Evidential Deep Learning

Alexander Amini

MIT 6.S191

January 26, 2021

Motivation: uncertainty in learning

Safety critical domains require ***fast, scalable, and calibrated*** *uncertainty estimation*

Evidential deep learning ➡️ *predict answer and amount of evidence (confidence)*

Neural networks: expectation vs reality

Expectation:
Training on a your dataset

Reality:
Testing in reality

Dogs

Driving

Neural networks: expectation vs reality

Expectation:
Training on a your dataset

Reality:
Testing in reality

Dogs

"All models are wrong, but some — *that know when they can be trusted* — are useful!"

\- **George E.P. Box (Adapted)**

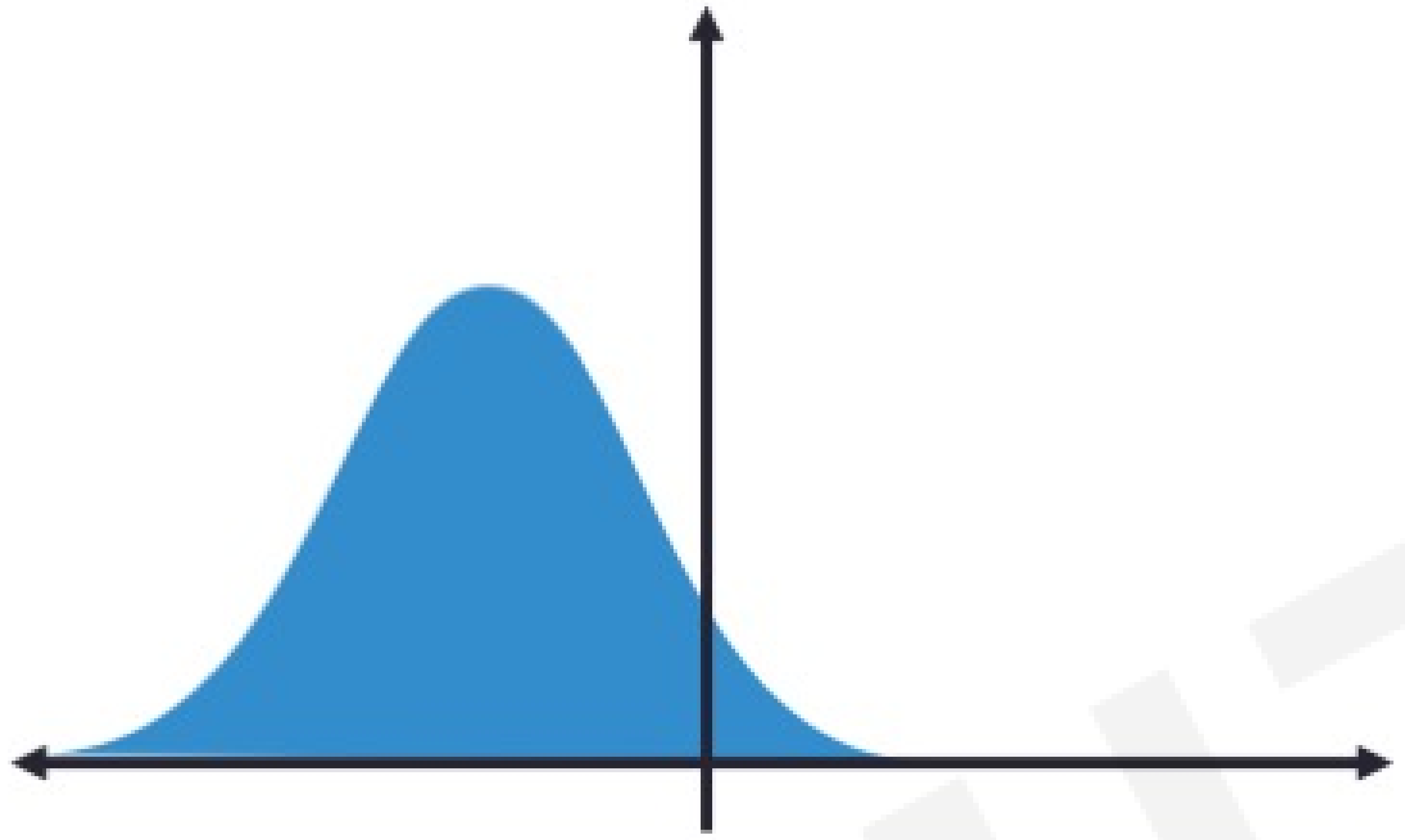
Driving

Knowing when we don't know is hard

…even for humans!

Topics for today

Learning probability distributions



Model distributions over labels: Softmax (discrete) & Gaussian (continuous)

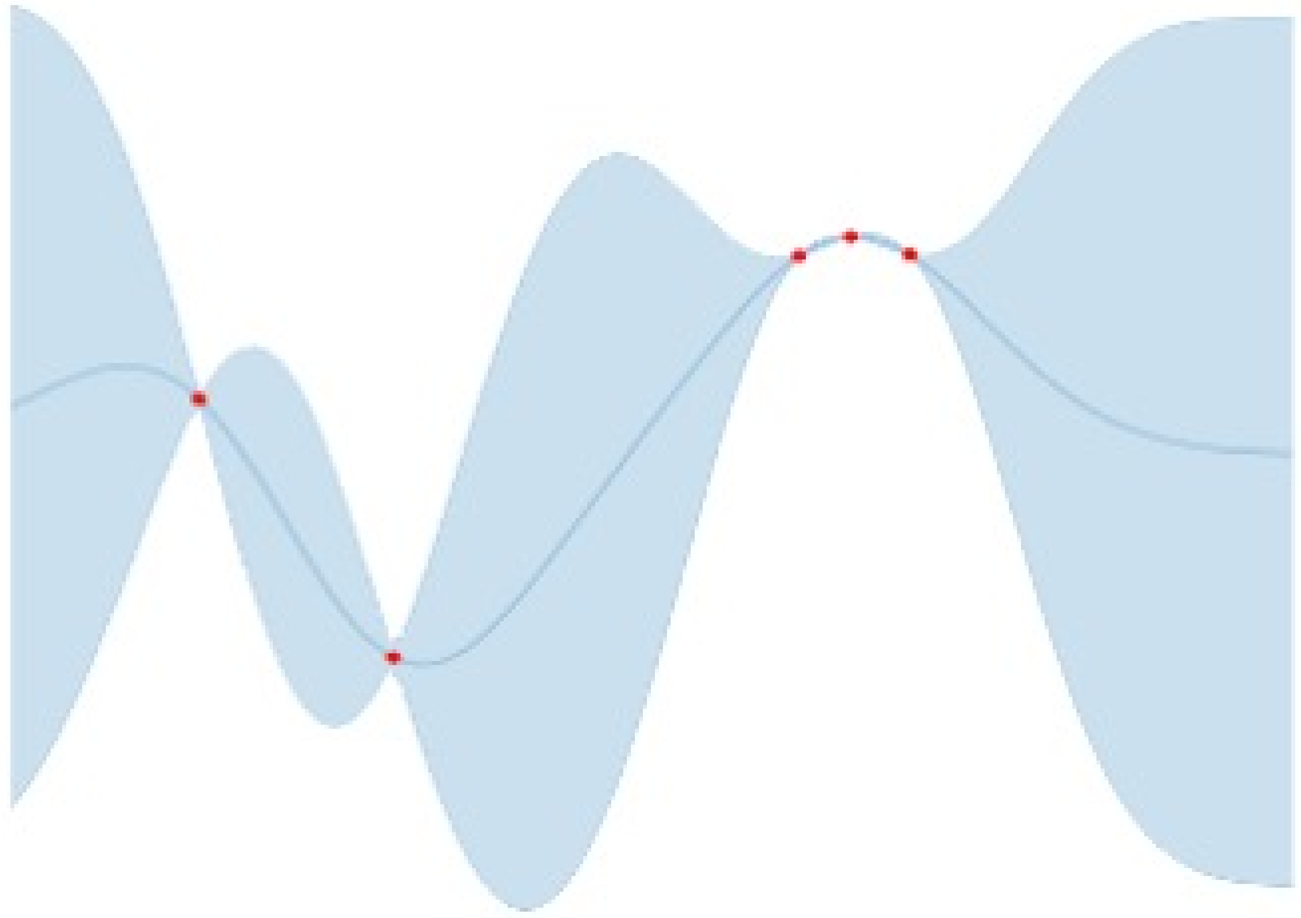
Different sources of uncertainty

$$p(y|x)$$

$$\mathbb{E}[\sigma^2] \quad \mathrm{Var}[\mu]$$

Data (aleatoric) uncertainty vs. Model (epistemic) uncertainty.

Fast and scalable uncertainty estimation



Evidential deep learning

Uncertainty modelling for quickly estimating confidence

Probabilistic learning

Data

x

(images, audio, signal, …)

Target

y

(label, class, …)

Prediction

$\mathbb{E}[y]$

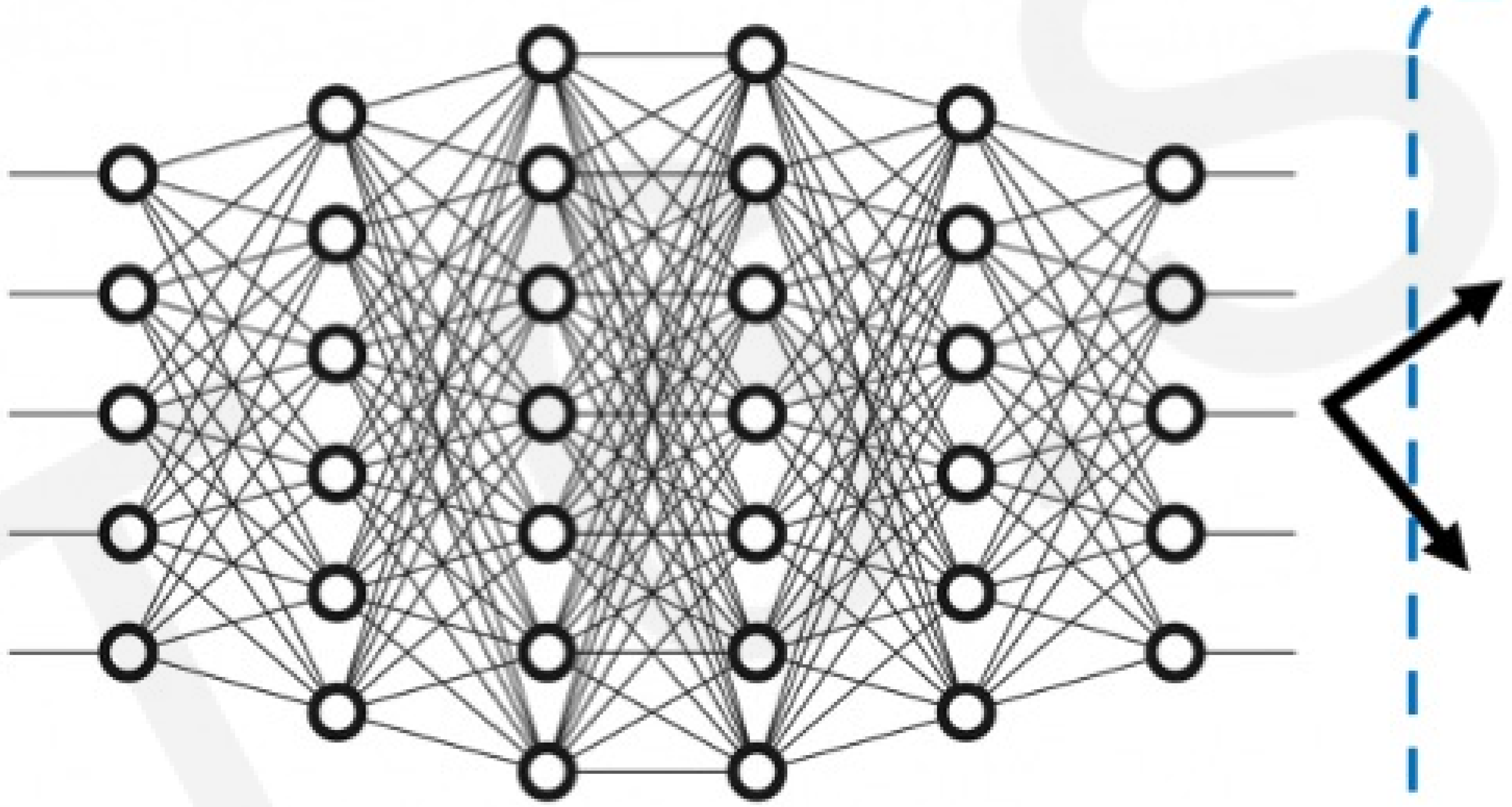
Uncertainty

$\mathrm{Var}[y]$

Learning probabilistic outputs

🤔 Wait, haven't we already learned this?!

x



$$p(y = \text{“}cat\text{”} \mid x)$$

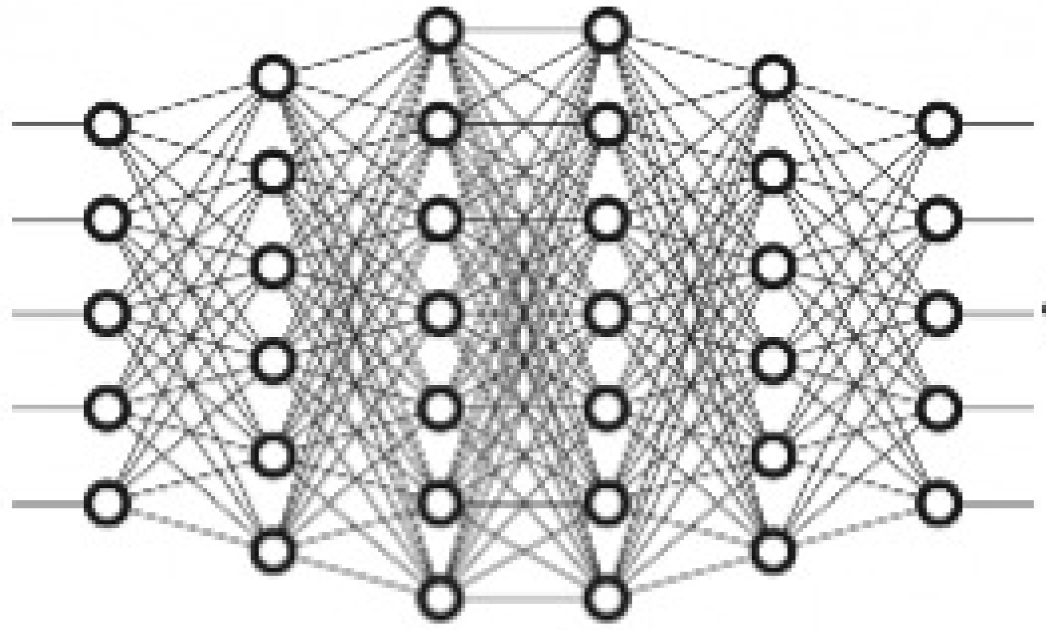
$$p(y = \text{“}dog\text{”} \mid x)$$

Probability distribution over
discrete class categories

Learning discrete class targets

Classification

x



$p(y = \text{“}cat\text{”} \mid x)$

$p(y = \text{“}dog\text{”} \mid x)$

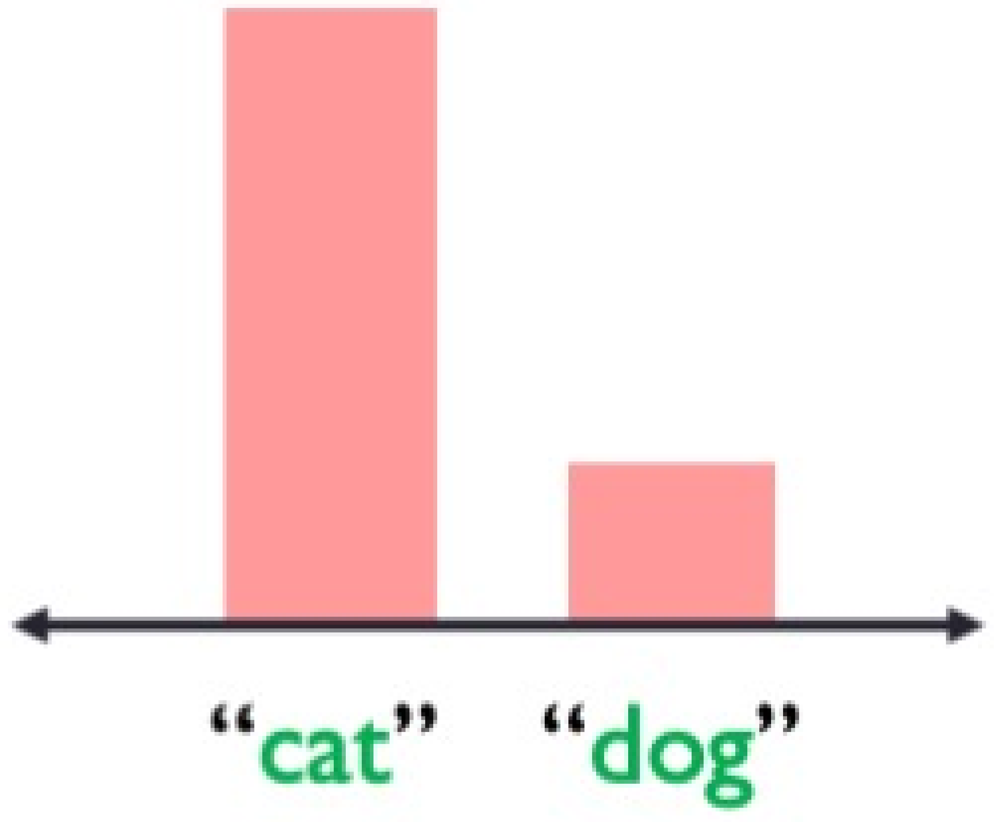
Activation: softmax(z) → $\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$

Loss: Neg. Log Likelihood (Cross Entropy) → $-\sum_{i=1}^{K} y_i \log p_i$

Why?

$$\underbrace{\boldsymbol{y}}_{\text{Class Labels}} \sim \underbrace{\text{Categorical}}_{\text{Likelihood function}}(\underbrace{\boldsymbol{p}}_{\text{Distribution parameters (probabilities)}})$$

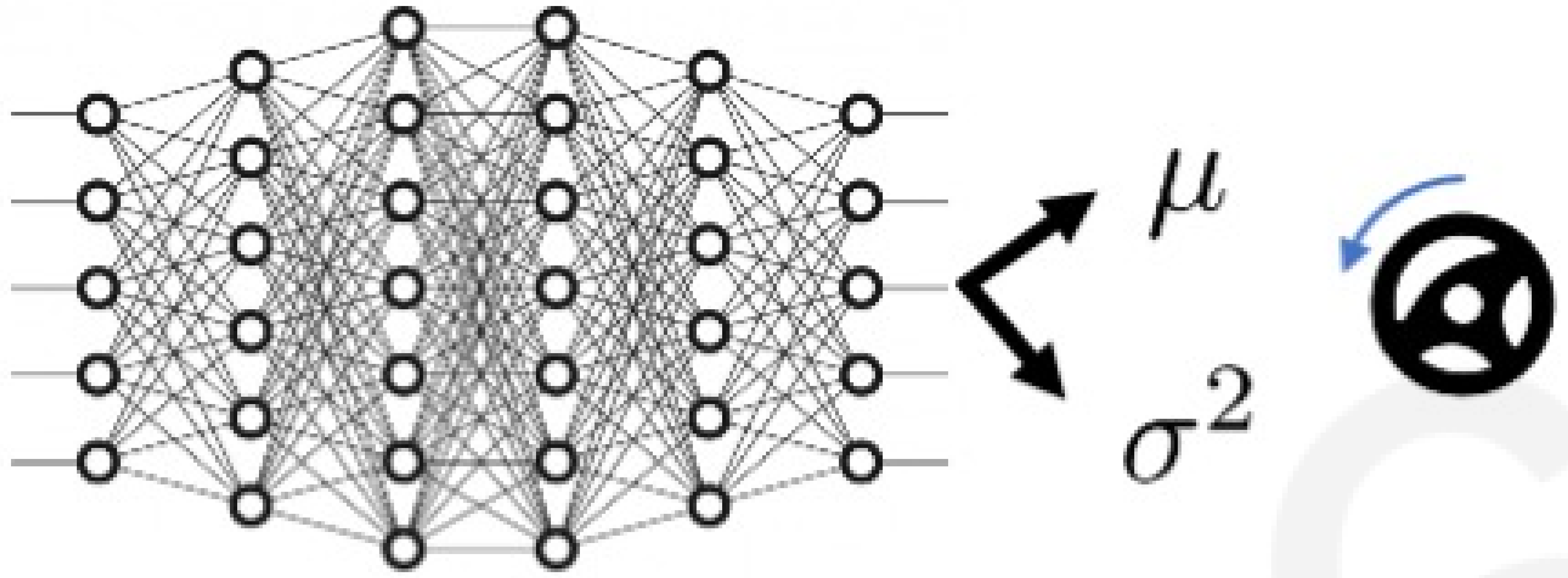
$$f(y = y_i \mid \boldsymbol{p}) = p_i$$



Learning continuous class targets

Regression


x


μ

σ^2

Activation: $\mu \in \mathbb{R}$, $\sigma > 0$ → $\mu = z_\mu$, $\sigma = \exp(z_\sigma)$

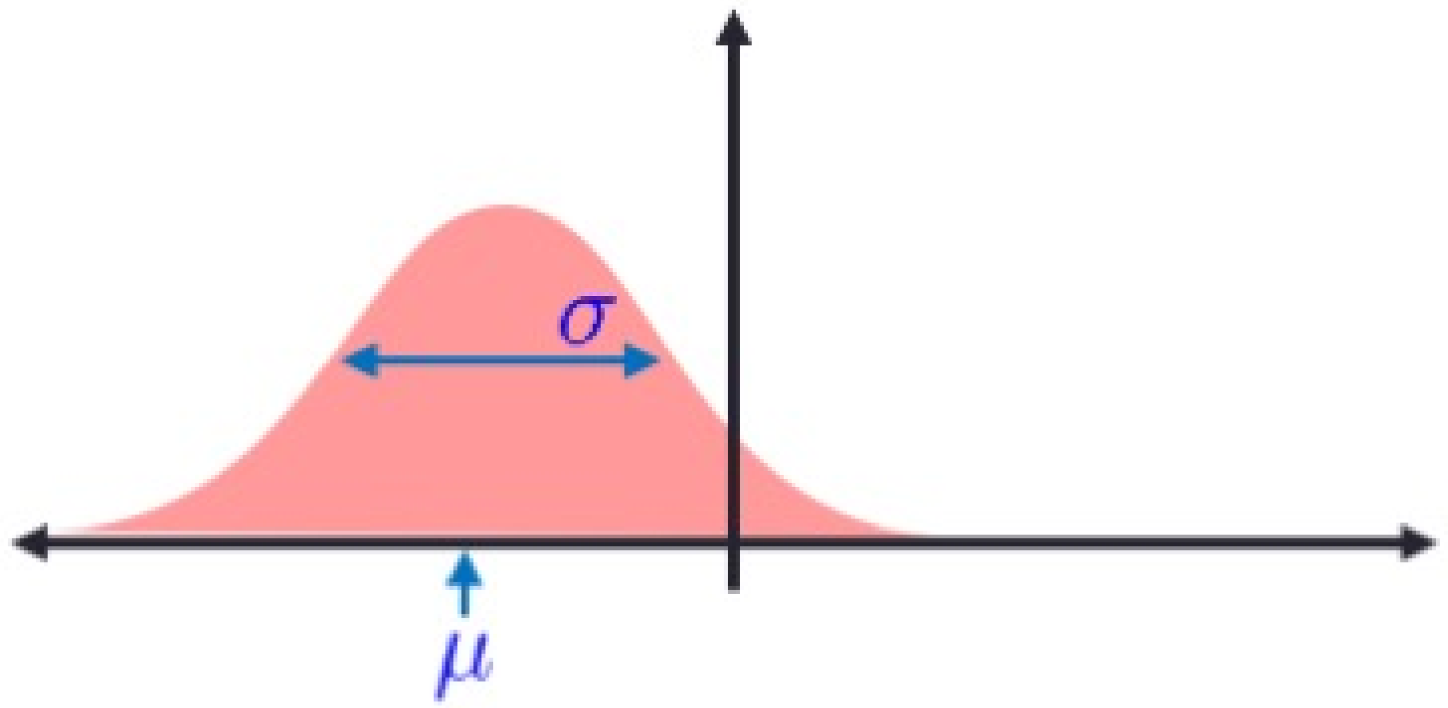
Loss: Neg. Log Likelihood → $-\log\left(\mathcal{N}(y|\mu,\sigma^2)\right)$

Why?

$$y \sim \text{Normal}(\mu, \sigma^2)$$

Target Labels · Likelihood function · Distribution parameters

$$f(y \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right)$$



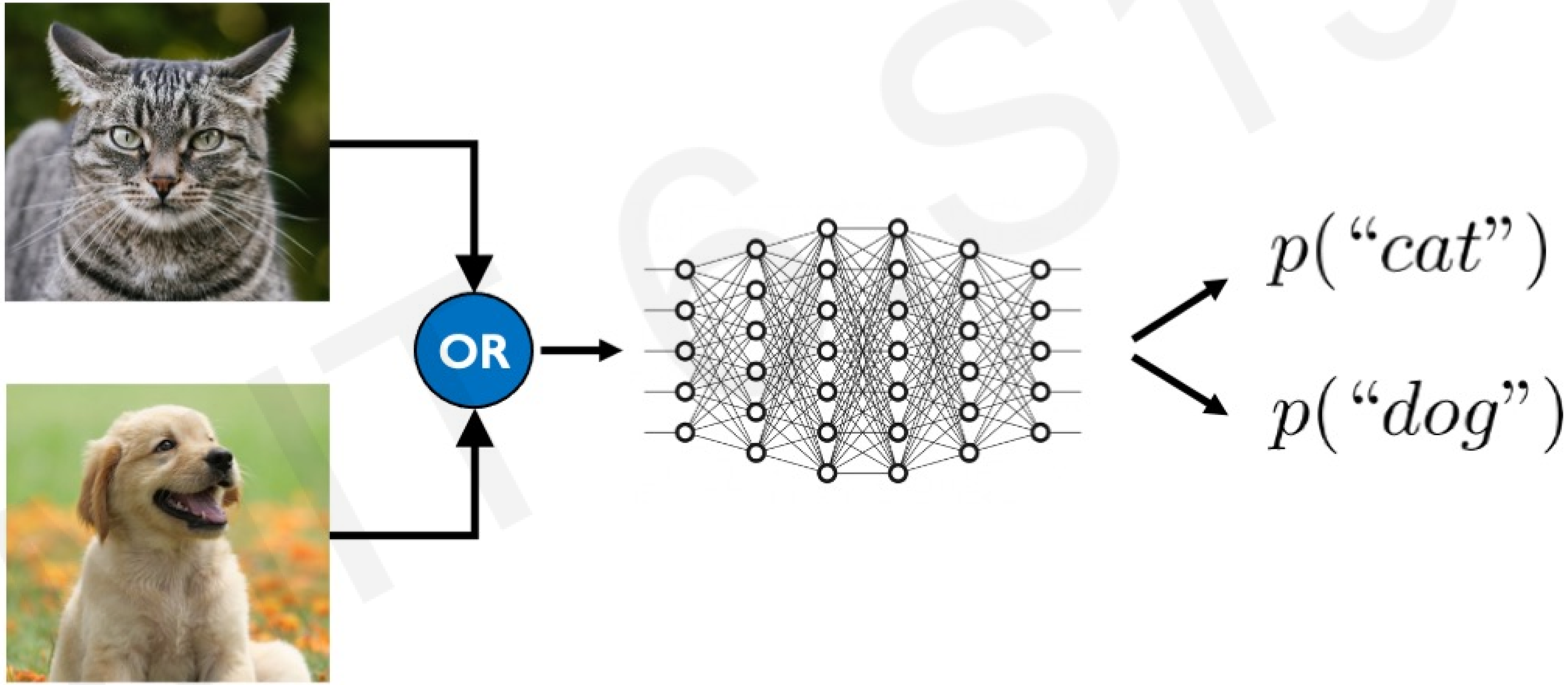
Likelihood estimation in deep learning

	Classification (discrete)	Regression (continuous)
Targets	$y \in \{1, \cdots, K\}$	$y \in \mathbb{R}$
Likelihood	$y \sim \text{Categorical}(\boldsymbol{p})$ `tfp.distributions.Categorical(probs=p)`	$y \sim \text{Normal}(\mu, \sigma^2)$ `tfp.distributions.Normal(mu, sigma)`
Parameters	$\boldsymbol{p} = \{p_1, \cdots, p_K\}$	(μ, σ^2)
Constraints	$\sum_i p_i = 1; \quad p_i > 0$	$\mu \in \mathbb{R}; \quad \sigma > 0$
Loss function	Cross Entropy $-\sum_{i=1}^{K} y_i \log p_i$ `dist.cross_entropy(y)`	Negative Log-Likelihood $-\log\left(\mathcal{N}(y \mid \mu, \sigma^2)\right)$ `-1 * dist.log_prob(y)`

Likelihood vs Confidence

⚠️ WARNING: ⚠️

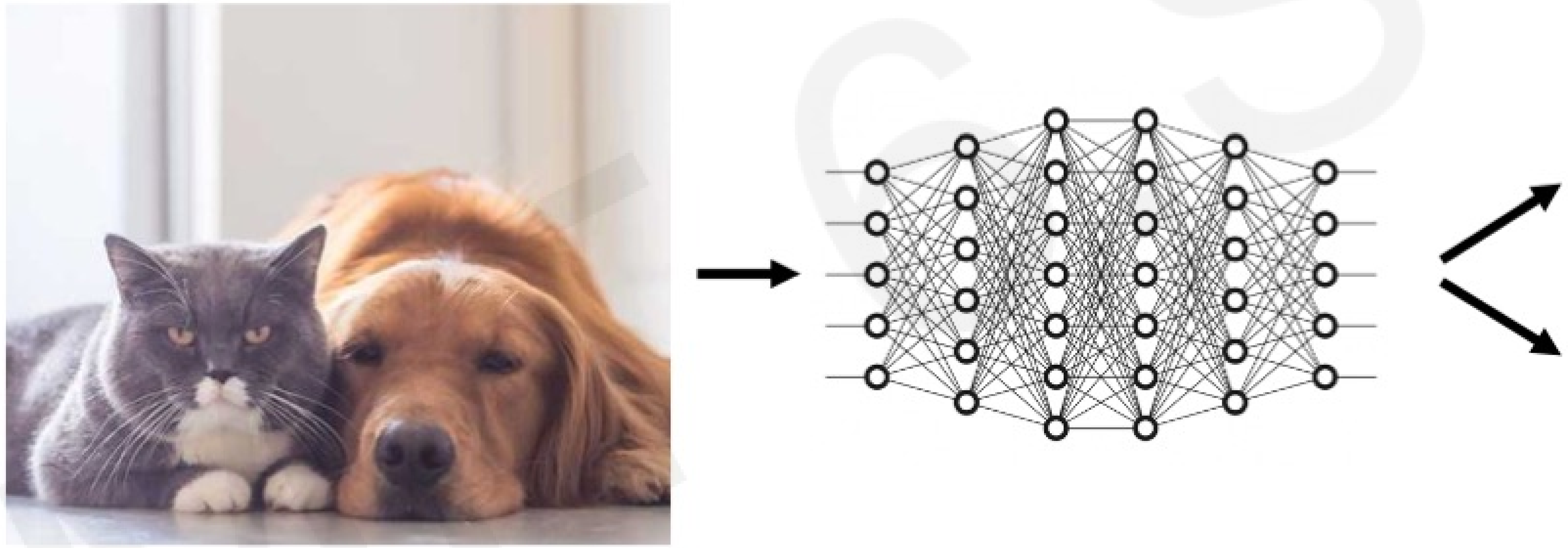
Do not mistake likelihood (probability) for model confidence



Likelihood vs Confidence

⚠️ WARNING: ⚠️

Do not mistake likelihood (probability) for model confidence



$$p(\text{“}cat\text{”}) = 0.5$$

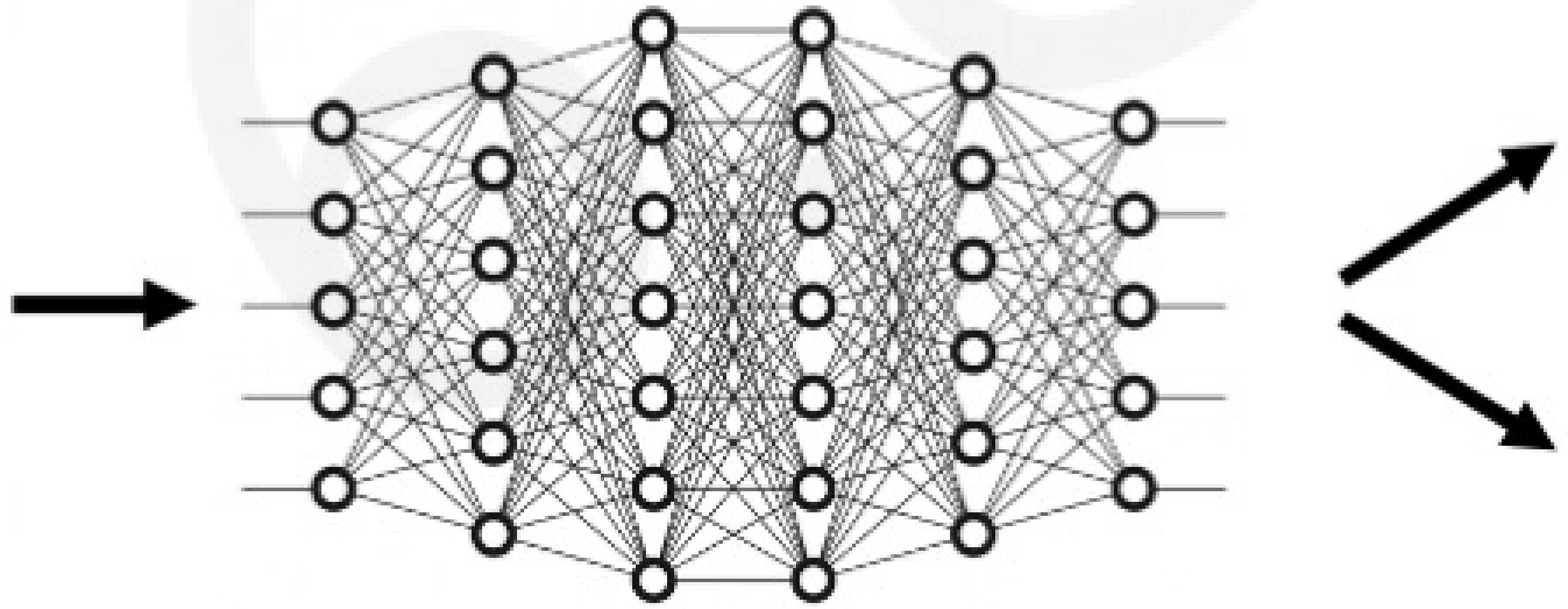
$$p(\text{“}dog\text{”}) = 0.5$$

Likelihood vs Confidence

⚠️ WARNING: ⚠️

Do not mistake likelihood (probability) for model confidence

The output likelihoods will be unreliable if the input is ***unlike anything during training***



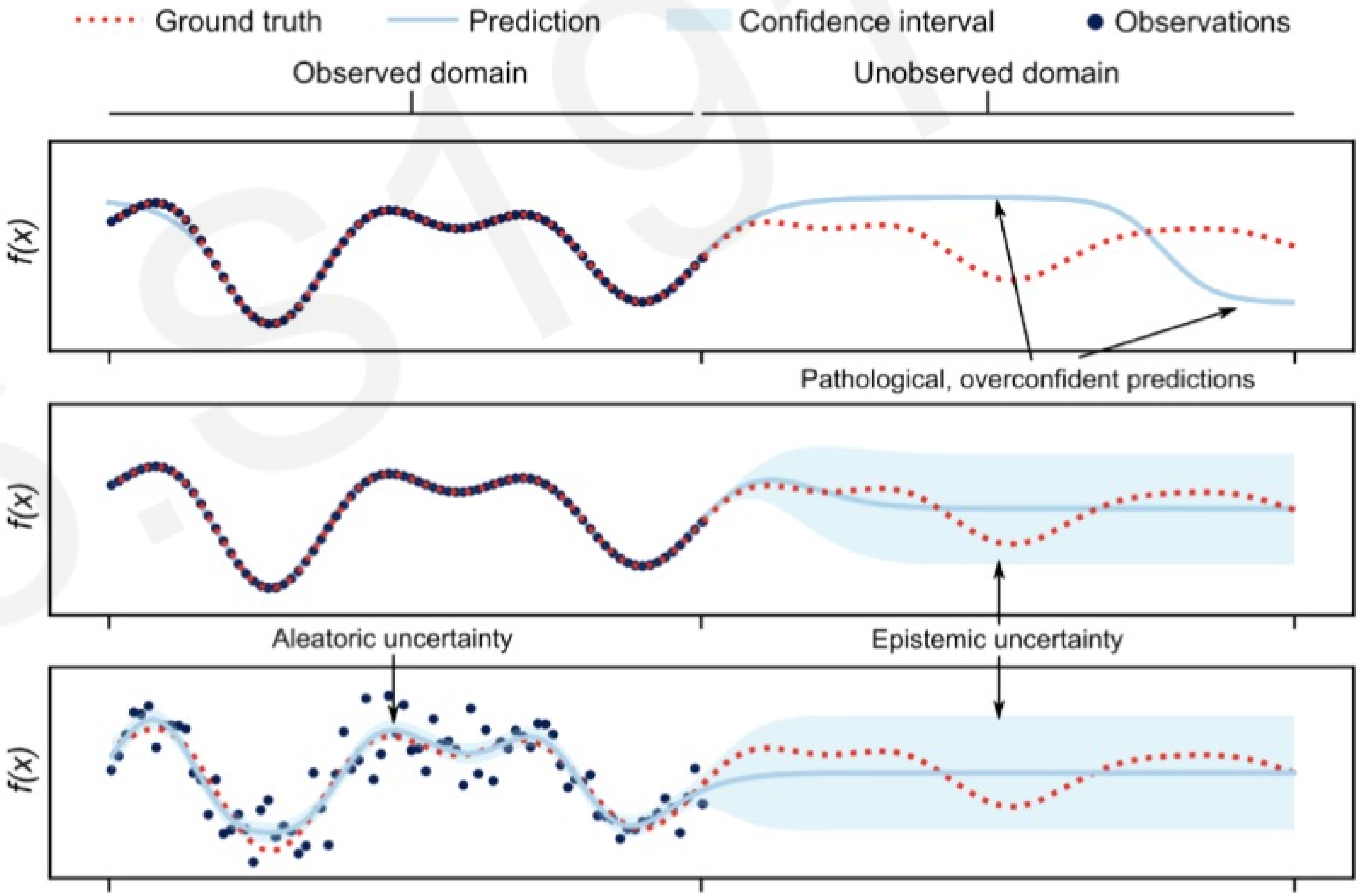
$p(\text{"cat"})$

$p(\text{"dog"})$

⭐ $p(\text{"cat"}) + p(\text{"dog"}) = 1$ ⭐

Types of uncertainty

Known Knowns	Known Unknowns
Things we are certain of	We know there are things we can't predict
Unknown Known	**Unknown Unknowns**
Others know but you don't know	Completely unexpected or unforeseeable events



Aleatoric vs Epistemic Uncertainty

Aleatoric Uncertainty

Data Uncertainty

Describes the confidence in the input data

High when input data is noisy

Cannot be reduced by adding more data

Epistemic Uncertainty

Model Uncertainty

Describes the confidence of the prediction

High when missing training data

Can be reduced by adding more data

Aleatoric vs Epistemic Uncertainty

Aleatoric Uncertainty

Data Uncertainty

Describes the confidence in the input data

High when input data is noisy

Cannot be reduced by adding more data

Epistemic Uncertainty

Model Uncertainty

Describes the confidence of the prediction

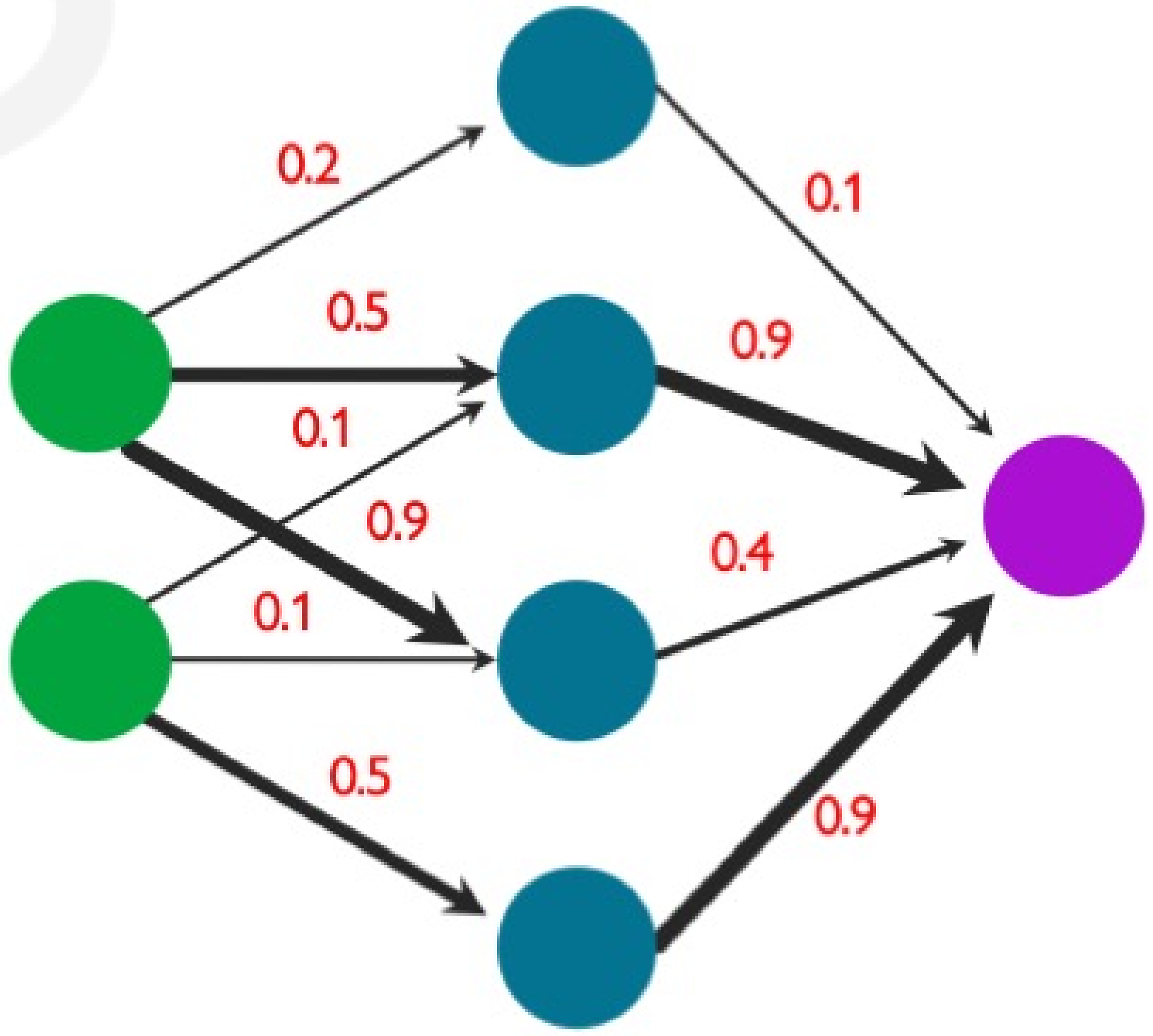
High when missing training data

Can be reduced by adding more data

Estimating epistemic uncertainty

- ✓ Aleatoric uncertainty can be learned directly using neural networks
- 💪 Epistemic uncertainty is **much more challenging** to estimate
- 🤔 How can a model understand when it does not know the answer?

One solution:
Don't train **deterministic NN**, but instead train a Bayesian NN!



Estimating epistemic uncertainty

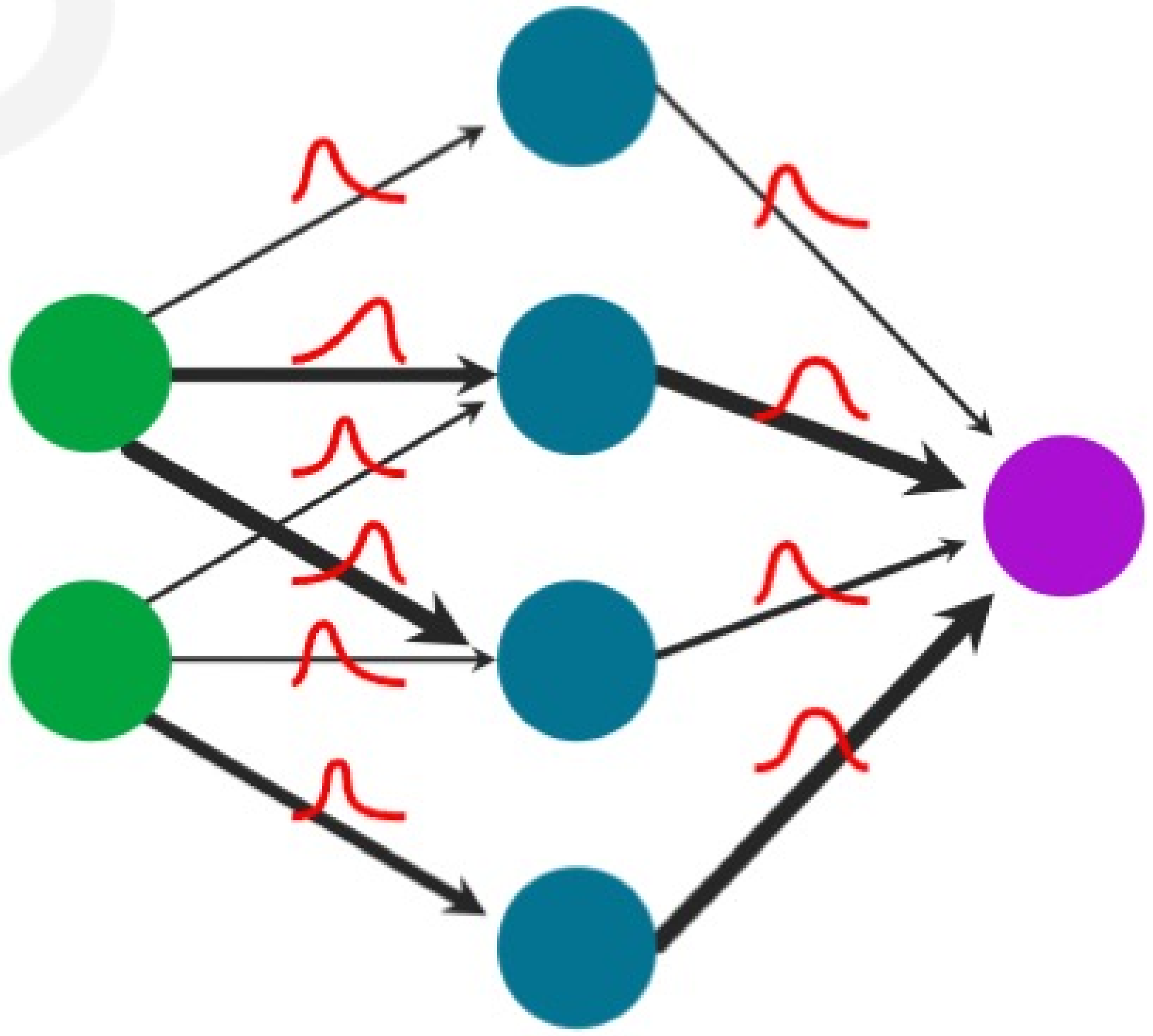
✓ Aleatoric uncertainty can be learned directly using neural networks

💪 Epistemic uncertainty is **much more challenging** to estimate

🤔 How can a model understand when it does not know the answer?

One solution:

Don't train deterministic NN, but instead train a **Bayesian NN**!



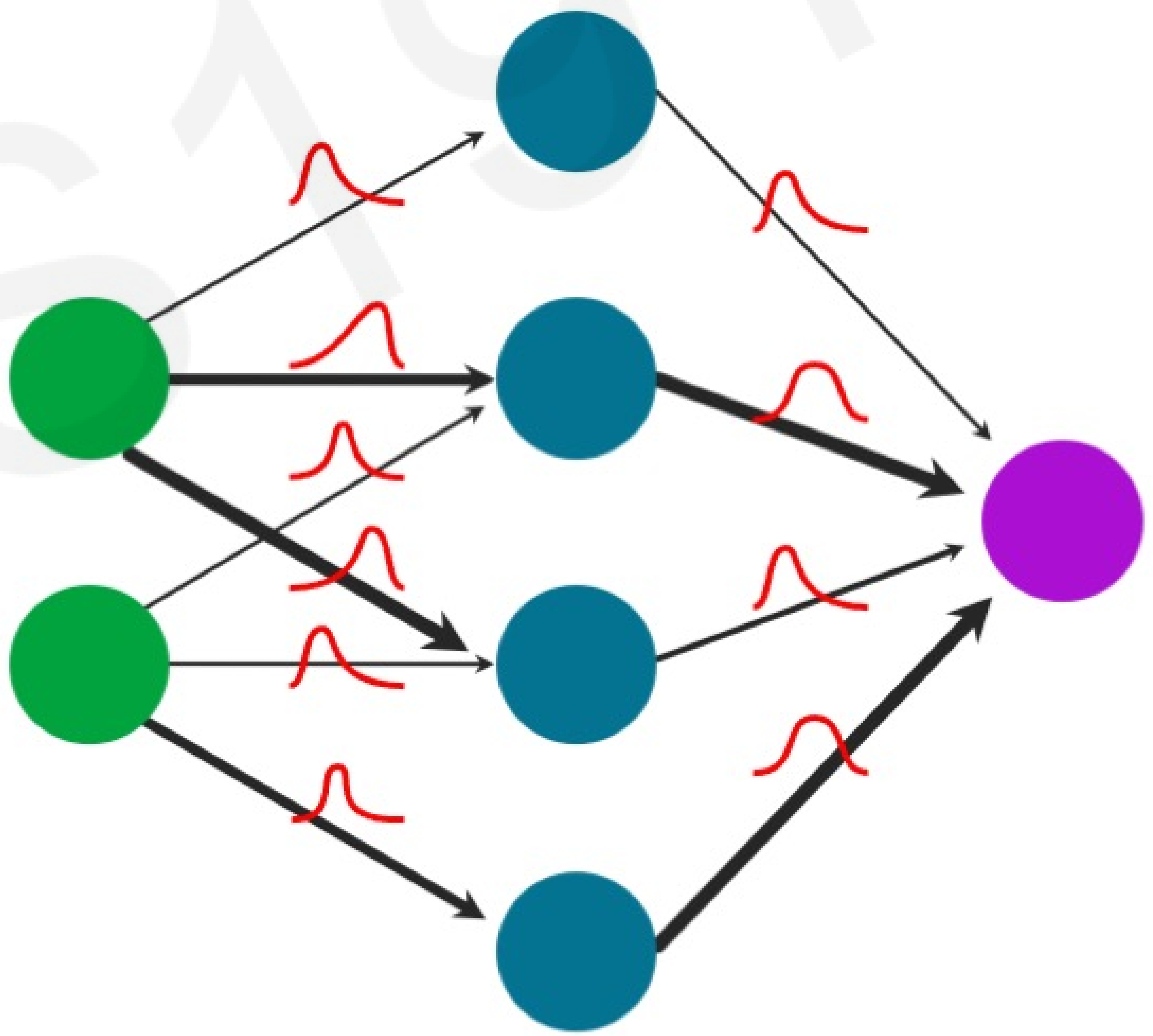
Bayesian deep learning for uncertainty

Deterministic neural networks (NNs) learn a fixed set of weights, $\boldsymbol{W}$

Bayesian neural networks aim to learn a posterior over weights, $P(\boldsymbol{W}|\boldsymbol{X},\boldsymbol{Y})$

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

Intractable!

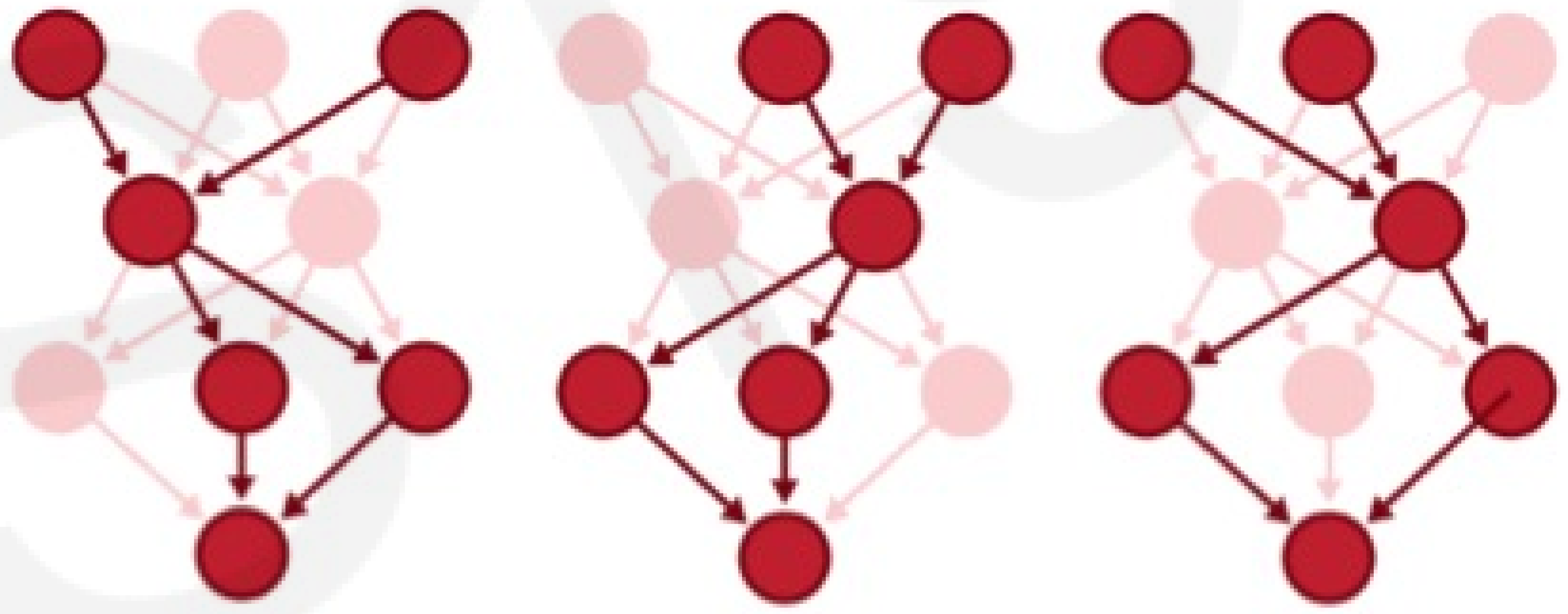


Approximations through sampling

Evaluate T stochastic forward passes using different samples of weights $\{\boldsymbol{W}_t\}_{t=1}^{T}$

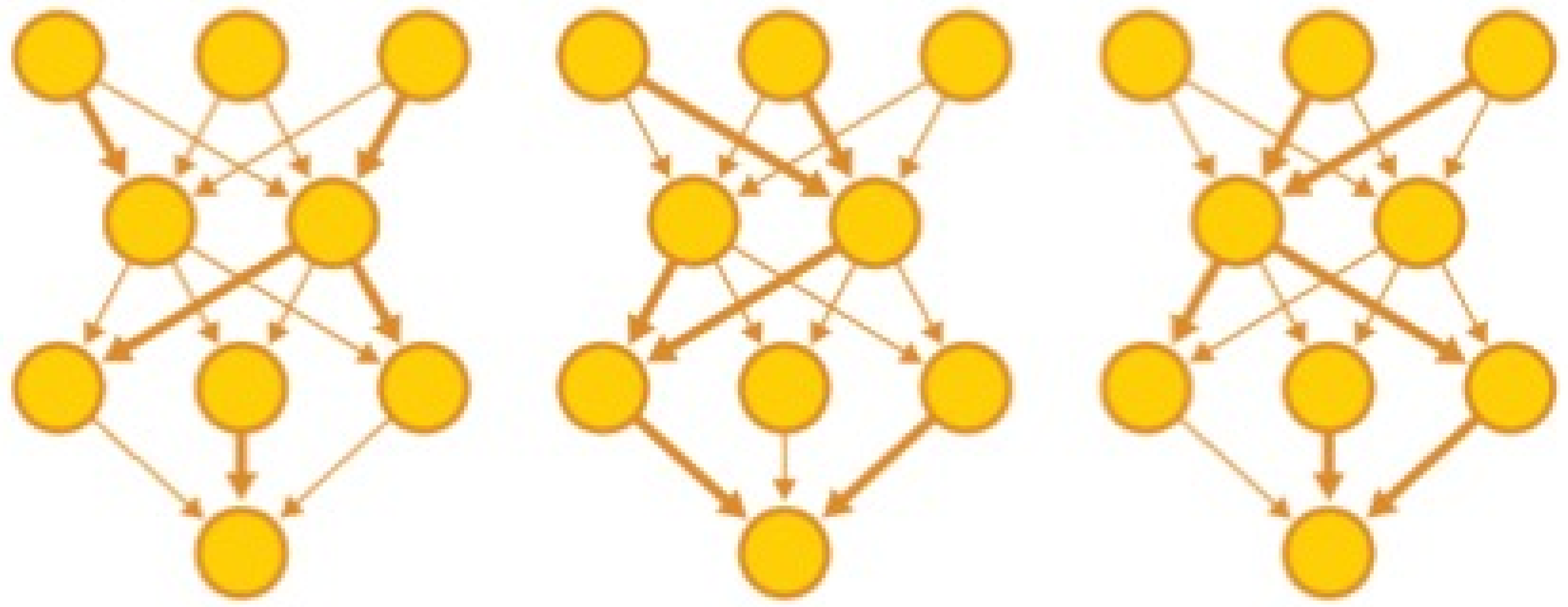
Dropout as a form of stochastic sampling

$$z_{w,t} \sim Bernoulli(p) \quad \forall\, w \in \boldsymbol{W}$$



Ensemble of T independently trained models, each learning a unique $\boldsymbol{W}_t$

$$\boldsymbol{W}_t = train(f; \boldsymbol{X}, \boldsymbol{Y})$$



$$\mathbb{E}(\widehat{\boldsymbol{Y}}|\boldsymbol{X}) = \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{X}|\boldsymbol{W}_t) \qquad Var(\widehat{\boldsymbol{Y}}|\boldsymbol{X}) = \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{X})^2 - \mathbb{E}(\widehat{\boldsymbol{Y}}|\boldsymbol{X})^2$$

Gal+ *ICML* 2016. Lakshminarayan+ *NIPS* 2017.

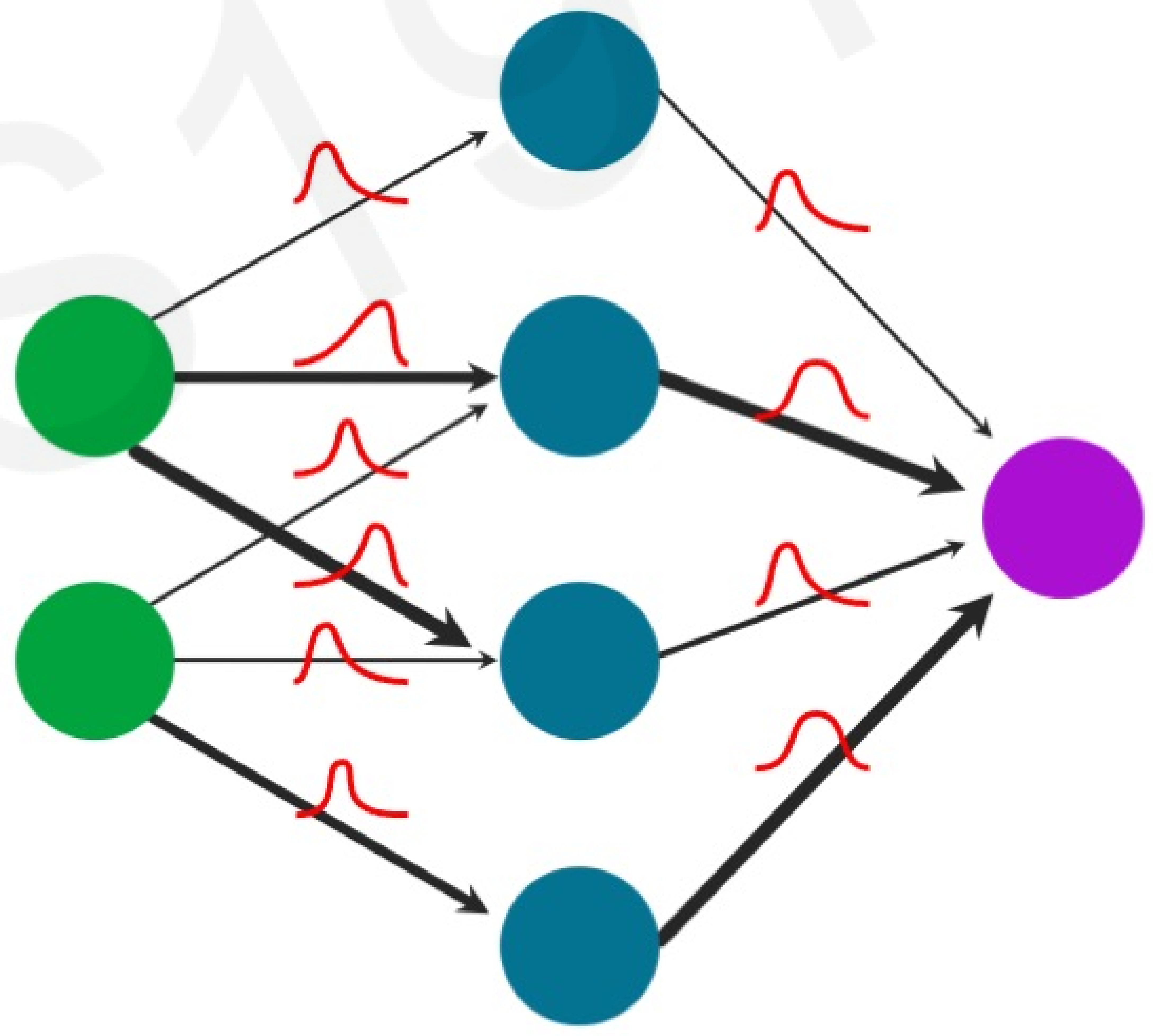
Downsides of Bayesian deep learning

Slow: Requires running the network T times for every input

Memory: Store T copies of the network in parallel

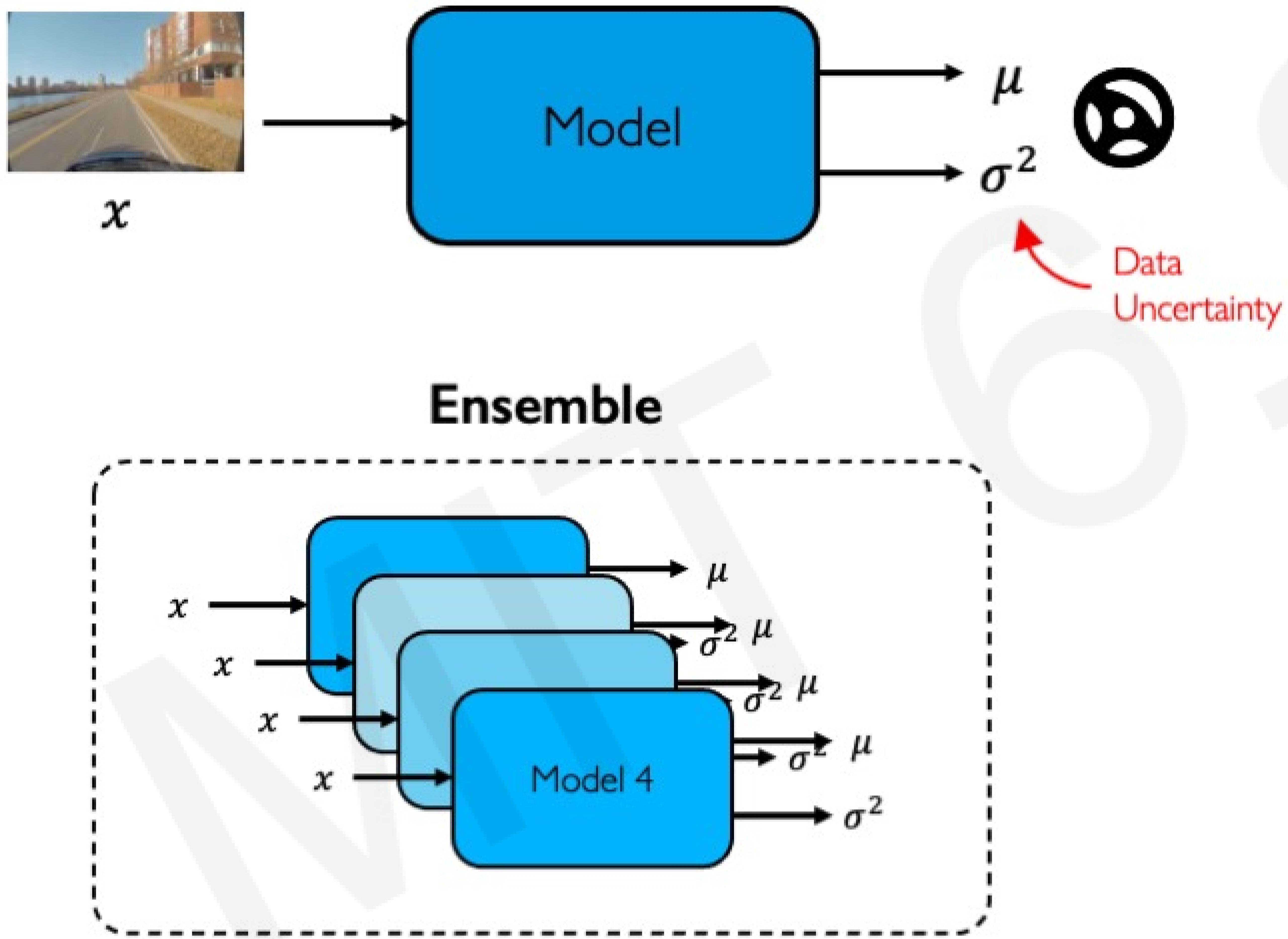
Efficiency: Sampling hinders real-time ability on edge devices (robotics)

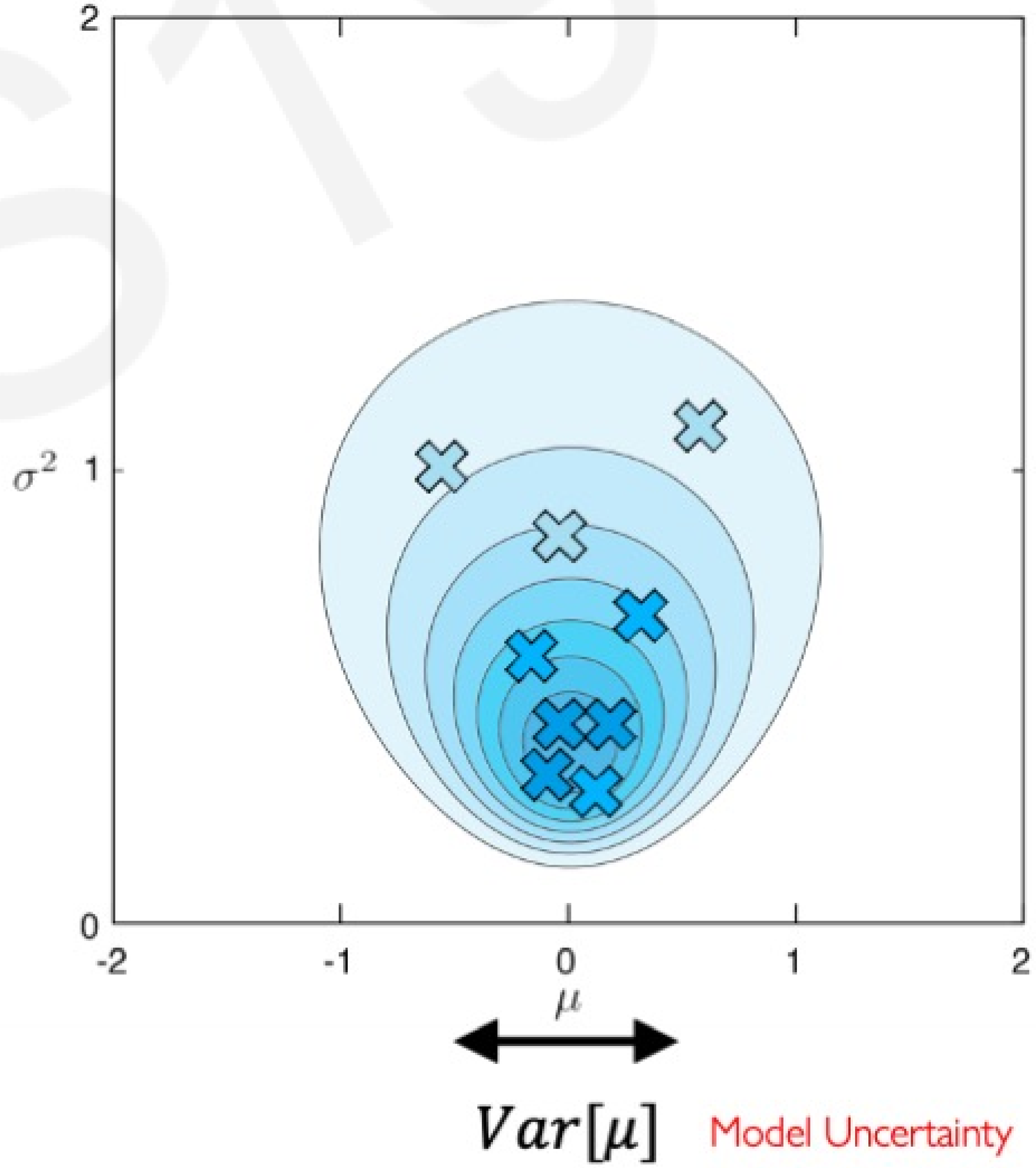
Calibration: Sensitive to choice of prior and is often over-confident



Beyond sampling for approximating uncertainty

Sampling an ensemble of models to approximate the uncertainty





Evidential deep learning

Treat learning as an **evidence acquisition** process

More evidence → increased predictive confidence

Low uncertainty
High confidence

High aleatoric (data) uncertainty

High epistemic (model) uncertainty

Evidential learning for regression

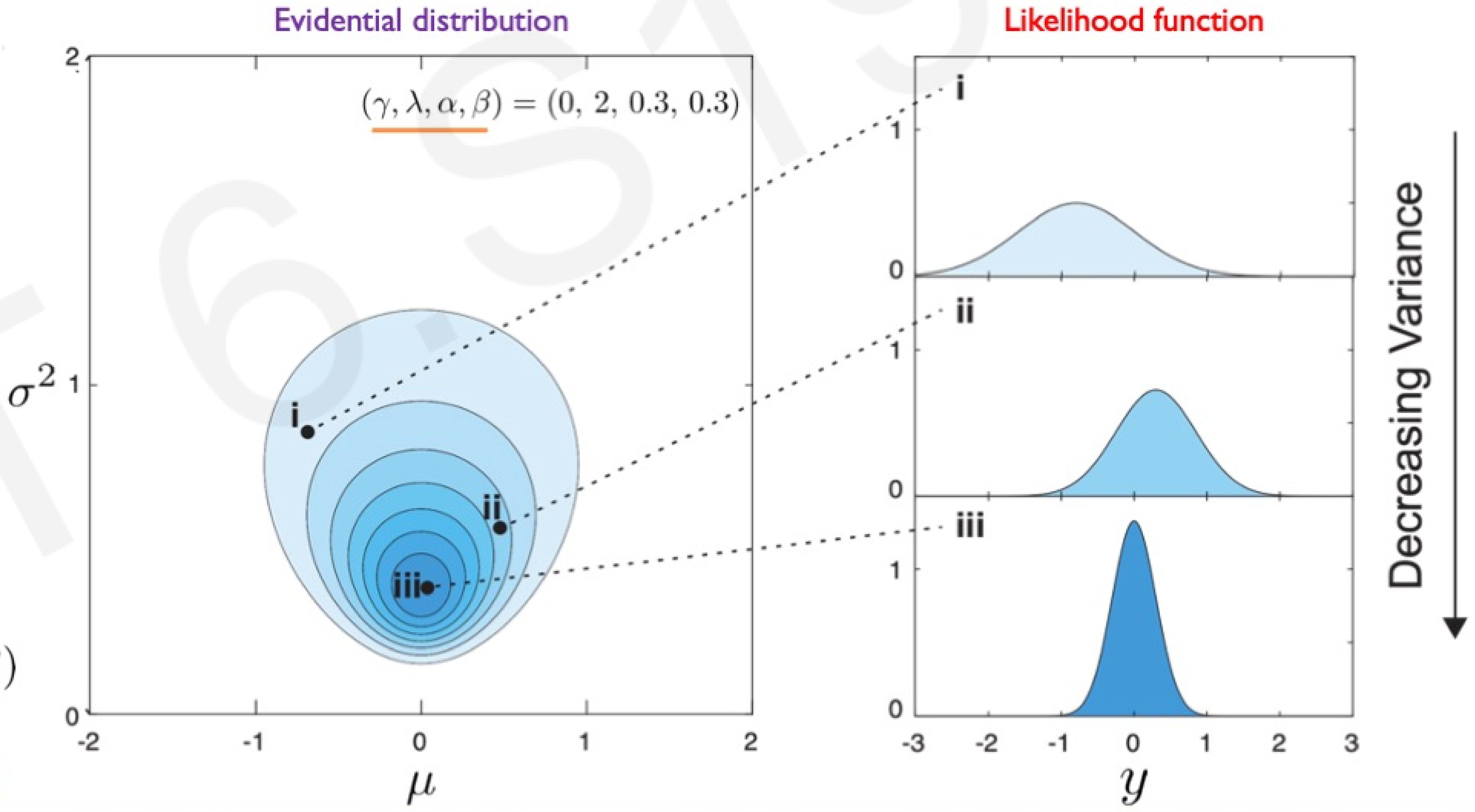
Sampling from an evidential distribution yields individual new distributions over the data

$$\underbrace{y}_{\text{Target Labels}} \sim \underbrace{\text{Normal}}_{\text{Likelihood function}}\underbrace{(\mu, \sigma^2)}_{\text{Distribution parameters}}$$

Assume the distribution parameters are not known, place priors over each and probabilistically estimate!

$$\mu \sim \text{Normal}(\gamma,\ \sigma^2 \upsilon^{-1})$$
$$\sigma^2 \sim \Gamma^{-1}(\alpha, \beta)$$

$$\underbrace{\mu, \sigma^2}_{\text{Distribution parameters}} \sim \underbrace{\text{NormalInvGamma}}_{\text{Evidential Prior}}\underbrace{(\gamma, \upsilon, \alpha, \beta)}_{\text{Model parameters}}$$



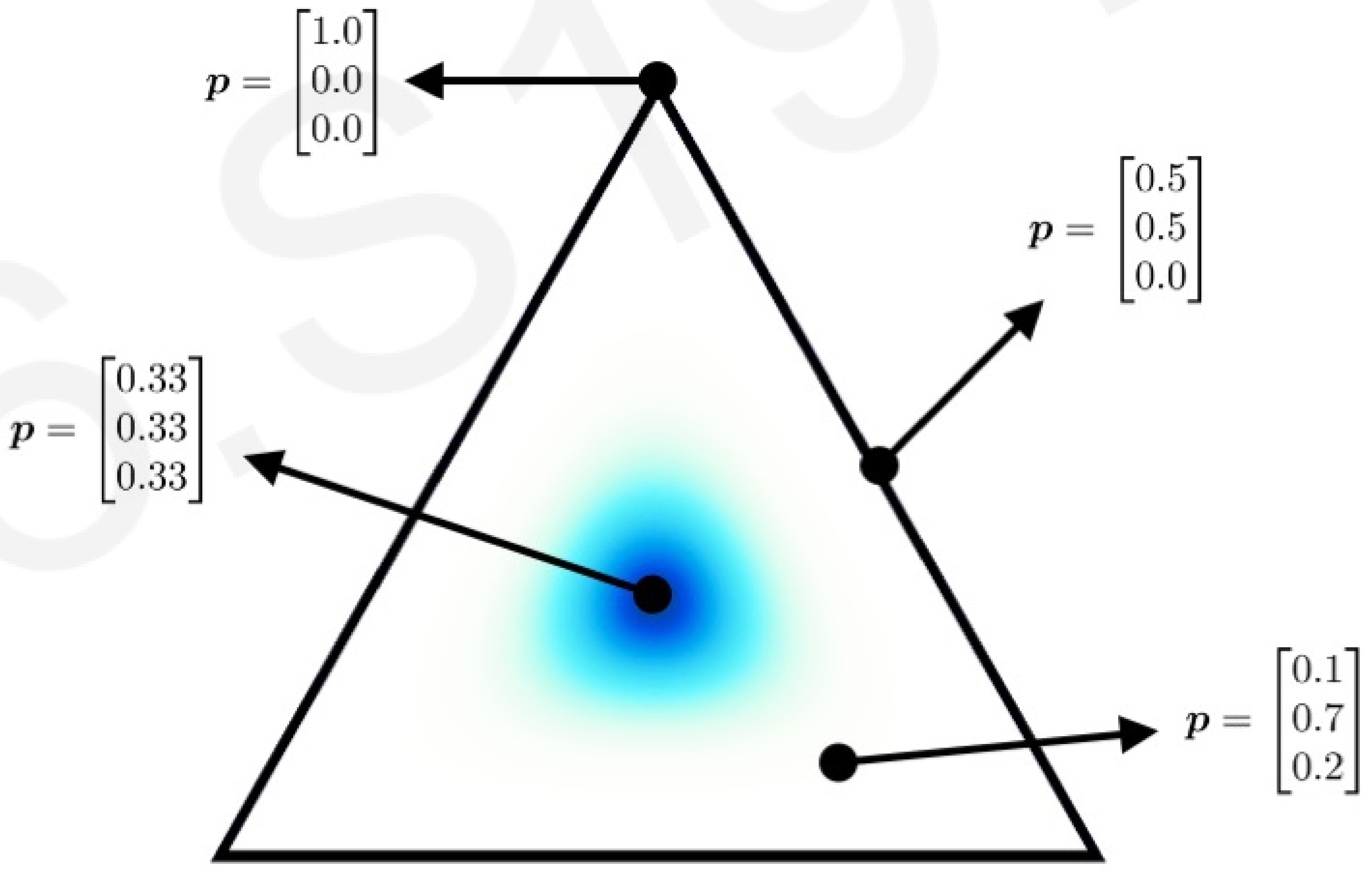
Evidential learning for classification

Sampling from an evidential distribution yields individual new distributions over the data

$$y \in \{1, \cdots, K\}$$

$$\underbrace{y}_{\text{Class Labels}} \sim \underbrace{\text{Categorical}}_{\text{Likelihood function}}(\underbrace{\boldsymbol{p}}_{\text{Distribution parameters (probabilities)}})$$

$$\underbrace{\boldsymbol{p}}_{\text{Distribution parameters}} \sim \underbrace{\text{Dirichlet}}_{\text{Evidential Prior}}(\underbrace{\boldsymbol{\alpha}}_{\text{Model parameters}})$$



$$K = 3; \quad \boldsymbol{\alpha} = (5, 5, 5)$$

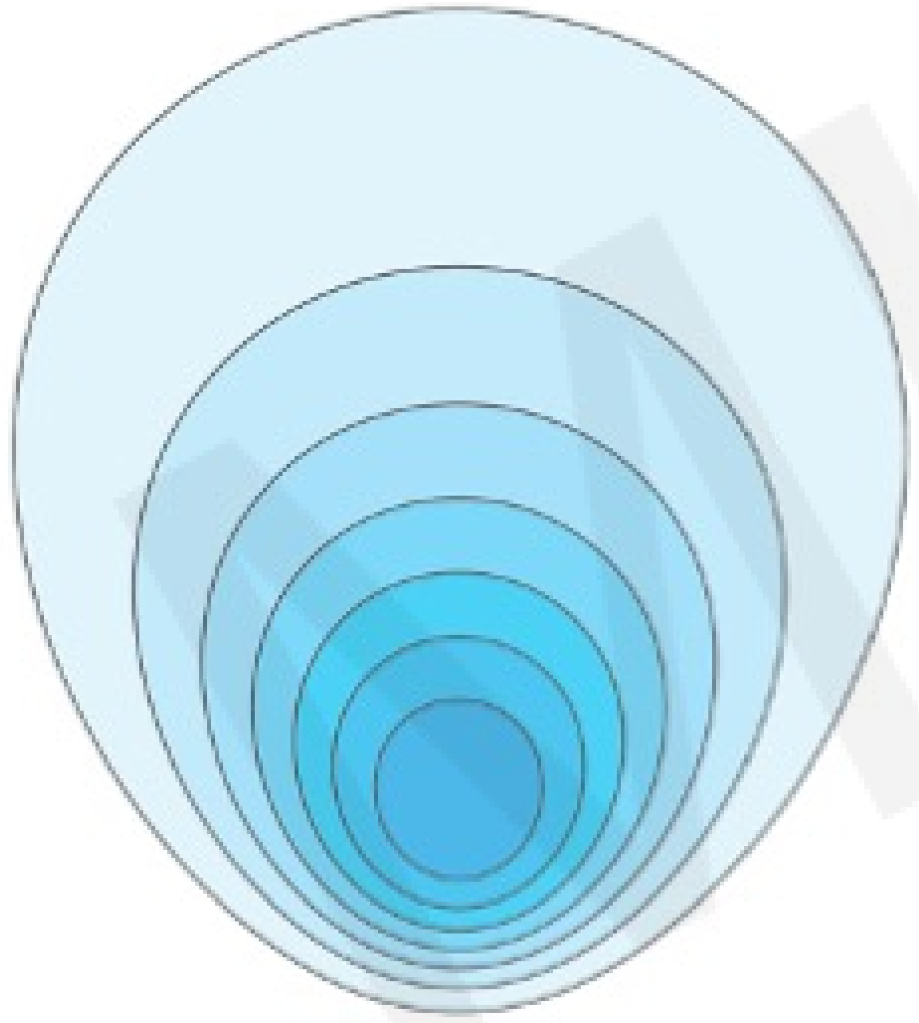
Evidential distributions for regression and classification

Regression (continuous)

$$y \in \mathbb{R}$$

$$y \sim \mathrm{Normal}(\mu, \sigma^2)$$

$$\mu, \sigma^2 \sim \mathrm{NormalInvGamma}(\gamma, \upsilon, \alpha, \beta)$$

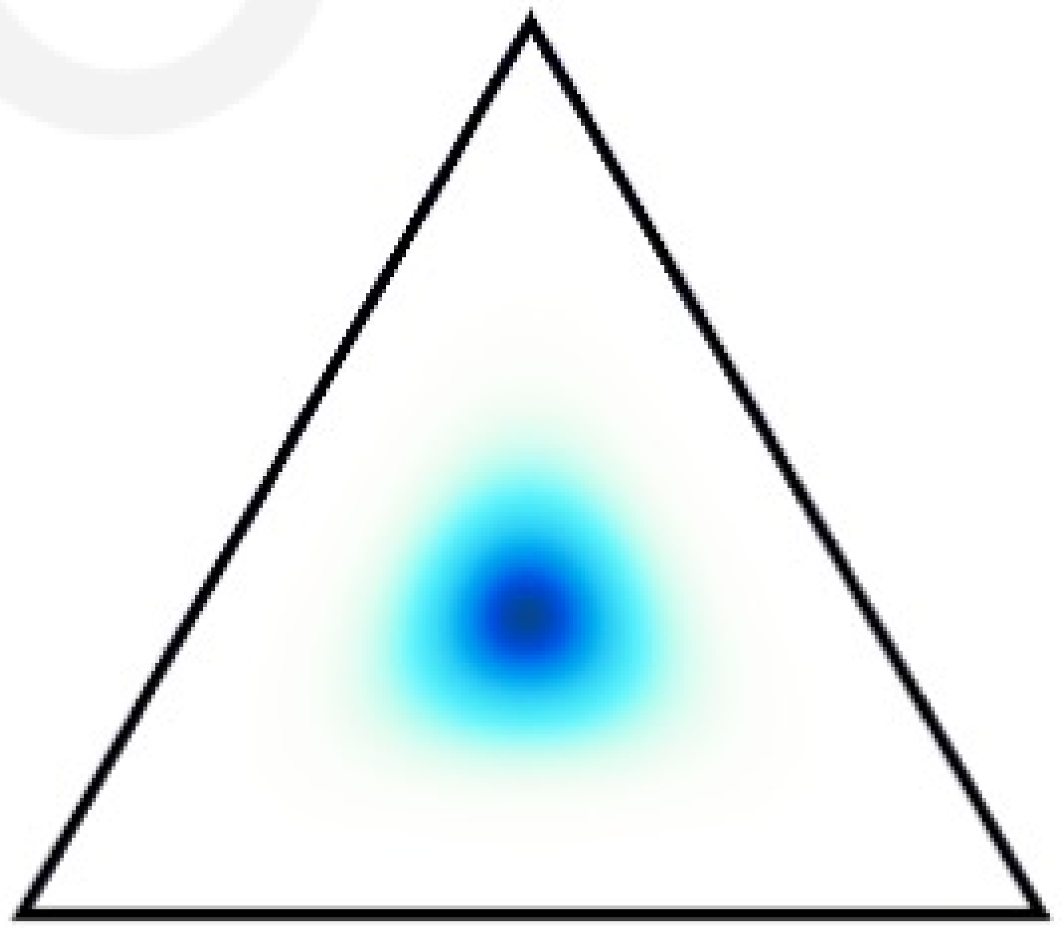


Classification (discrete)

$$y \in \{1, \cdots, K\}$$

$$y \sim \mathrm{Categorical}(\boldsymbol{p})$$

$$\boldsymbol{p} \sim \mathrm{Dirichlet}(\boldsymbol{\alpha})$$



Side note:

Choice of evidential distribution is closely related to conjugate priors in Bayesian inference.

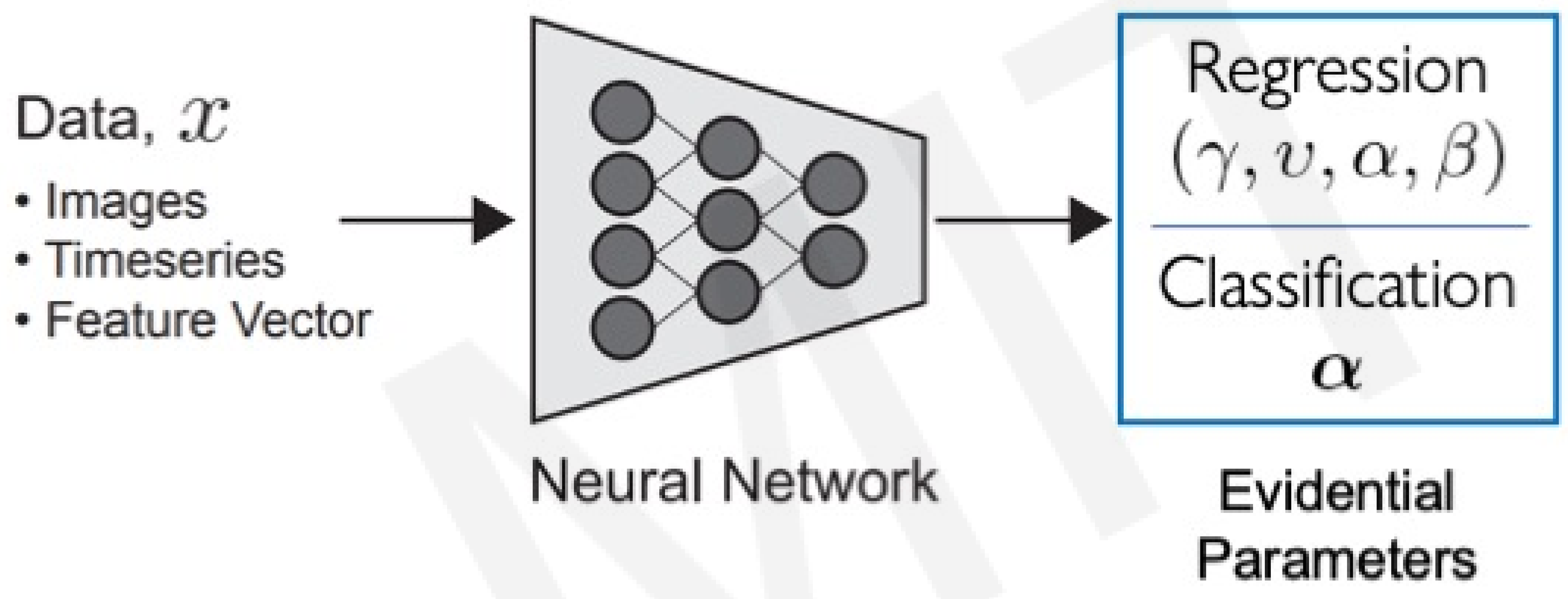
It is often easiest to pick your evidential distribution to be a conjugate prior of your likelihood

$$p(\theta|y) = \frac{p(y|\theta)\, p(\theta)}{\int_{\theta'} p(y|\theta')\, p(\theta')\, d\theta'}$$

Model and training

Model

Train the network to output the parameters of an evidential distribution



Optimization

Multi-objective training:

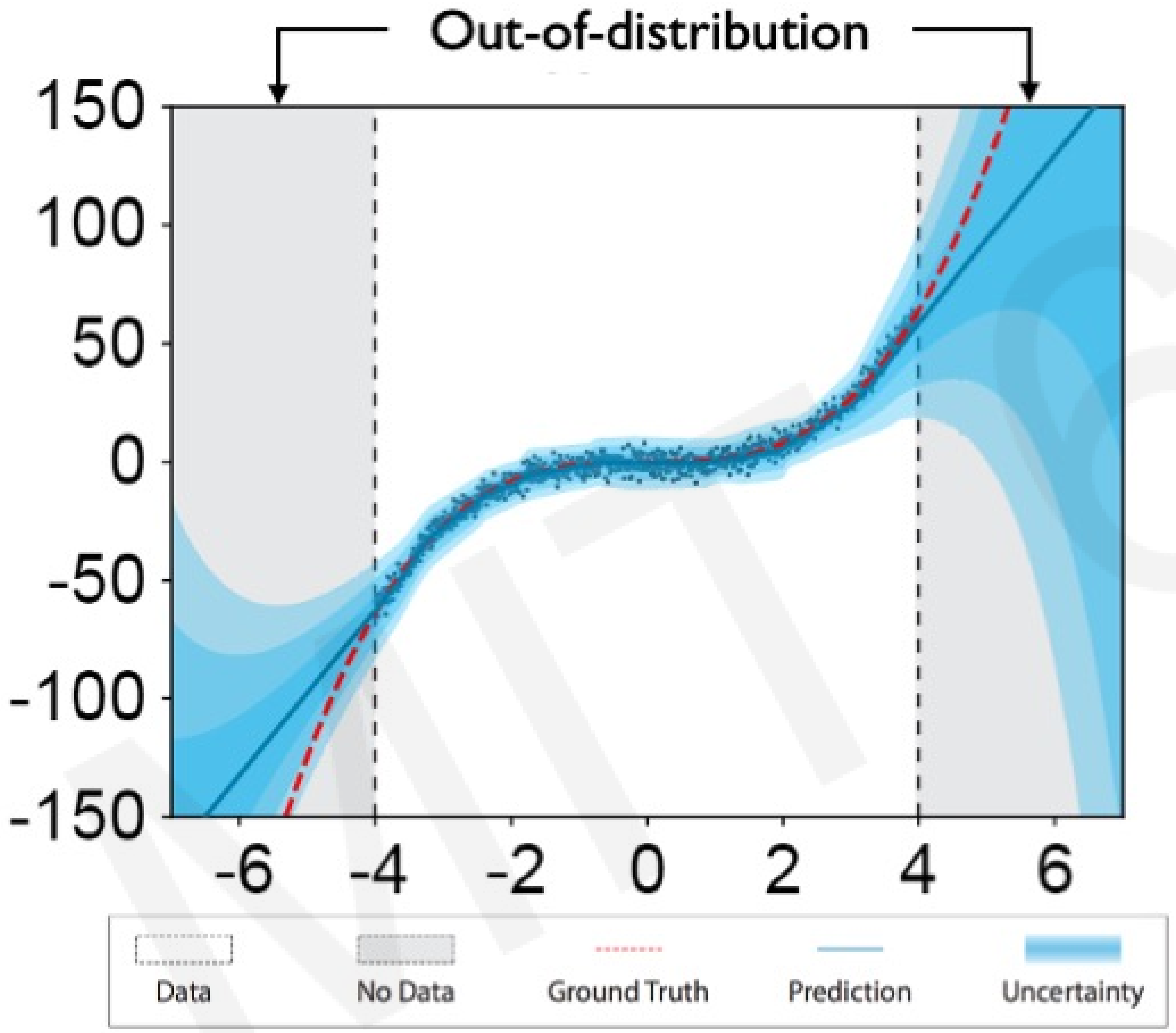
$$\int_{\boldsymbol{\theta}} \underbrace{p(y|\boldsymbol{\theta})}_{\text{likelihood data}} \underbrace{p(\boldsymbol{\theta}|\boldsymbol{m})}_{\text{evidential prior}} d\boldsymbol{\theta}$$

$$\underbrace{\|y - \mathbb{E}[\mu]\|}_{\text{errors}} \cdot \underbrace{\Phi(\boldsymbol{m})}_{\text{predicted evidence}}$$

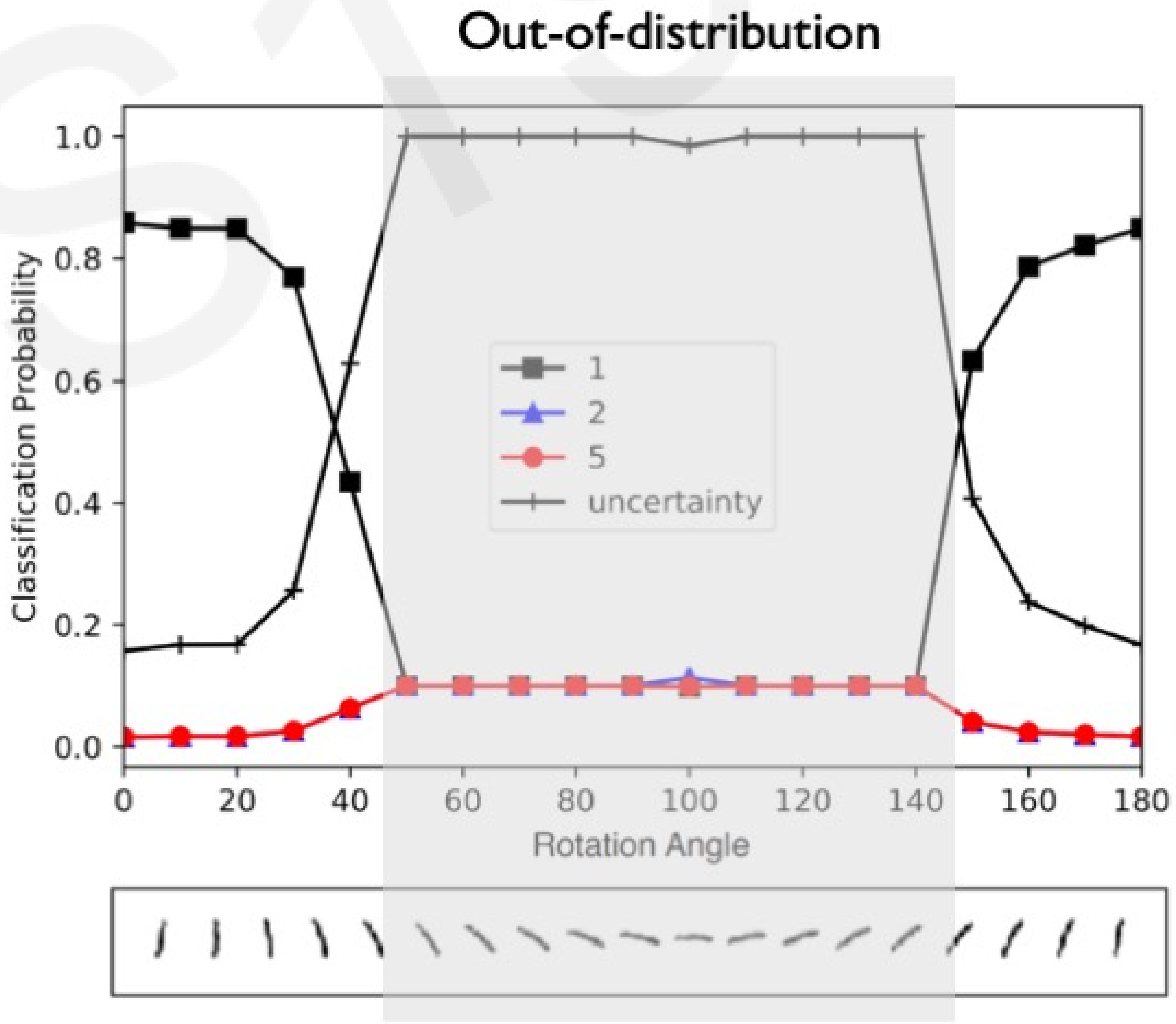
Amini+ NeurIPS 2020. Sensoy+ *NeurIPS* 2018.

Toy learning problems

Regression (continuous)



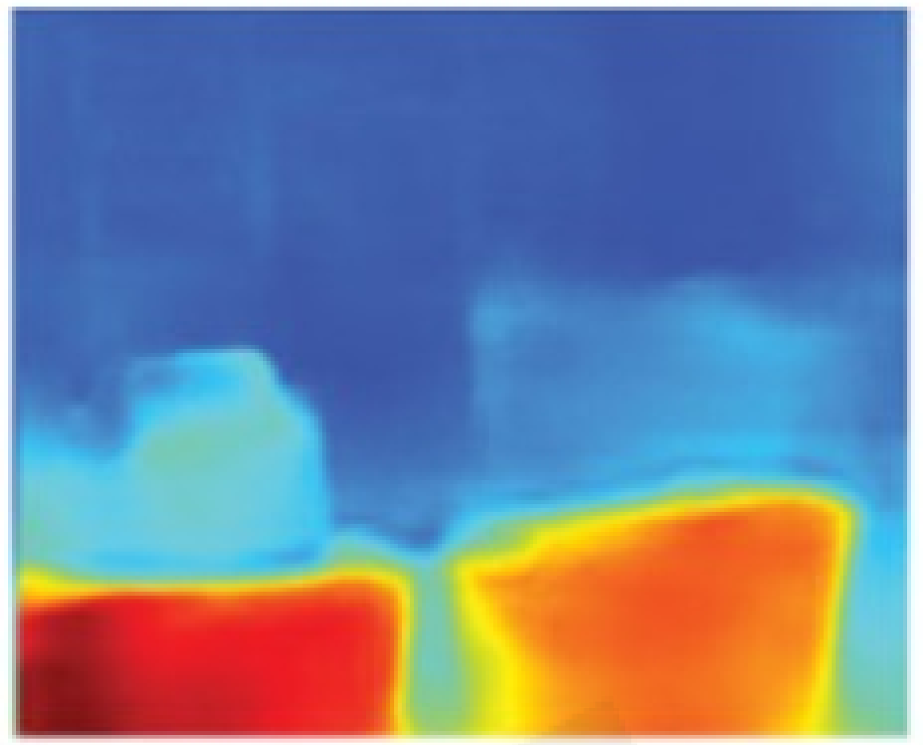
Classification (discrete)



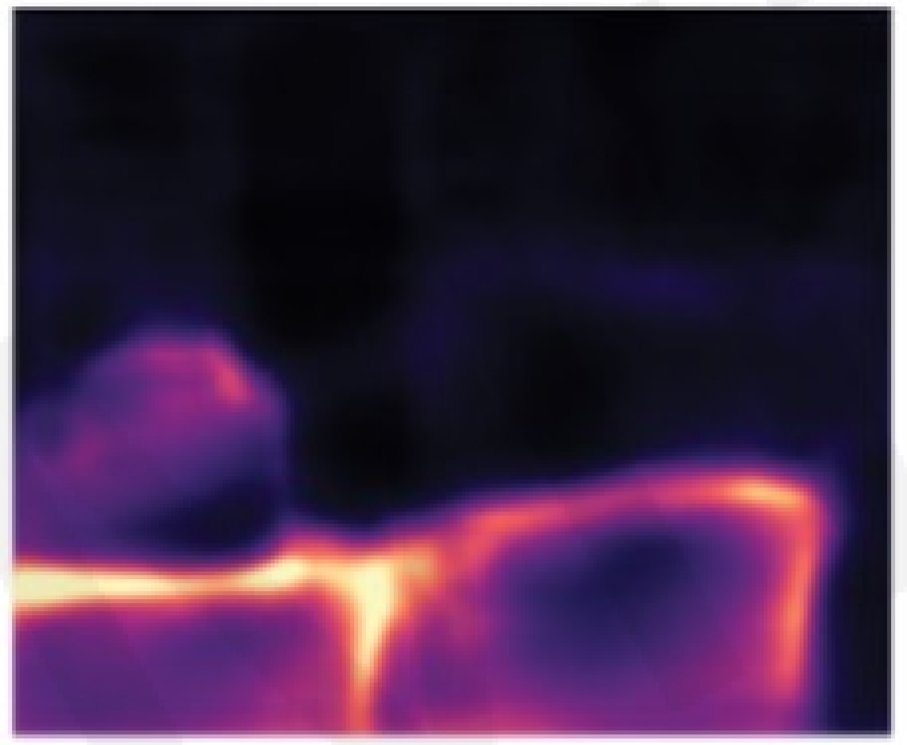
Applications of evidential learning

Monocular Depth Estimation

RGB input

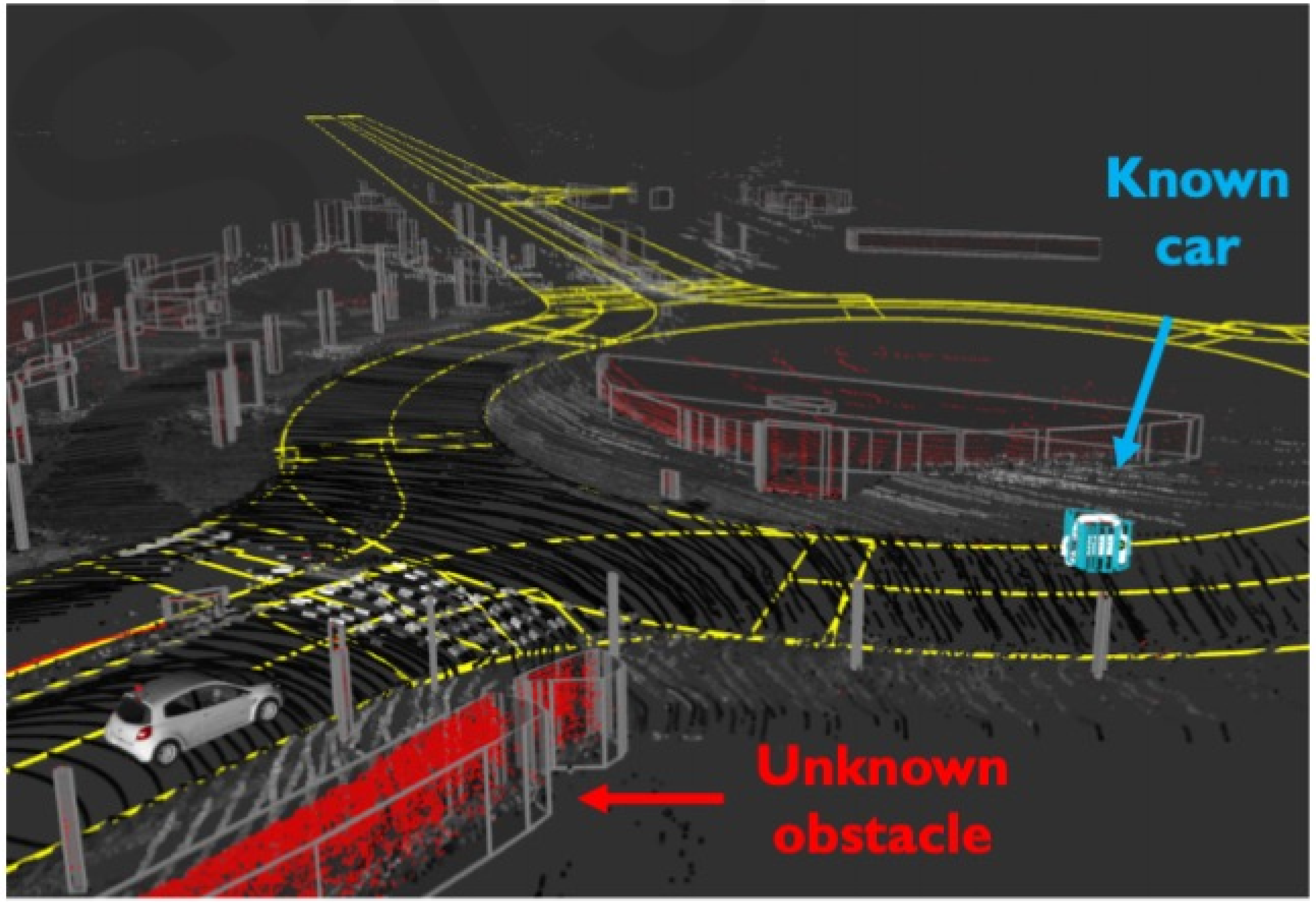


Predicted depth



Predicted uncertainty

LiDAR Object Classification

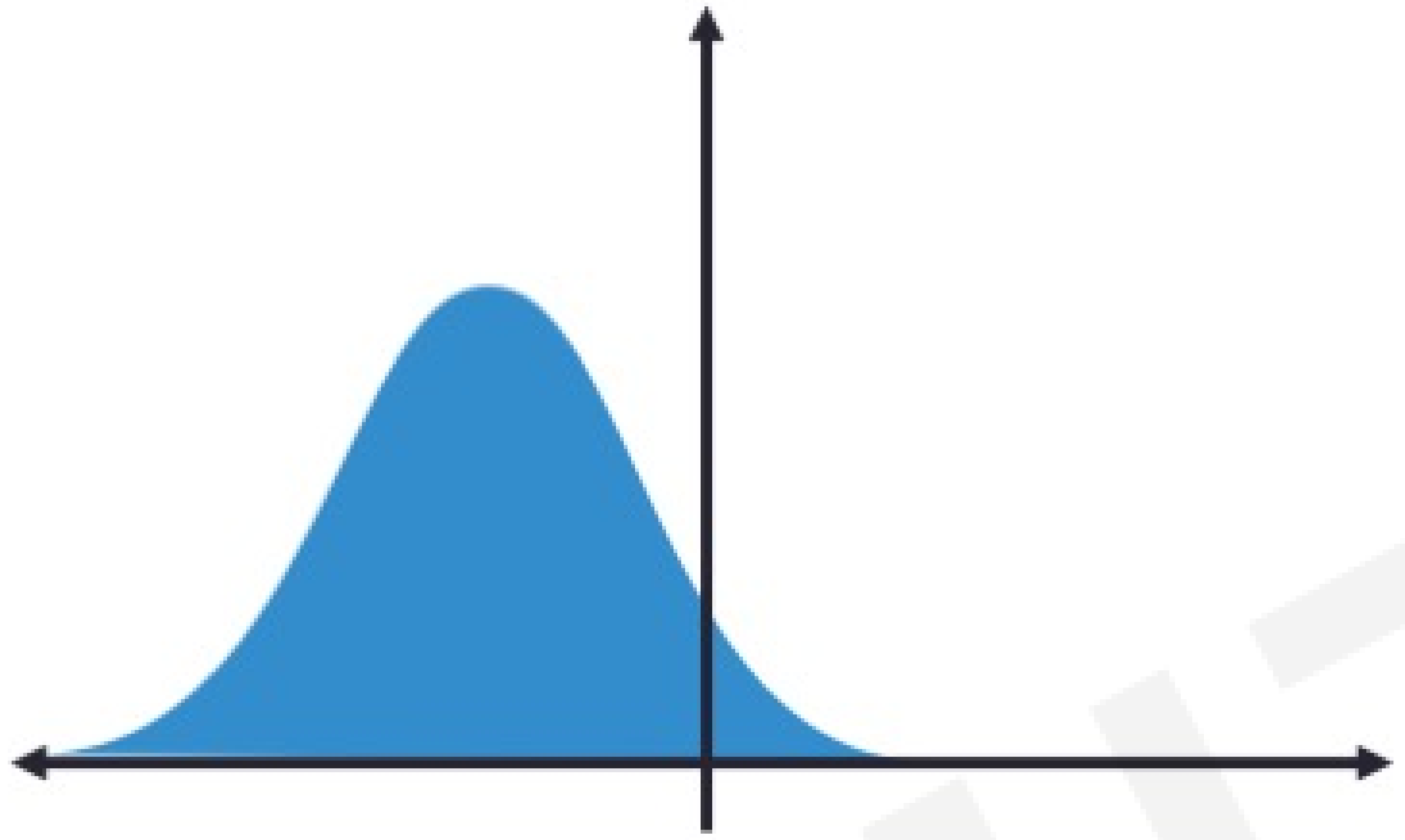


Comparison of uncertainty estimation approaches

	Likelihood estimation	Bayesian NN	Evidential NN
Prior placed over:	Data	Weights	Likelihood
Weights are:	Deterministic	Stochastic	Deterministic
Fast (no sampling)	✓		✓
Captures epistemic uncertainty		✓	✓

Summary

Learning probability distributions



Model distributions over labels: Softmax (discrete) & Gaussian (continuous)

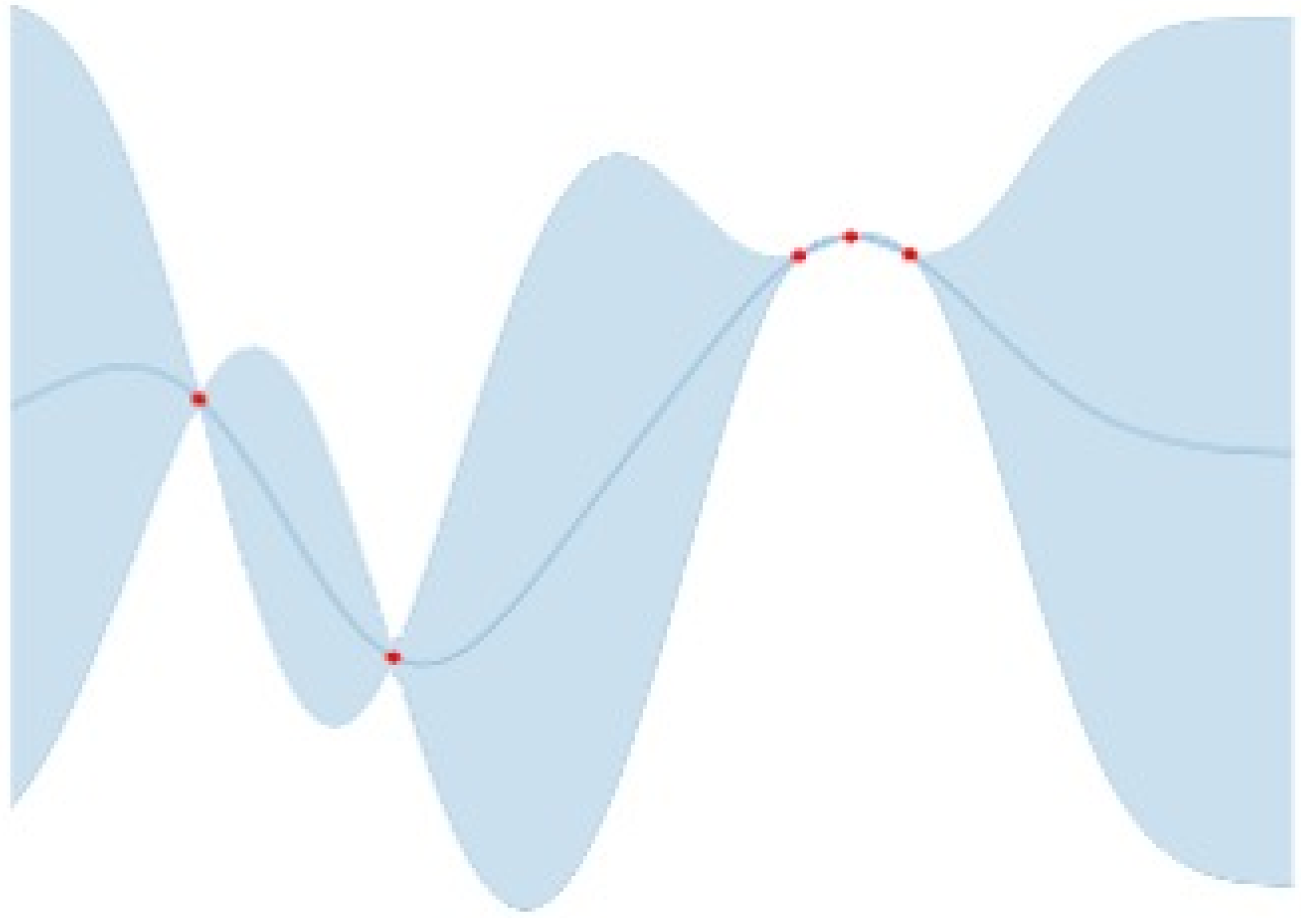
Different sources of uncertainty

$$p(y|x)$$

$$\mathbb{E}[\sigma^2] \qquad \mathrm{Var}[\mu]$$

Data (aleatoric) uncertainty vs. Model (epistemic) uncertainty.

Fast and scalable uncertainty estimation



Evidential deep learning
Uncertainty modelling for quickly estimating confidence