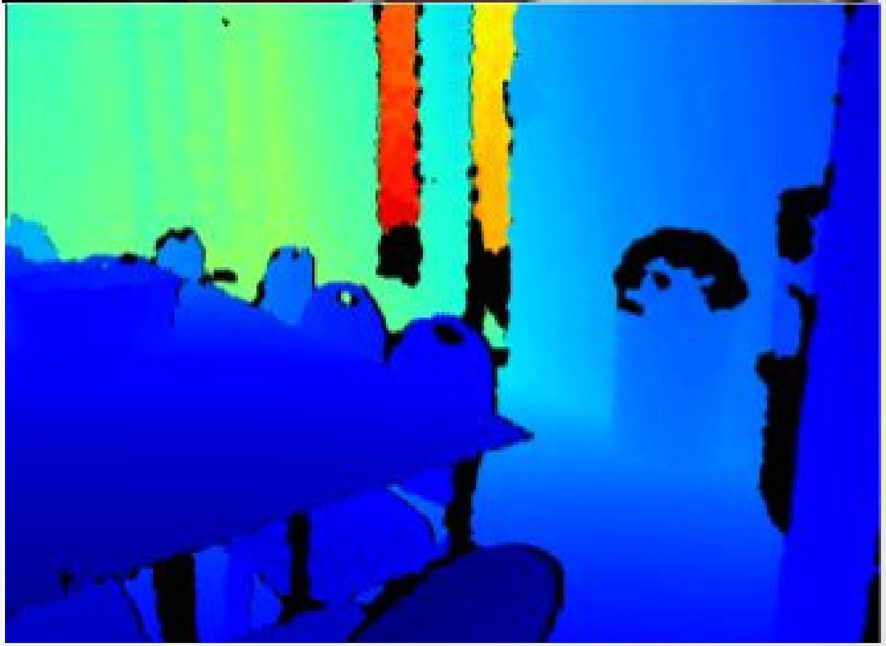
NYUv2 40-Class

- Indoor segmentation dataset
- 40 different semantic classes
- 464 different indoor scenes.
- 1449 images
- 640×480



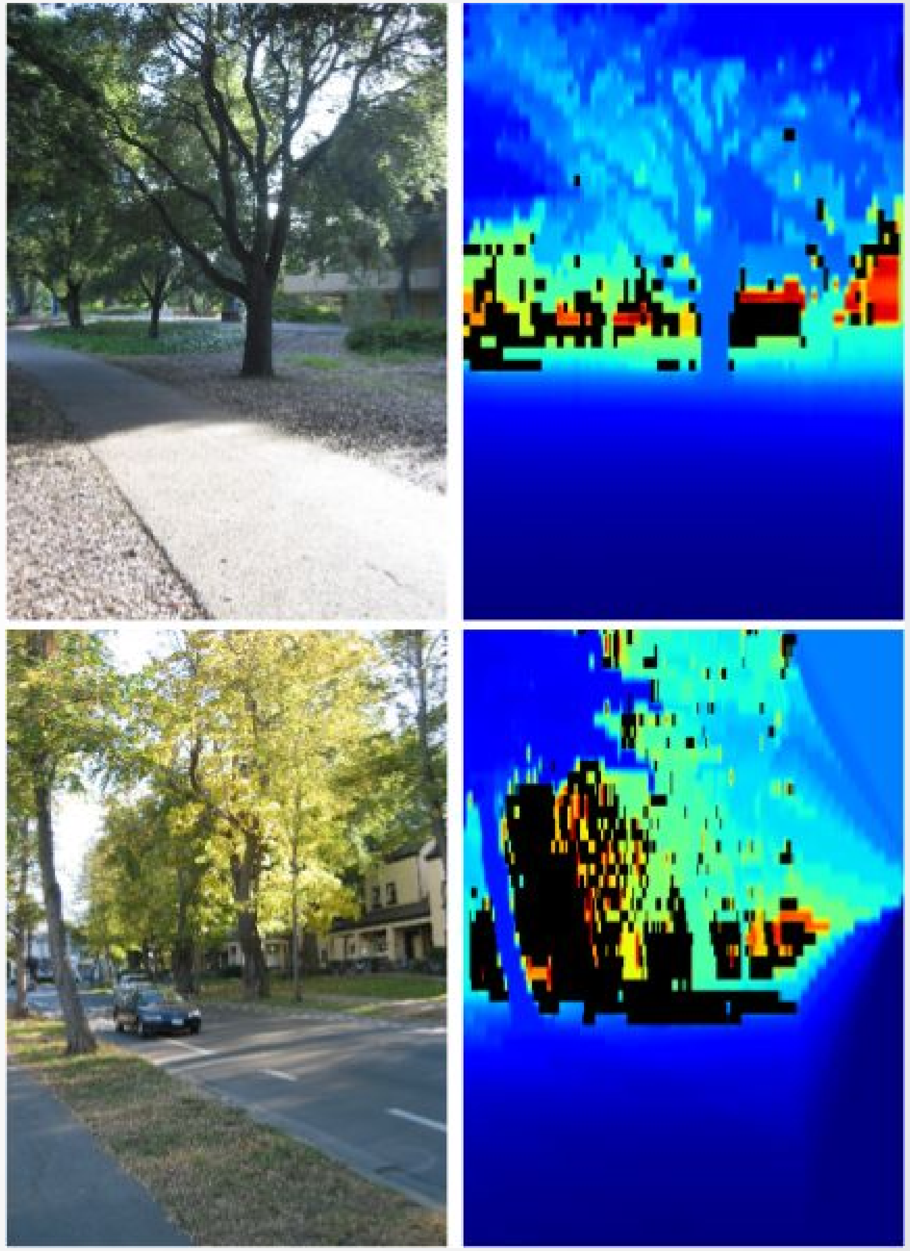
NYUv2 Depth

- Indoor dataset
- 464 different indoor scenes.
- 1449 images
- 640×480



Make 3D

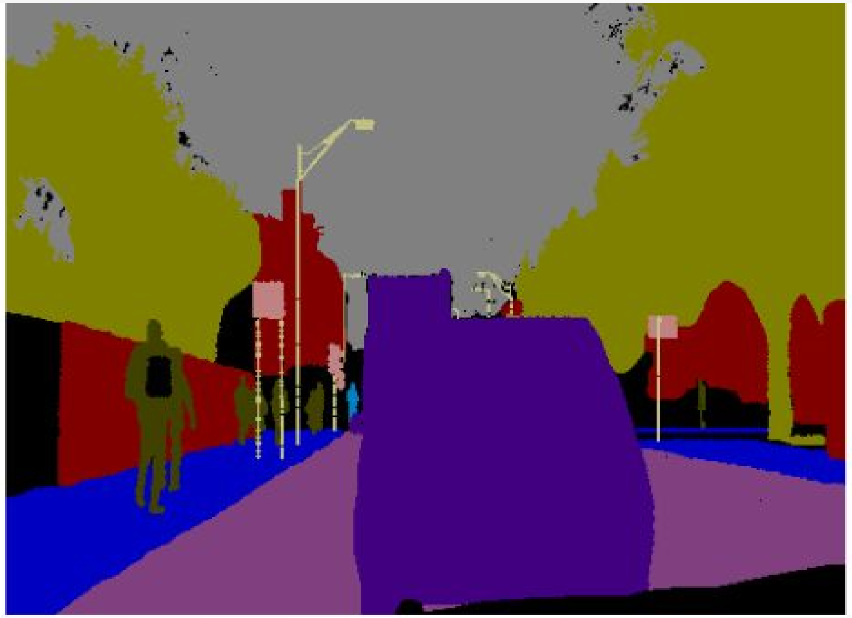
- 400 training, 134 test
- 3-D laser scanner.
- Resized to 345×460

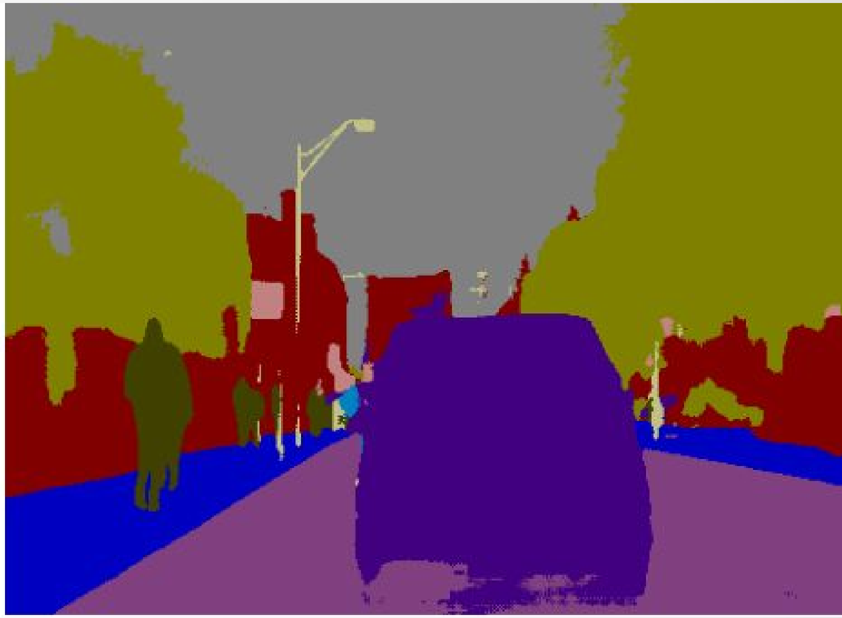


Outline:

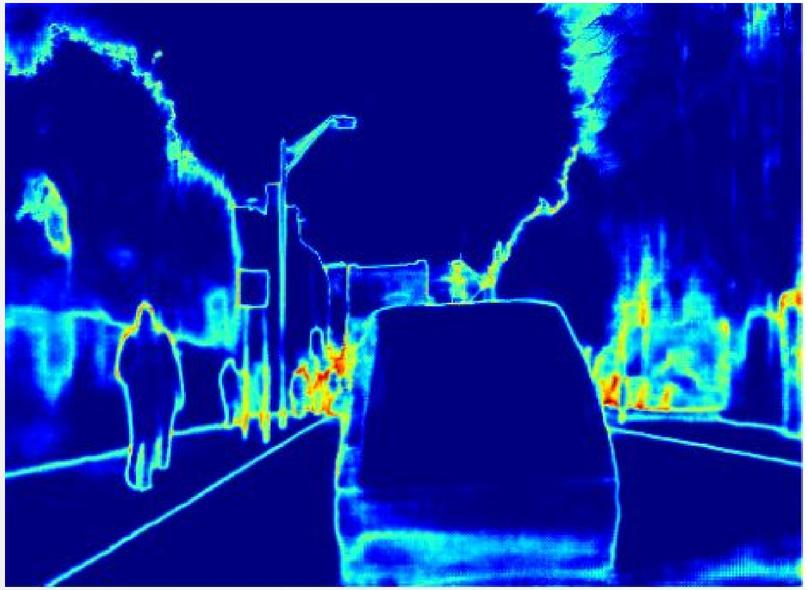
- Motivation
- Types of Uncertainty
- Bayesian Neural Networks
- Dropout Variational Inference
- Modeling uncertainties
- Experiments
- **Results Analysis**
- Summary

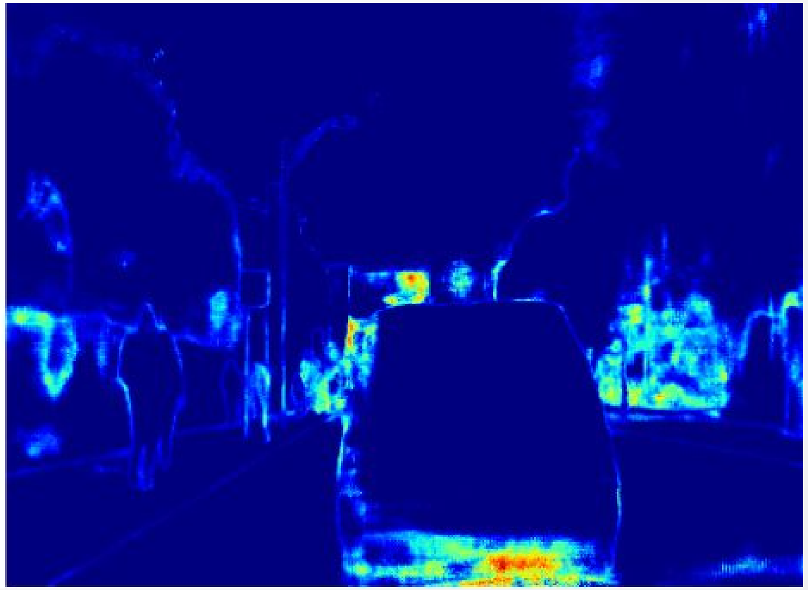
Semantic Segmentation





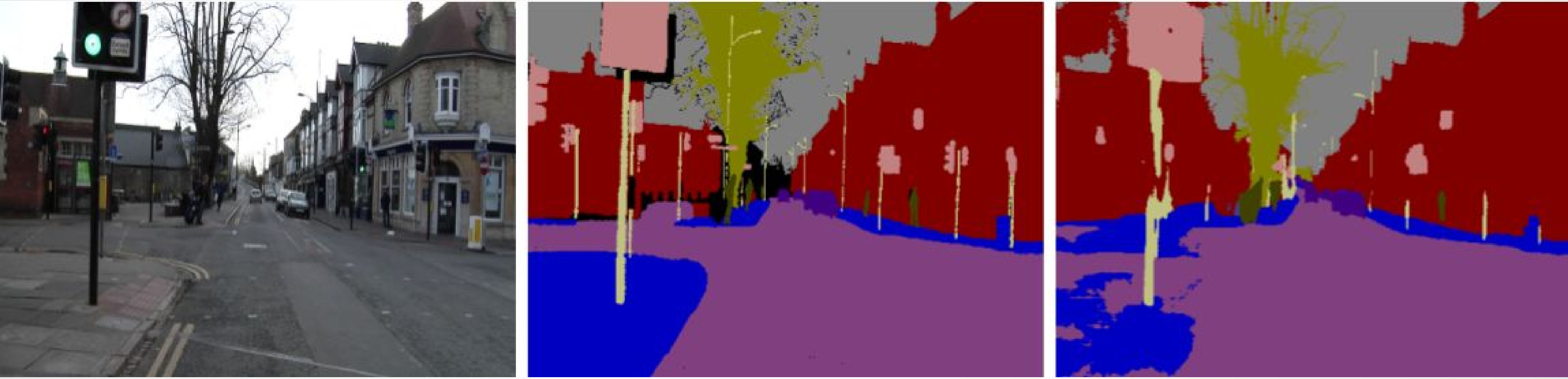
Aleatoric



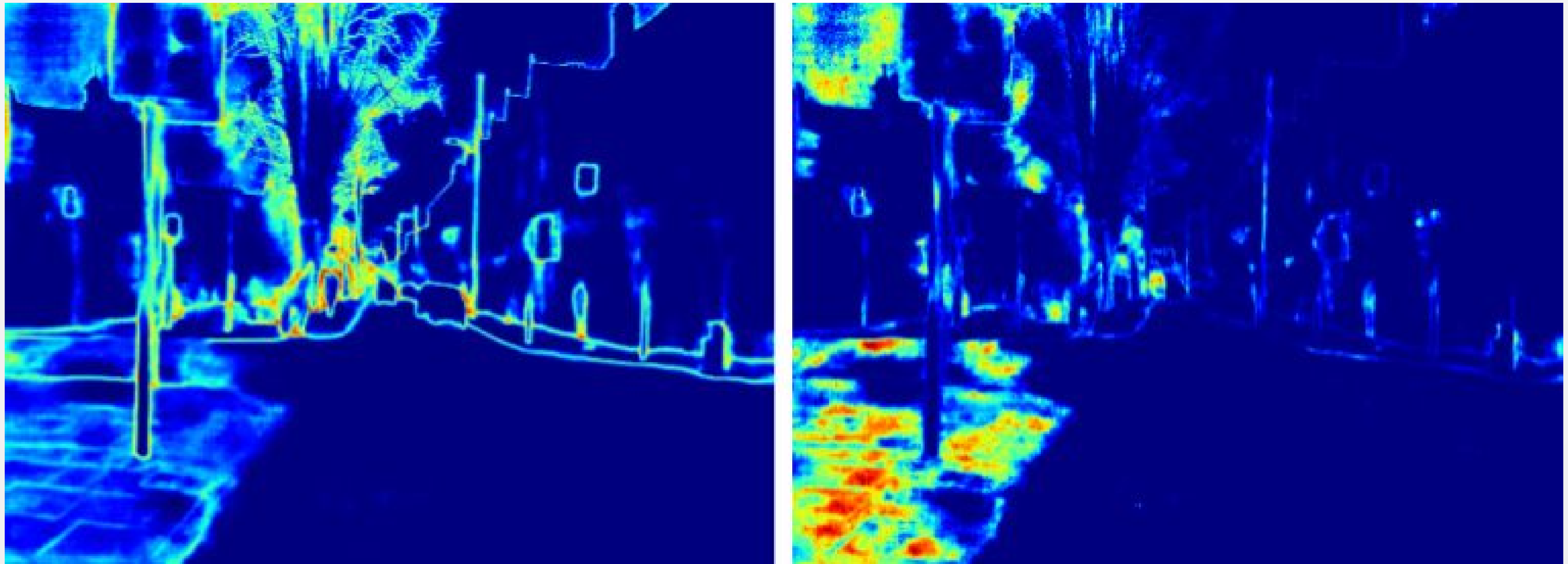


Epistemic

Semantic Segmentation

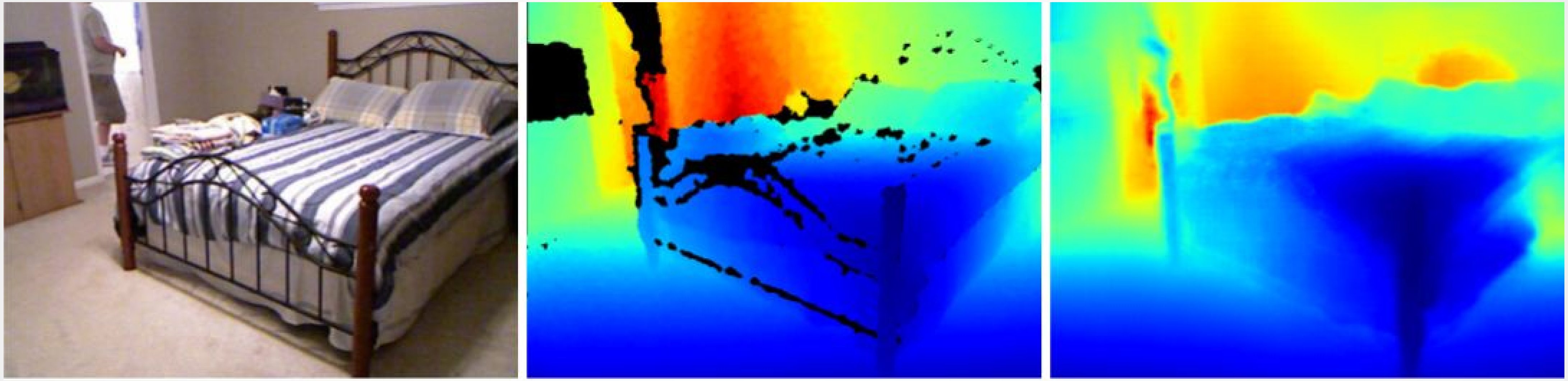


Aleatoric

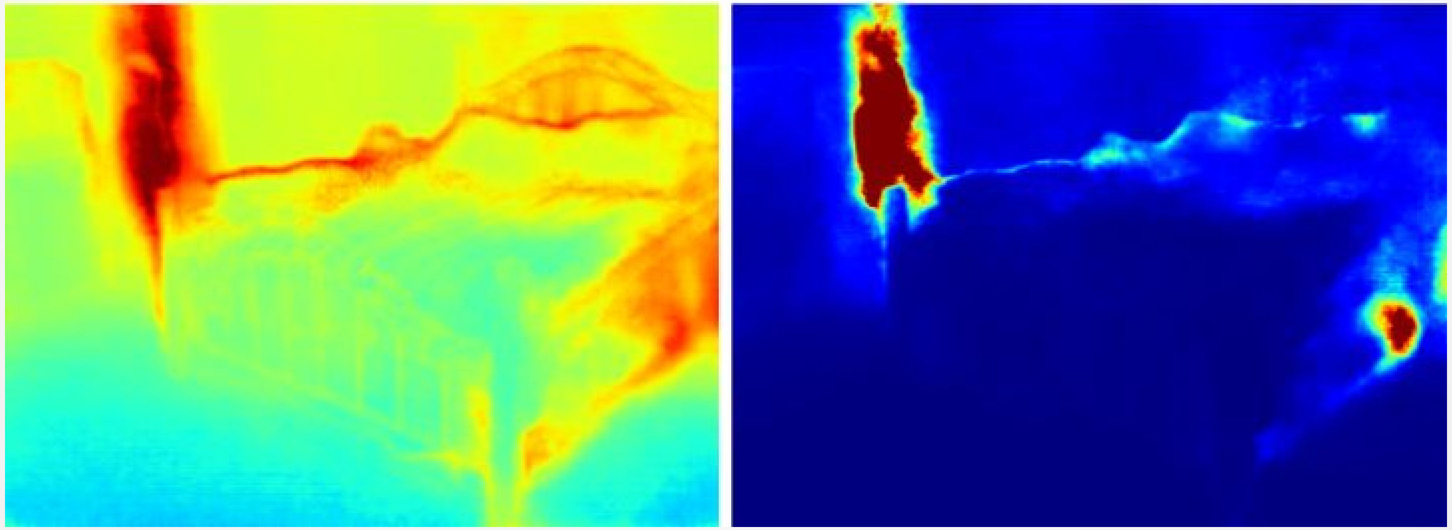


Epistemic

Pixel-wise Depth Regression

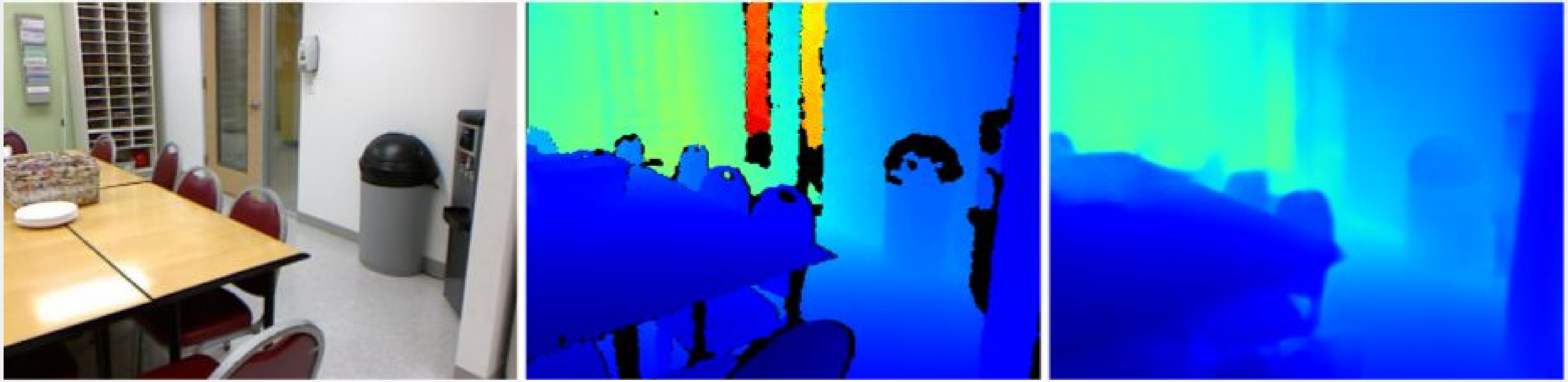


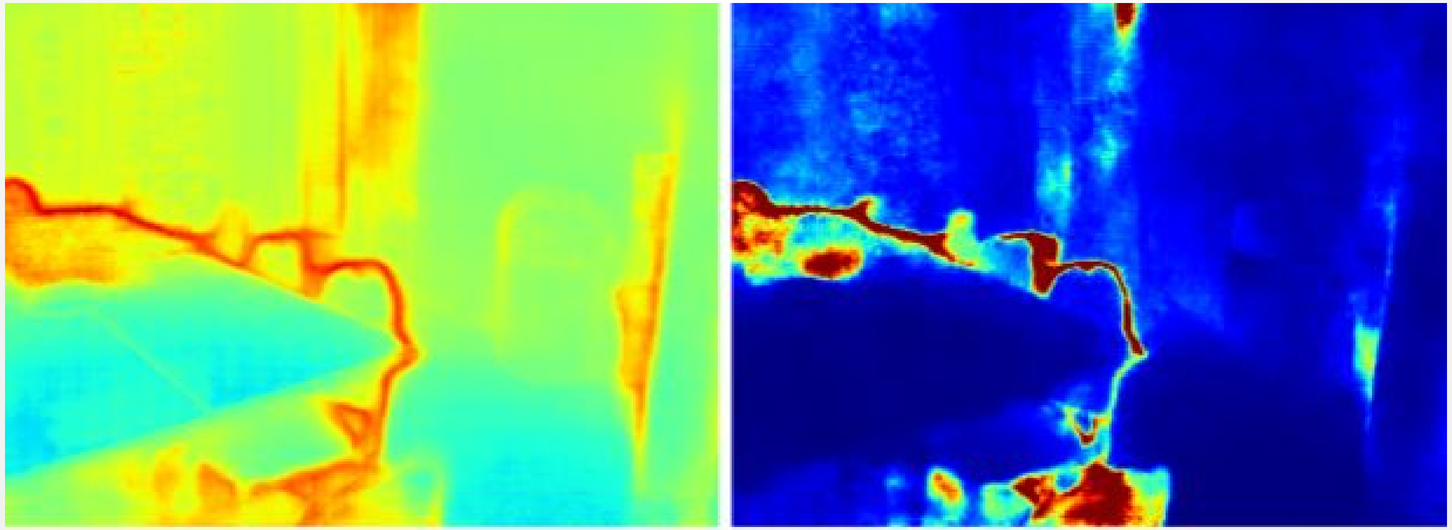
Aleatoric



Epistemic

Pixel-wise Depth Regression





Aleatoric

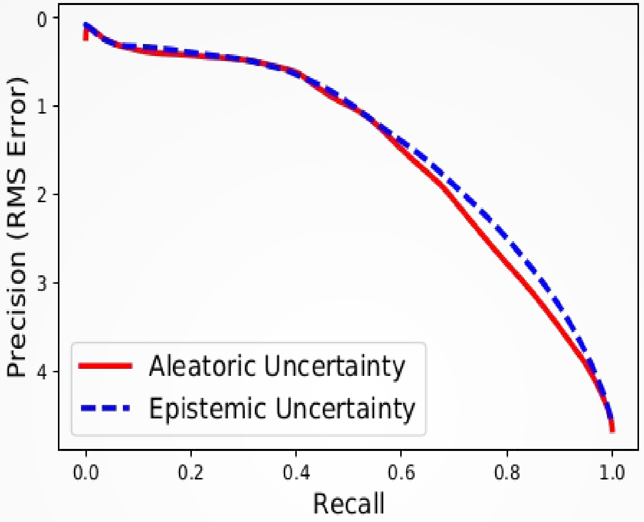
Epistemic

Quality of Uncertainty Metric

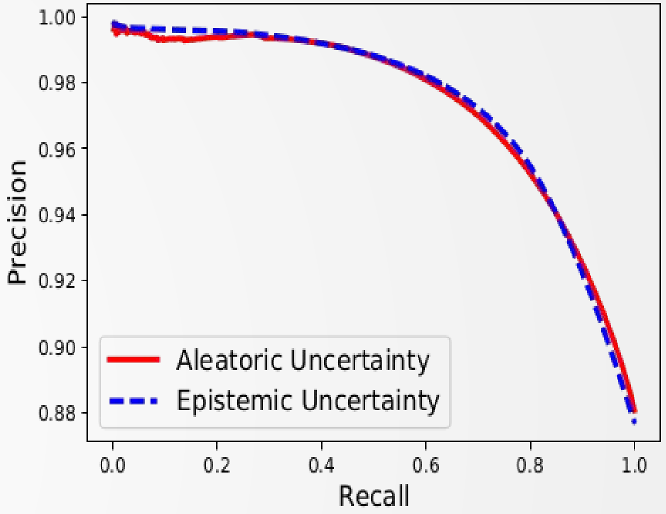
- **Uncertainty correlates with accuracy**

Quality of Uncertainty Metric

- Uncertainty correlates with accuracy
- **Precision decreases as we increase uncertainty**



(b) Regression (Make3D)



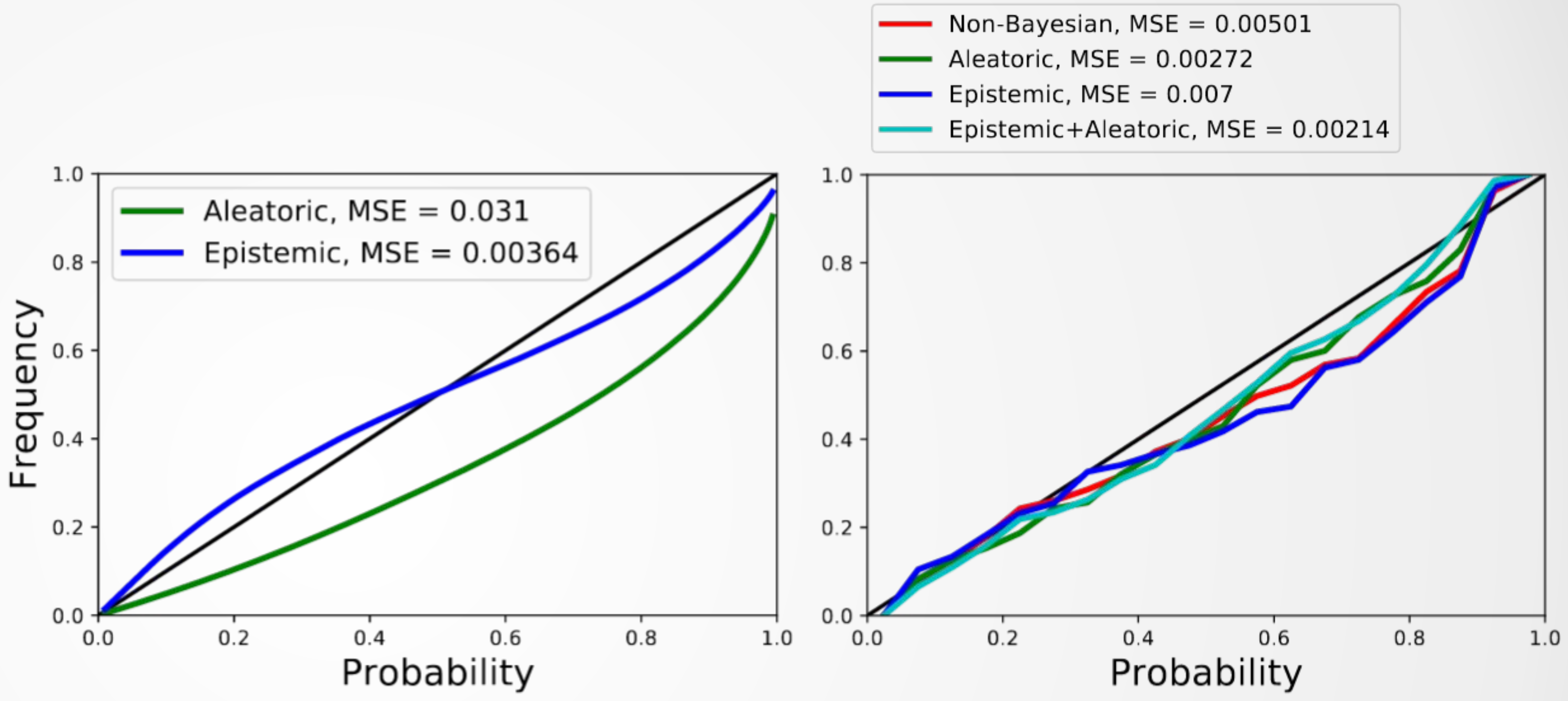
(a) Classification (CamVid)

Calibration

- If prediction has a probability of “p”, we would like this prediction to be correct with frequency “p”

Calibration

- **If prediction has a probability of "p", we would like this prediction to be correct with frequency "p"**



(a) Regression (Make3D) (b) Classification (CamVid)

Dataset size

- **Modeling epistemic variance should be more beneficial when our training data is small**

Dataset size

- Modeling epistemic variance should be more beneficial when our training data is small
- **Epistemic variance captures data from different distribution**

Train dataset	Test dataset	RMS	Aleatoric variance	Epistemic variance
Make3D / 4	Make3D	5.76	0.506	7.73
Make3D / 2	Make3D	4.62	0.521	4.38
Make3D	Make3D	3.87	0.485	2.78
Make3D / 4	NYUv2	-	0.388	15.0
Make3D	NYUv2	-	0.461	4.87

Train dataset	Test dataset	IoU	Aleatoric entropy	Epistemic logit variance ($\times 10^{-3}$)
CamVid / 4	CamVid	57.2	0.106	1.96
CamVid / 2	CamVid	62.9	0.156	1.66
CamVid	CamVid	67.5	0.111	1.36
CamVid / 4	NYUv2	-	0.247	10.9
CamVid	NYUv2	-	0.264	11.8

Improvement over Baseline

CamVid Results	IoU Accuracy
DenseNet (State of the art baseline)	67.1
+ Aleatoric Uncertainty	67.4
+ Epistemic Uncertainty	67.2
+ Aleatoric & Epistemic	**67.5**

NYU Depth Results	Rel. Error
DenseNet (State of the art baseline)	0.167
+ Aleatoric Uncertainty	0.149
+ Epistemic Uncertainty	0.162
+ Aleatoric & Epistemic	**0.145**

Source: *http://alexgkendall.com/talks/*

Outline:

- Motivation
- Types of Uncertainty
- Bayesian Neural Networks
- Dropout Variational Inference
- Modeling uncertainties
- Experiments
- Results Analysis
- **Summary**

Summary:

- Aleatoric uncertainty:
  - Can be used for real time applications
  - Can be used alone for large datasets
- Epistemic uncertainty:
  - Can detect samples out of the training data
  - Useful for small datasets
  - Expensive to evaluate (MC Sampling)

Thank you