FCEN, UBA

Inferencia Bayesian

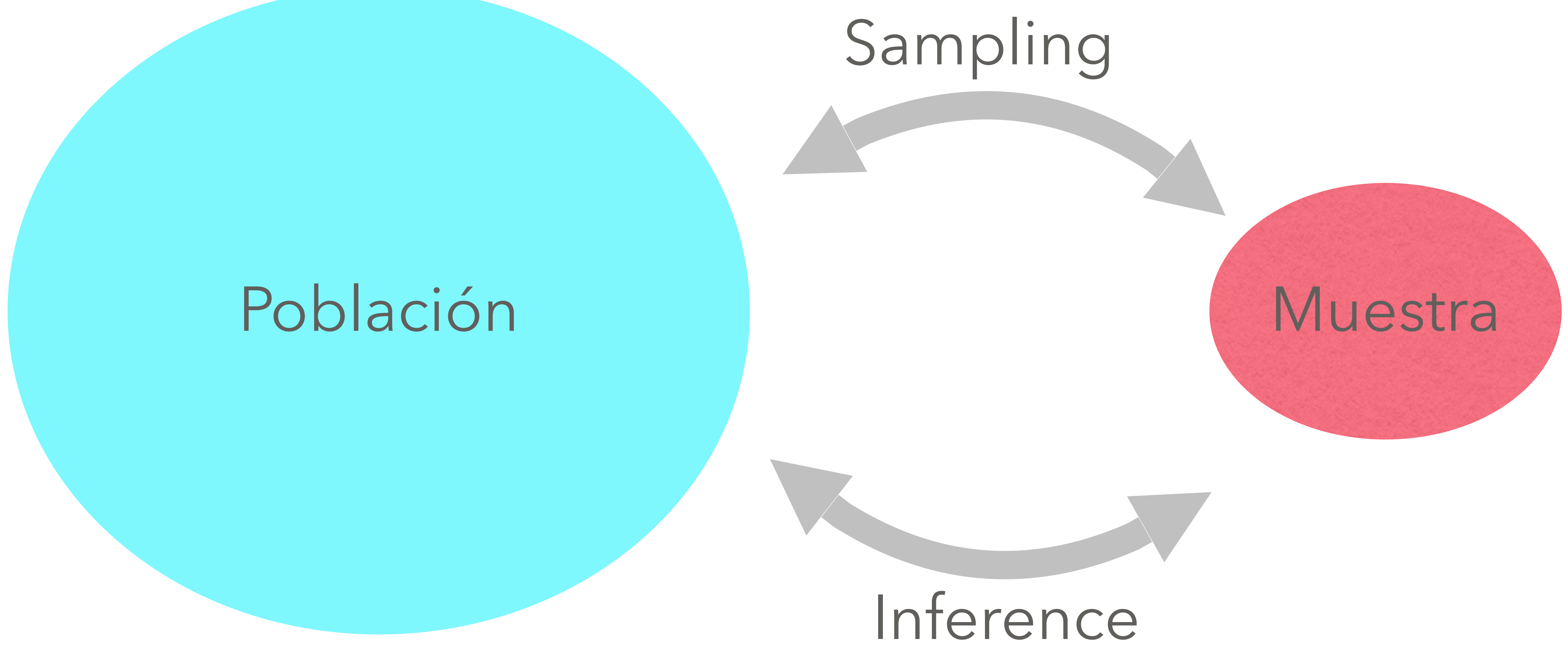
Cecilia Garraffo

Outline

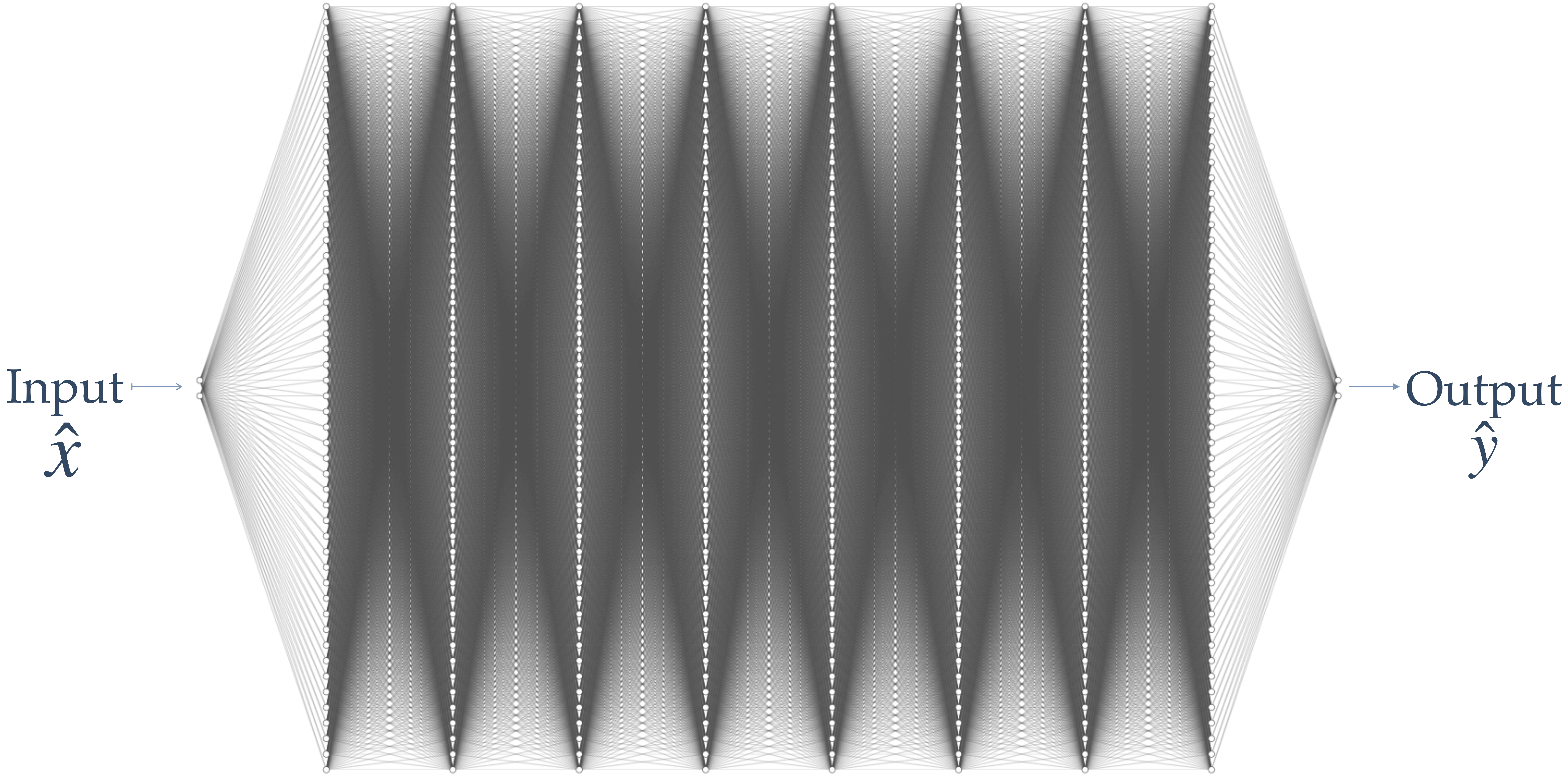
- Inferencia Bayesian
- Markov Chain Monte Carlo
- Inferencia Variacional
- Redes Neuronales Bayesianas
- Bayes con backprop y flipout
- Mean Field Variational Bayes

Inferencia Estadística

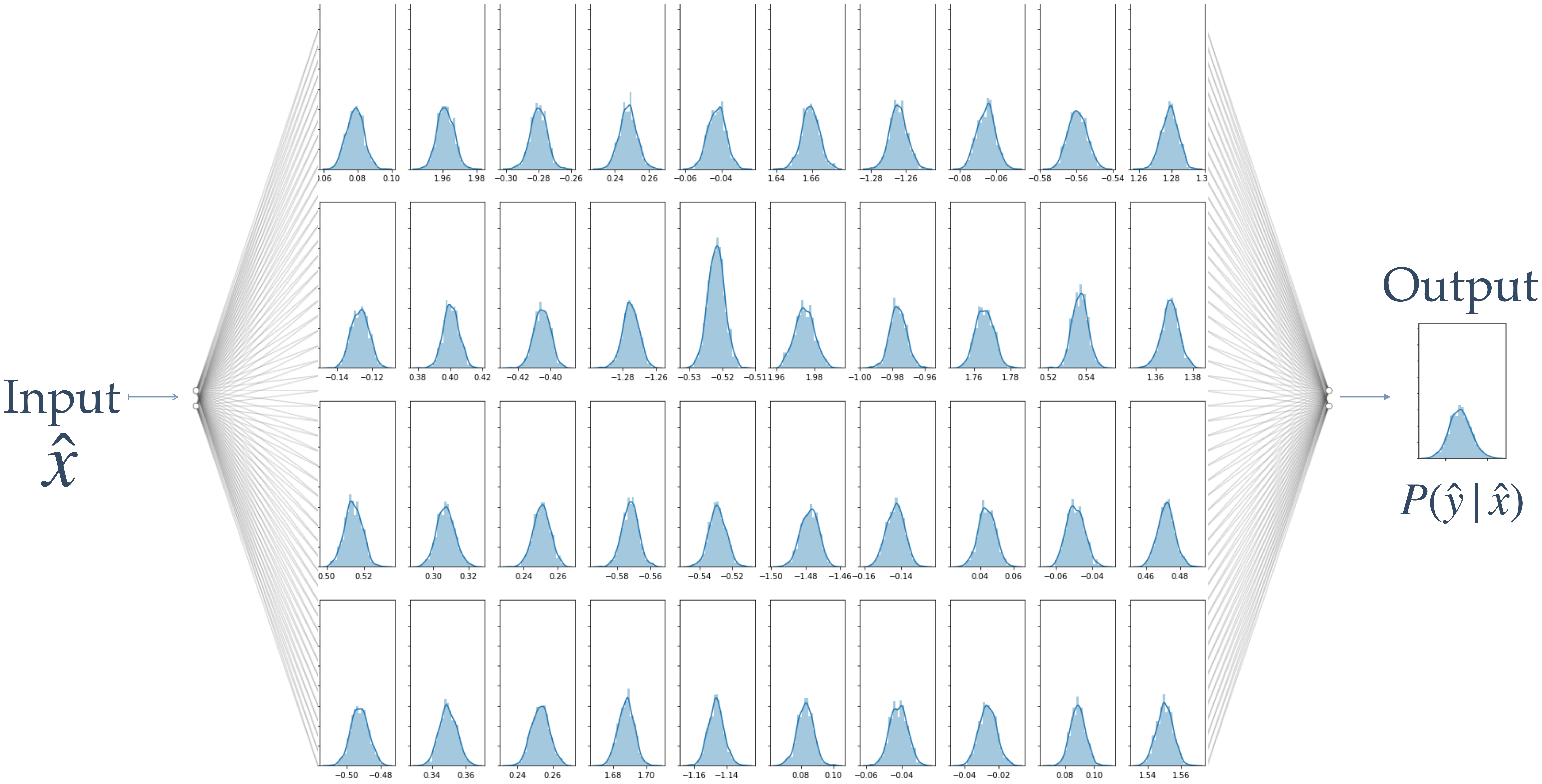
Sacar conclusions de una distribución de probabilidades a partir de una muestra



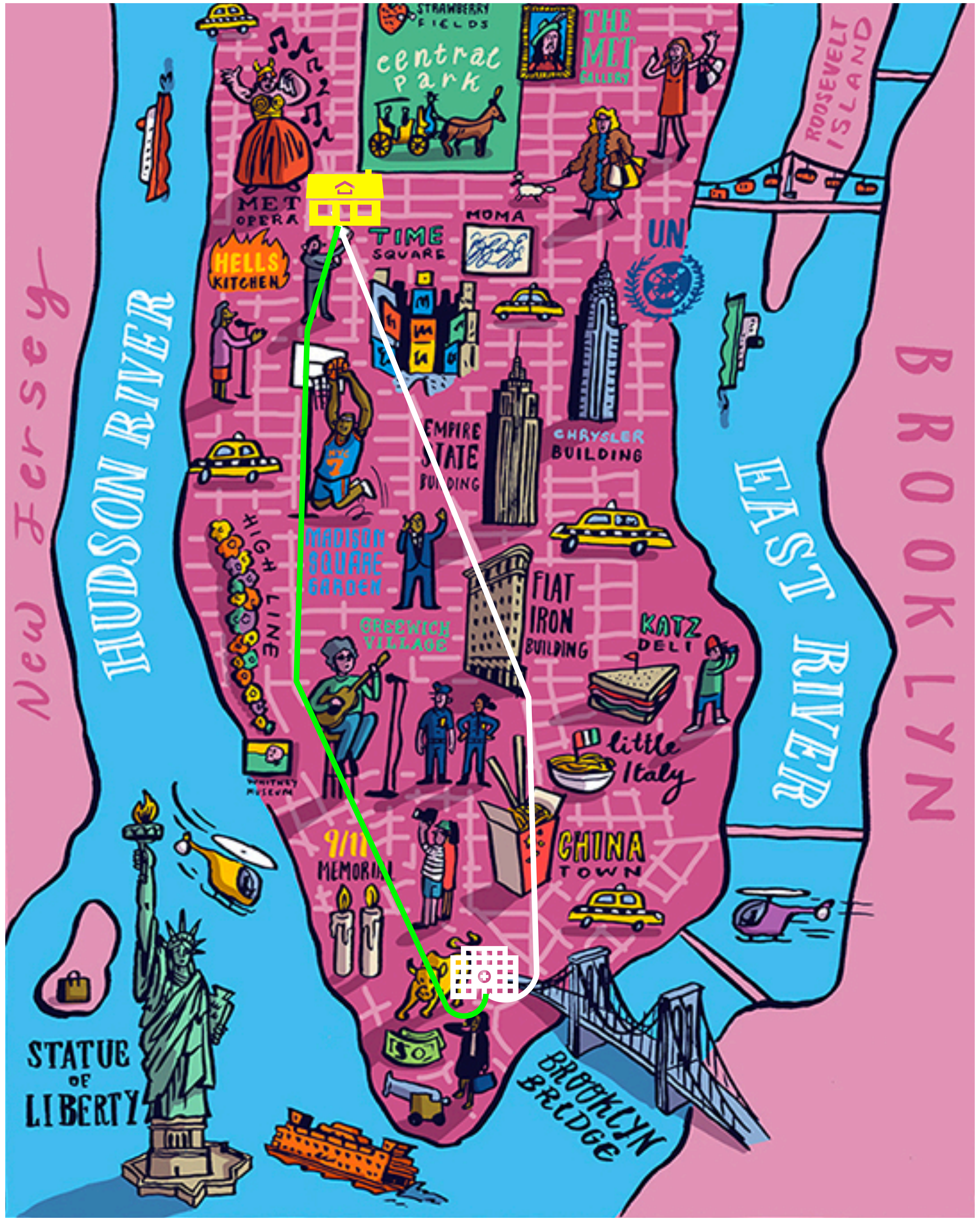
Inference in Machine Learning



Inference in Machine Learning



Statistical Inference with Quantified Uncertainty



Your house Columbus Circle

NY-Presbyterian Hospital / Lower Manhattan

Route 1 ⟶ 25min ± 9min

Route 2 ⟶ 29min ± 2min

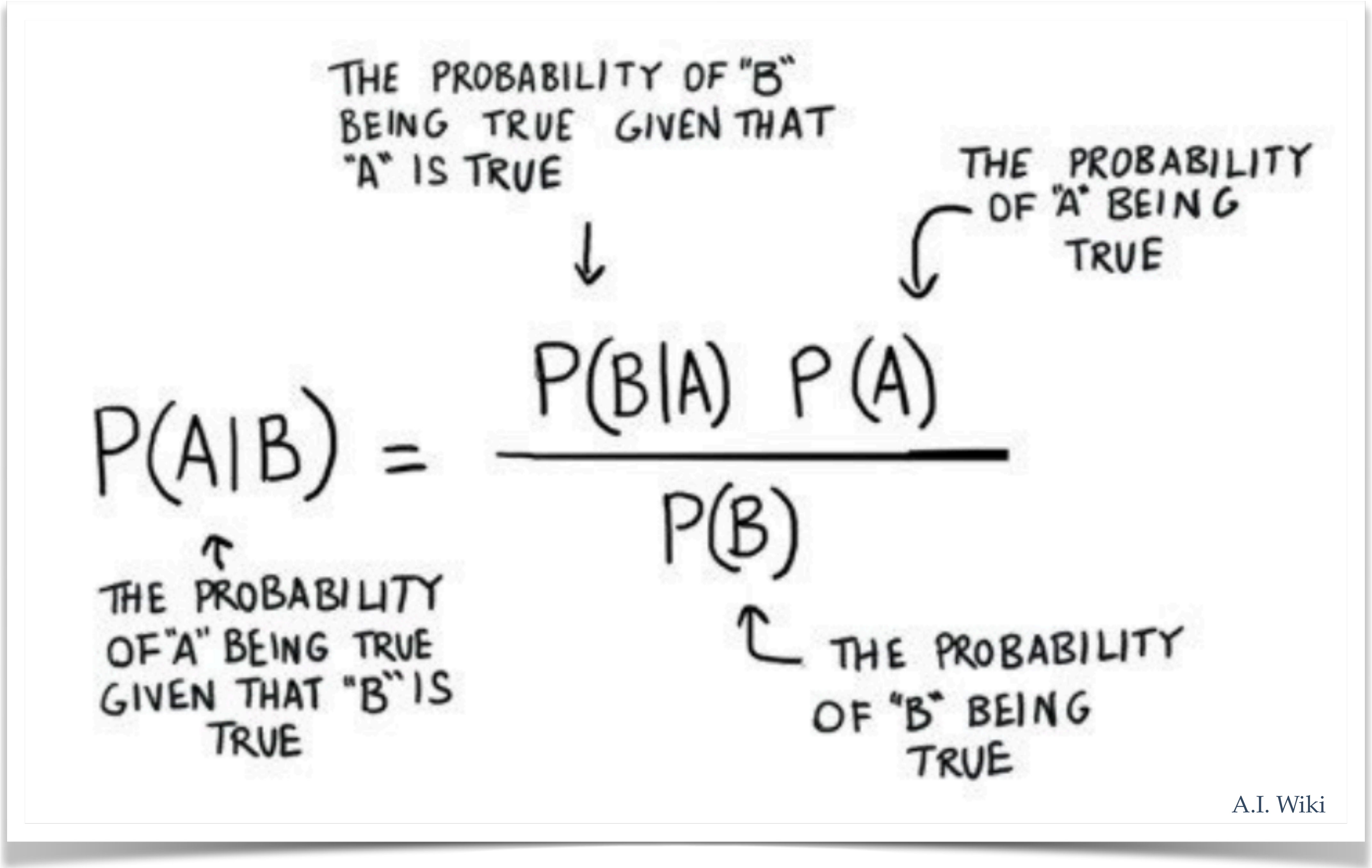
Uncertainties are important to make informed decisions

Frequentist approach (Bootstrapping)

Bayesian Inference (Includes prior knowledge)

Bayesian Inference

Probability as a measure of *believability in an event*



Likelihood Prior

$$p(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{p(y)}$$

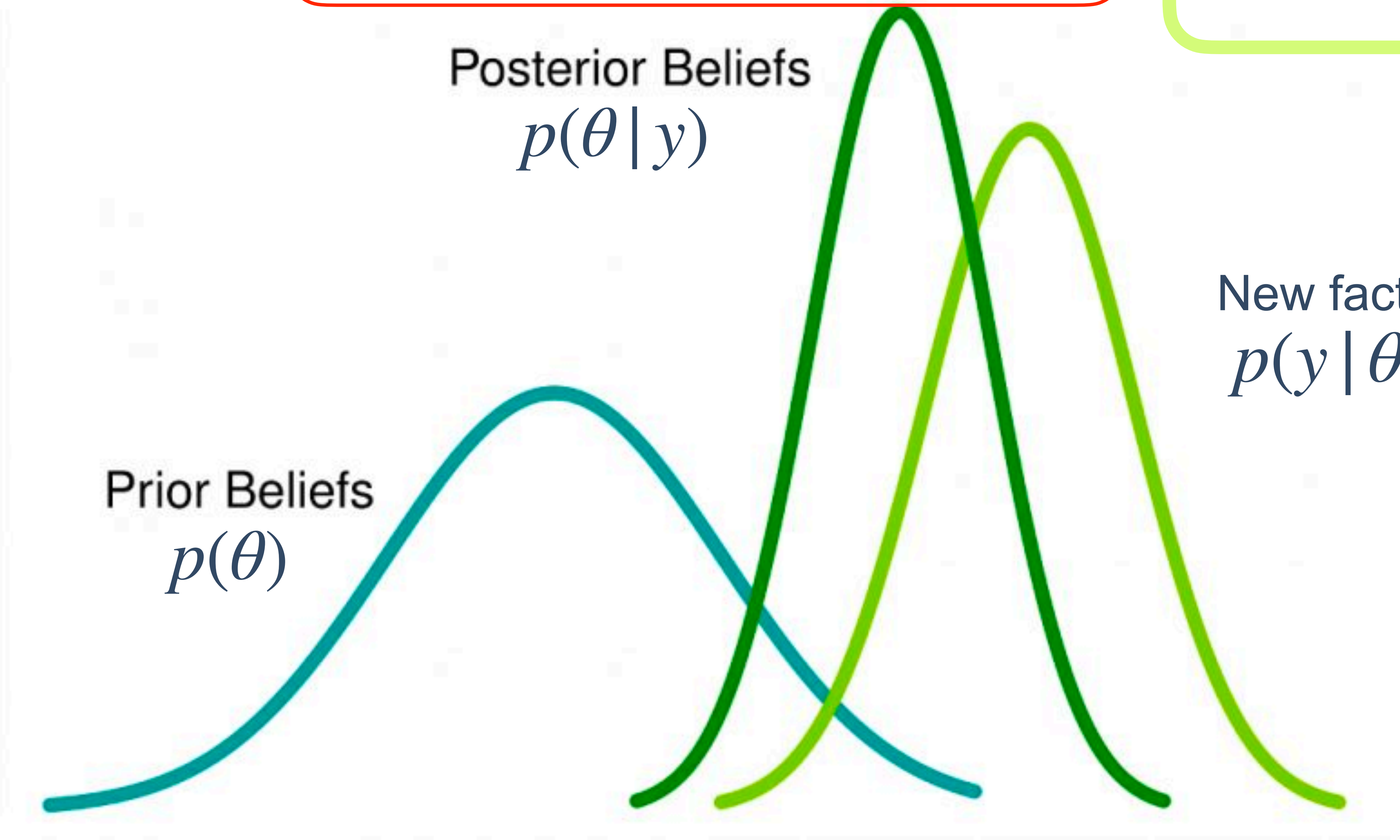
Model Data Evidence

Bayesian Inference

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}$$

Intractable

$$p(y) = \int_\theta p(y|\theta)p(\theta)d\theta$$



"When the facts change, I change my mind. What do you do, sir? "

John Maynard Keynes

Bayesian Inference: Markov Chain Monte Carlo

$$p(\theta \,|\, y) \propto p(y \,|\, \theta) p(\theta)$$

$$p(\theta^0 \,|\, y) \propto p(y \,|\, \theta^0) p(\theta^0)$$

$$\theta = (\mu, \sigma)$$

$$\mu^j \sim N(\mu^{(j-1)}, \sigma)$$

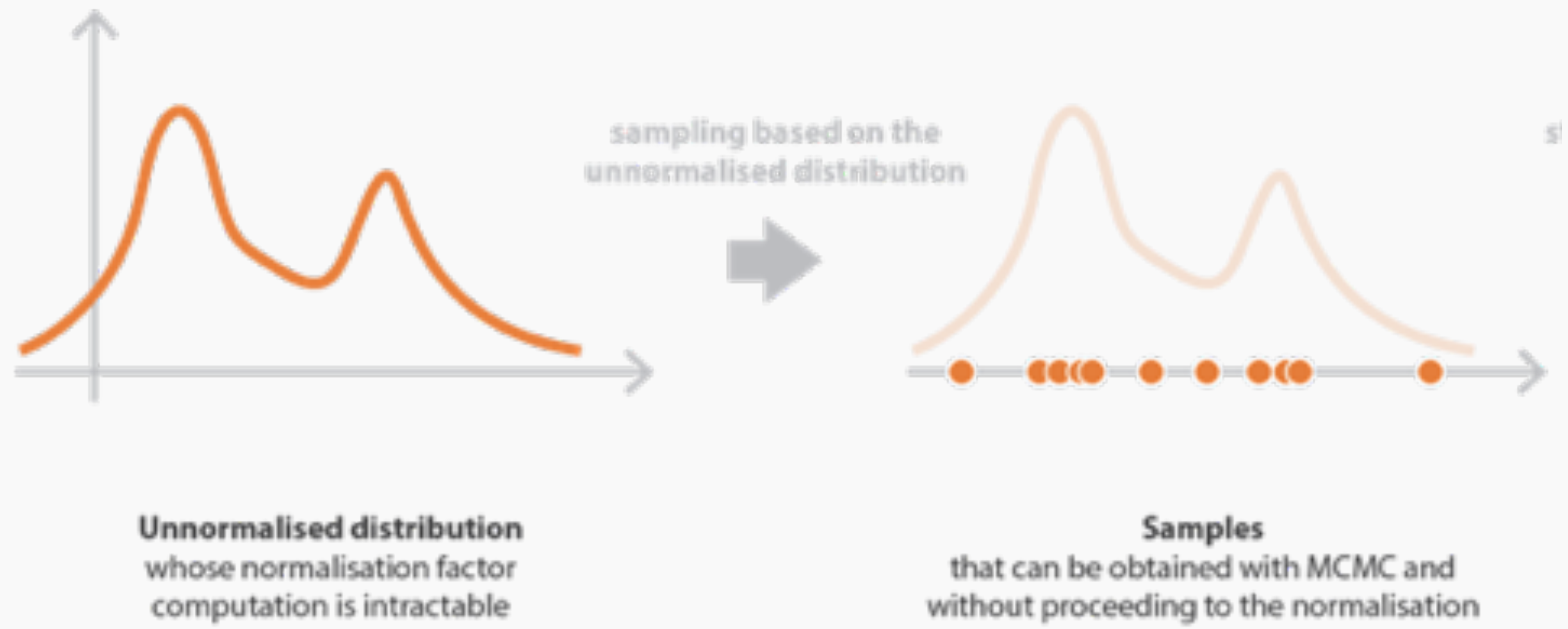
Step of random walk

$$\frac{p(\theta^j \,|\, y)}{p(\theta^{j-1} \,|\, y)} = \frac{p(y \,|\, \theta^i) p(\theta^i)}{p(y \,|\, \theta^{j-1}) p(\theta^{j-1})}$$

>1 → accept

<1 → flip a coin

Histogram of my samples will be identical to the true posterior distribution… eventually

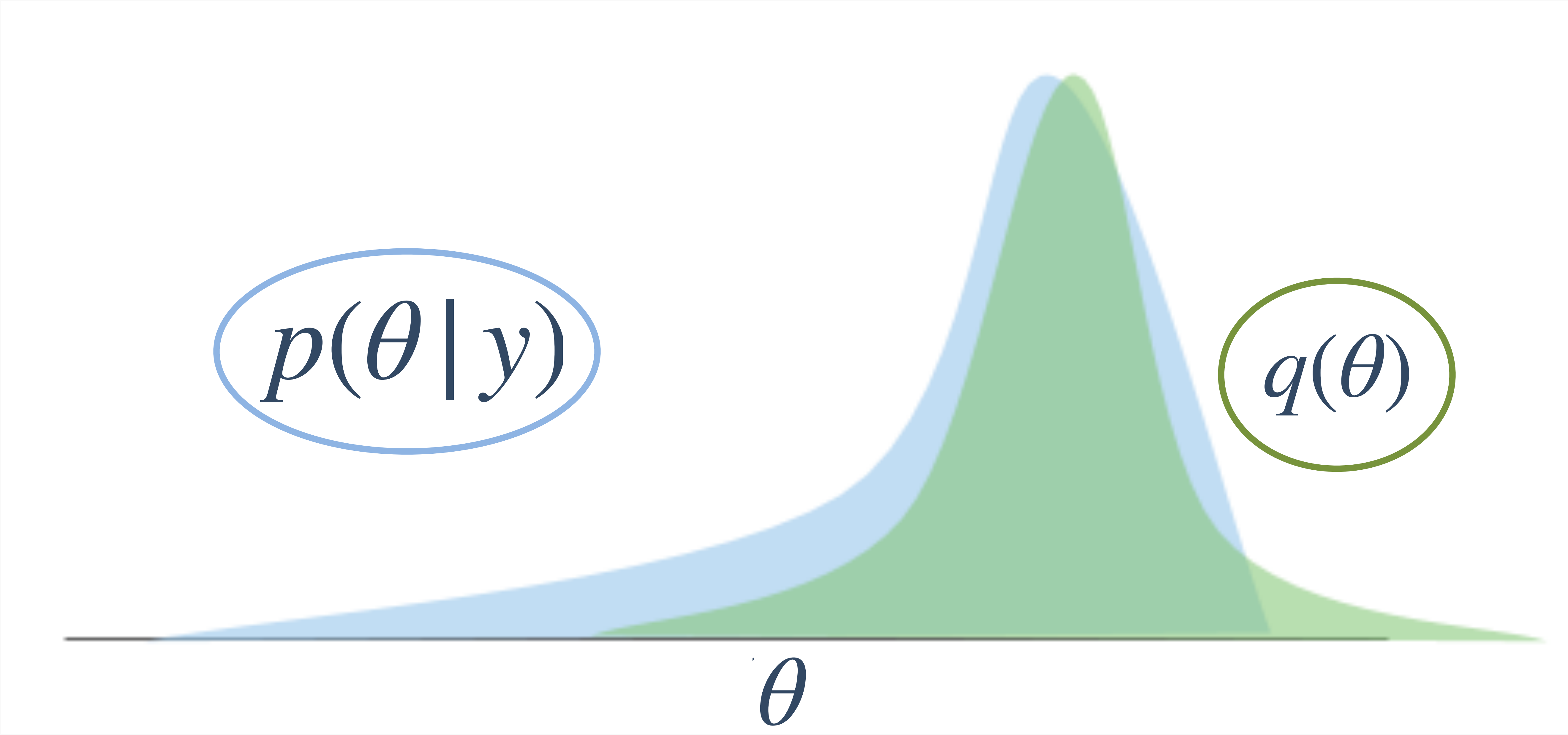


MCMC is eventually accurate, but not scalable to large models

Approximate Bayesian Inference: Variational Inference

Optimization approach -> Q a family of "nice" distributions

$$p(\theta \mid y) = \frac{p(y \mid \theta)\, p(\theta)}{\int p(y, \theta)\, d\theta}$$



Approximate Bayesian Inference: Variational Inference

$$p(\theta \mid y) = \frac{p(y \mid \theta)\, p(\theta)}{\int p(y, \theta) d\theta} \quad \int p(y, \theta) d\theta = p(y) \longleftarrow \text{evidence}$$

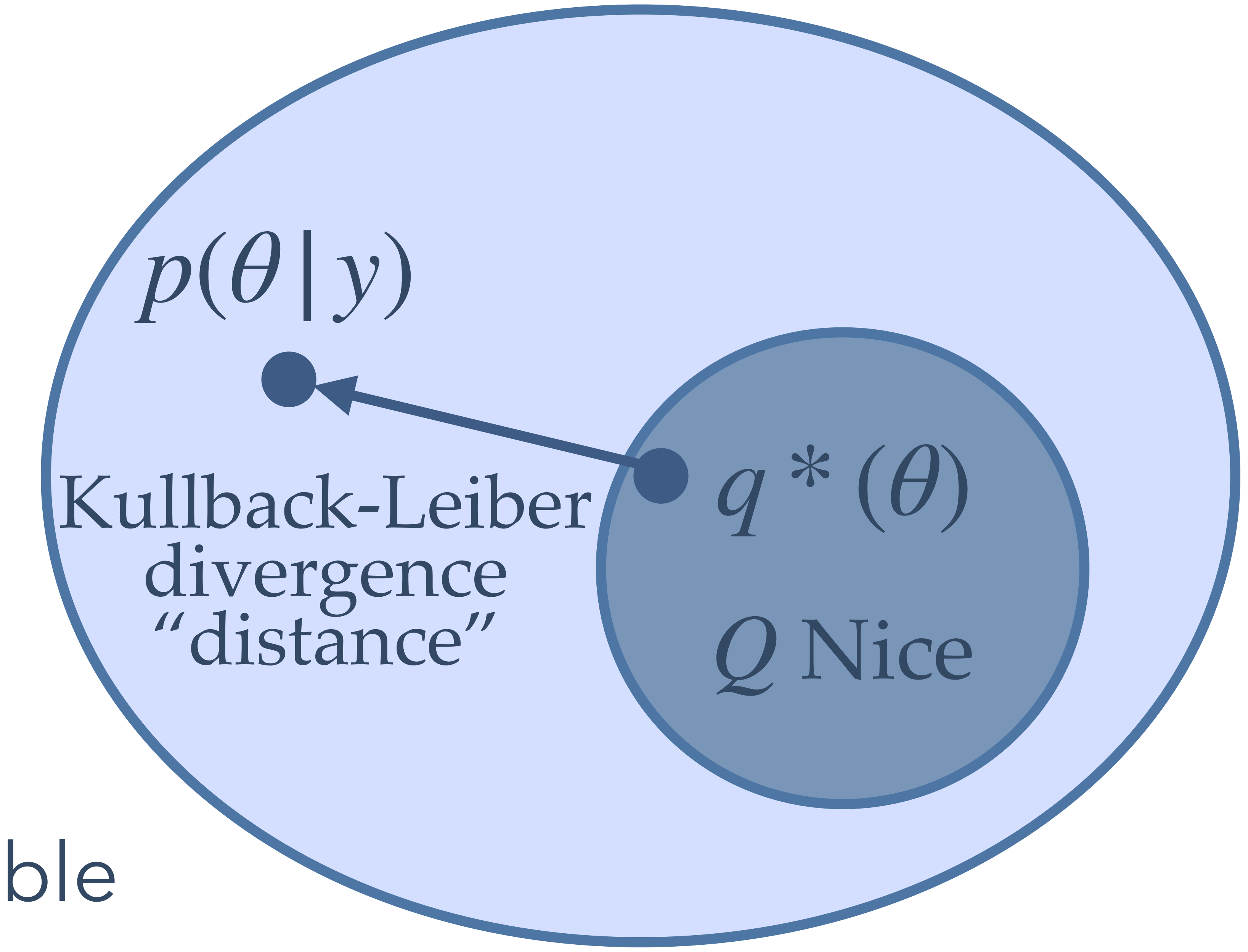
$$p(y) = \int \int_{\dots} \int p(y \mid \theta_1, \theta_2, \theta_3, \dots, \theta_m) d\theta_1 d\theta_2 d\theta_3 \dots d\theta_m$$

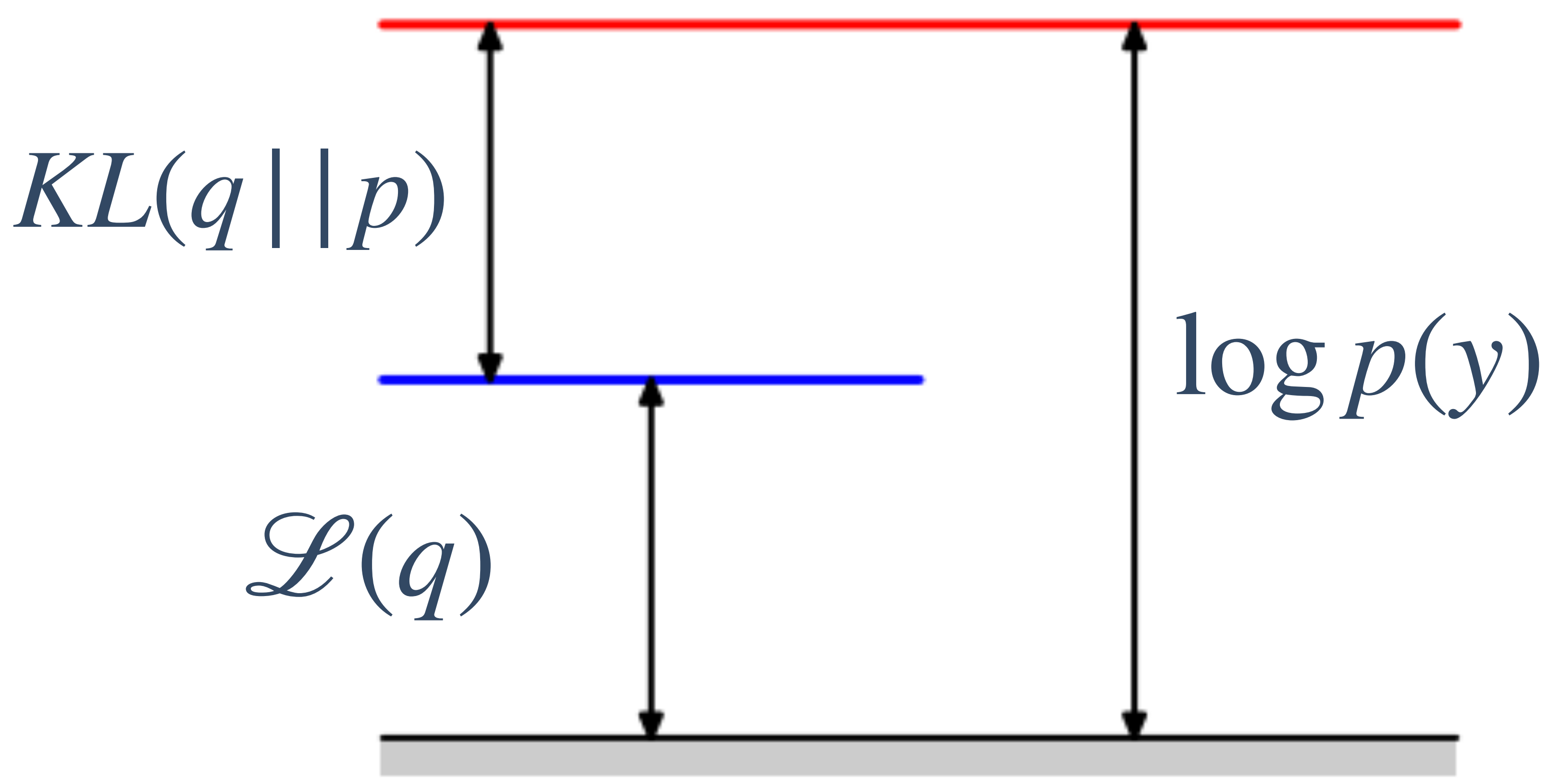
intractable

approximation

$$p(\theta \mid y) \approx q^*(\theta) \longleftarrow \text{tractable family of distributions}$$

$$q^*(\theta) = argmin_{q \in Q} KL(q(\cdot)) \mid\mid p(\cdot \mid y))$$





Bishop

Approximate Bayesian Inference: Variational Inference

$$q^*(\theta) = argmin_{q \in Q} KL(q(\cdot))||p(\cdot|y))$$

we need to optimize for q

$$\theta^* = argmin_{\theta} KL[(q(\omega|\theta))||p(\omega|y)]$$

we can, instead, find the parameters θ of a distribution on the weights $q(\omega|\theta)$

$$KL(q(\cdot))||p(\cdot|y) := \int q(\theta) \log \frac{q(\theta)}{p(\theta|y)} d\theta$$

$$p(\omega|y) = \frac{p(y|\omega)p(\omega)}{p(y)}$$

$$\theta^* = argmin_{\theta} \int (q(\omega|\theta)) \log \frac{q(\omega|\theta)p(y)}{p(\omega)p(y|\omega)} d\omega$$

$$= argmin_{\theta} \int q(\omega|\theta) \log \frac{q(\omega|\theta)}{p(\omega)p(y|\omega)} d\omega \cancel{+ argmin_{\theta}[\underbrace{\log p(y)}_{=\text{const}} \underbrace{\int q(\omega|\theta) d\omega}_{=1}]}$$

$$\theta^* = argmin_{\theta}\left[KL[q(\omega|\theta))||p(\omega)] - \mathbb{E}_{q(\omega|\theta)}[\log p(y|\omega)]\right]$$

$$\mathcal{L}(y|\theta) = KL[q(\omega|\theta))||p(\omega)] - \mathbb{E}_{q(\omega|\theta)}[\log p(y|\omega)]$$ ← Loss function

Prior Likelihood

Approximate Bayesian Inference: Bayes by Backprop

$$\mathscr{L}(y\,|\,\theta) = KL[q(\omega\,|\,\theta))\,||\,p(\omega)] - \mathbb{E}_{q(\omega|\theta)}[\log p(y\,|\,\omega)] \longleftarrow \text{Loss function}$$

Prior — Likelihood

Back propagate

$$\nabla_\theta \mathscr{L}(y\,|\,\theta)$$

Approximate Bayesian Inference: Bayes by Backprop

$$\mathscr{L}(y\,|\,\theta) = KL[q(\omega\,|\,\theta))\,||\,p(\omega)] - \mathbb{E}_{q(\omega|\theta)}[\log p(y\,|\,\omega)]$$ ⟵ Loss function

Prior Likelihood

Back propagate

$$\nabla_\theta \mathscr{L}(y\,|\,\theta) \to \nabla_\theta \mathscr{L}(\omega, \theta)$$

Let's define the *pointwise* (sampled) loss for a given ω and θ:

$$\mathcal{L}(\omega, \theta) = \log q(\omega|\theta) - \log p(\omega) - \log p(y|\omega)$$

Notice that if you take the expectation under $q(\omega|\theta)$:

$$\mathbb{E}_{q(\omega|\theta)}[\mathcal{L}(\omega, \theta)] = \mathrm{KL}[q(\omega|\theta) \,\|\, p(\omega)] - \mathbb{E}_{q(\omega|\theta)}[\log p(y|\omega)] = \mathcal{L}(y|\theta)$$

Approximate Bayesian Inference: Bayes by Backprop

$$\mathscr{L}(y|\theta) = \underbrace{KL[q(\omega|\theta)) || p(\omega)]}_{\text{Prior}} - \underbrace{\mathbb{E}_{q(\omega|\theta)}[\log p(y|\omega)]}_{\text{Likelihood}}$$

$\longleftarrow$ Loss function

$\downarrow$ Back propagate

$$\nabla_\theta \mathscr{L}(y|\theta) \rightarrow \nabla_\theta \mathscr{L}(\omega, \theta)$$

$$q(\theta) = N(\overbrace{\mu, \sigma}^{\theta})$$

ω are samples of this distribution

$$\omega = \mu + \sigma \odot \epsilon \;, \quad q(\epsilon)\;, \quad q(\epsilon)d\epsilon = q(\omega|\theta)d\omega$$

Weight Uncertainty in Neural Networks,
Blundell et al. 1015(https://arxiv.org/pdf/1505.05424.pdf)

Suppose you can write $\omega = t(\epsilon, \theta)$, with $\epsilon \sim q(\epsilon)$ (independent of θ), so:

$$q(\omega|\theta)d\omega = q(\epsilon)d\epsilon$$

and thus,

$$\int f(\omega, \theta)q(\omega|\theta)d\omega = \int f(t(\epsilon, \theta), \theta)q(\epsilon)d\epsilon = \mathbb{E}_{q(\epsilon)}[f(t(\epsilon, \theta), \theta)]$$

Approximate Bayesian Inference: Bayes by Backprop

$$\mathscr{L}(y|\theta) = \underbrace{KL[q(\omega|\theta))||p(\omega)]}_{\text{Prior}} - \underbrace{\mathbb{E}_{q(\omega|\theta)}[\log p(y|\omega)]}_{\text{Likelihood}} \longleftarrow \text{Loss function}$$

Back propagate

$$\nabla_\theta \mathscr{L}(y|\theta) \rightarrow \nabla_\theta \mathscr{L}(\omega,\theta)$$

$$q(\theta) = N(\overbrace{\mu,\sigma}^{\theta})$$

ω are samples of this distribution

$$\omega = \mu + \sigma \odot \epsilon\ , \quad q(\epsilon)\ , \quad q(\epsilon)d\epsilon = q(\omega|\theta)d\omega$$

Weight Uncertainty in Neural Networks,
Blundell et al. 1015(https://arxiv.org/pdf/1505.05424.pdf)

$$\frac{\partial}{\partial\theta}\mathbb{E}_{q(\omega|\theta)}[f(\omega,\theta)] = \mathbb{E}_{q(\epsilon)}\left[\frac{\partial f(\omega,\theta)}{\partial\omega}\frac{\partial\omega}{\partial\theta} + \frac{\partial f(\omega,\theta)}{\partial\theta}\right] \longrightarrow \begin{cases} \frac{\partial}{\partial\mu}\mathbb{E}_{q(\omega|\theta)}[\mathscr{L}(\omega,\theta(\mu,\sigma))] = \mathbb{E}_{q(\epsilon)}\left[\frac{\partial\mathscr{L}(\omega,\theta)}{\partial\theta}\frac{\partial\omega}{\partial\mu} + \frac{\partial\mathscr{L}(\omega,\theta)}{\partial\mu}\right] \\ \frac{\partial}{\partial\sigma}\mathbb{E}_{q(\omega|\theta)}[\mathscr{L}(\omega,\theta(\mu,\sigma))] = \mathbb{E}_{q(\epsilon)}\left[\frac{\partial\mathscr{L}(\omega,\theta)}{\partial\theta}\frac{\partial\omega}{\partial\sigma} + \frac{\partial\mathscr{L}(\omega,\theta)}{\partial\sigma}\right] \end{cases}$$

this is the same for both μ and σ

$$\frac{\partial}{\partial\theta}\mathscr{L}(y|\theta) \longrightarrow \nabla_\mu\mathscr{L}(\omega,\theta)\ ,\ \nabla_\sigma\mathscr{L}(\omega,\theta) \xrightarrow{\text{update de variational parameters}} \left\{\begin{aligned} \mu &= \mu - \alpha\nabla_\mu\mathscr{L} \\ \sigma &= \sigma - \alpha\nabla_\sigma\mathscr{L} \end{aligned}\right\} \longleftarrow \text{Your standard gradient descent}$$

Approximate Bayesian Inference: Flipout

The problem is that to make this computationally possible, we sample a single ϵ per mini-batch and, therefore, sharing the same weight perturbation.

$$\omega = \mu + \sigma \odot \epsilon$$

$\downarrow$

This introduces correlations between the gradients of each mini-batch

$\downarrow$

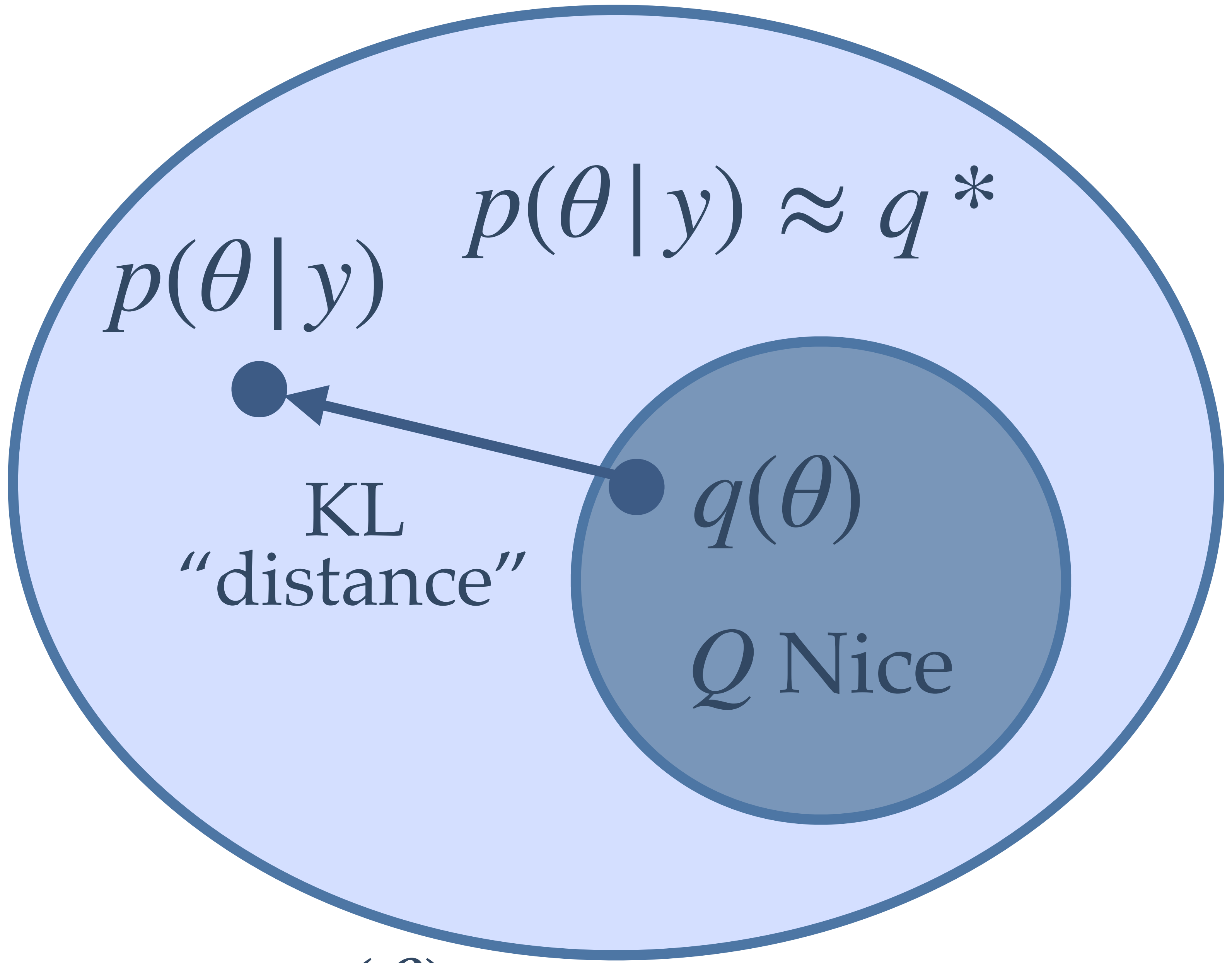
Flipout decorrelates them by simply adding a flip-the-coin effect:

$$q(\epsilon) \longrightarrow \epsilon \xrightarrow{\text{flip the coin}} + \text{ or } - \longrightarrow \pm\epsilon \longrightarrow \omega = \mu \pm \sigma \odot \epsilon$$

For more details see Flipout: Efficient Pseudo-Independent Weight Perturbations on Mini-Batches
Wen et al. (https://arxiv.org/pdf/1803.04386.pdf)

Approximate Bayesian Inference: Variational Inference

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



$$q*(\theta) = argmin_{q\in Q} KL(q(\cdot))||p(\cdot|y)$$

Kullback-Leibler divergence:

$$KL(q(\cdot))||p(\cdot|y) := \int q(\theta)\log\frac{q(\theta)}{p(\theta|y)}d\theta$$

$$= \int q(\theta)\log\frac{q(\theta)p(y)}{p(\theta,y)}d\theta = \int q(\theta)\left[\log p(y) + \log\frac{q(\theta)}{p(\theta,y)}\right]d\theta$$

Evidence lower bound (ELBO)

$$= \log p(y)\int q(\theta)d\theta - \int q(\theta)\log\frac{p(\theta,y)}{q(\theta)}d\theta = \log p(y) - \int q(\theta)\log\frac{p(\theta,y)}{q(\theta)}d\theta$$

one can prove (check Bishop)

$$KL \geq 0 \implies \log p(y) \geq \text{ELBO} \rightarrow q* = argmax_{q\in Q} ELBO(q) = argmax_{q\in Q}\int q(\theta)\log\frac{p(\theta,y)}{q(\theta)}d\theta$$

Approximate Bayesian Inference: Mean Field Variational Bayes

$$q^* = argmax_{q \in Q} \underbrace{\int q(\theta) \log \frac{p(\theta, y)}{q(\theta)} d\theta}_{\text{ELBO}}$$

$$q(\theta) = \prod_i^m q_i(\theta_i) \longleftarrow$$

We assume q factorizes with respect to θs

$$\text{ELBO} = \int q(\theta) \log \frac{p(\theta, y)}{q(\theta)} d\theta = \int q(\theta) \log p(\theta, y) d\theta - \int q(\theta) \log q(\theta) d\theta$$

$q(\theta) \approx \prod_i q_i(\theta_i)$

$$= \int \prod_i q_i(\theta_i) \log \frac{p(\theta, y)}{\prod_i q_i(\theta_i)} d\theta = \int \prod_i q_i(\theta_i) \left[\log p(\theta, y) - \sum_i \log q_i(\theta_i)\right] d\theta$$

choose one q_j

$$= \int q_j(\theta_j) \left[\int \log p(\theta, y) \prod_{i \neq j} q_i(\theta_i) d\theta_i\right] d\theta_j - \int q_j(\theta_j) \log q_j(\theta_j) \underbrace{\left[\int \prod_{i \neq j} q_i(\theta_i) d\theta_i\right]}_{=1} d\theta_j - \underbrace{\int \prod_{i \neq j} q_i(\theta_i) \log q_i(\theta_i) d\theta_i}_{= \text{const.}}$$

$$= \int q_j(\theta_j) \log \tilde{p}(\theta_j, y) d\theta_j - \int q_j(\theta_j) \log q_j(\theta_j) d\theta_j + \text{const.}$$

$$= \int q_j(\theta_j) \log \frac{\tilde{p}(\theta_j, y)}{q_j(\theta_j)} d\theta_j + \text{const.} = -KL(q_j(\theta_j)) || \tilde{p}(\theta_j, y)) + \text{const.}$$

Approximate Bayesian Inference: Mean Field Variational Bayes

Optimize for q_j :

$$\text{ELBO} = -KL(q_j(\theta_j))\,||\,\tilde{p}(\theta_j, y)) + \text{const.} = -\int q_j(\theta_j)\log\frac{q_j(\theta_j)}{\tilde{p}(\theta_j, y_j)}d\theta_j + \text{const.}$$

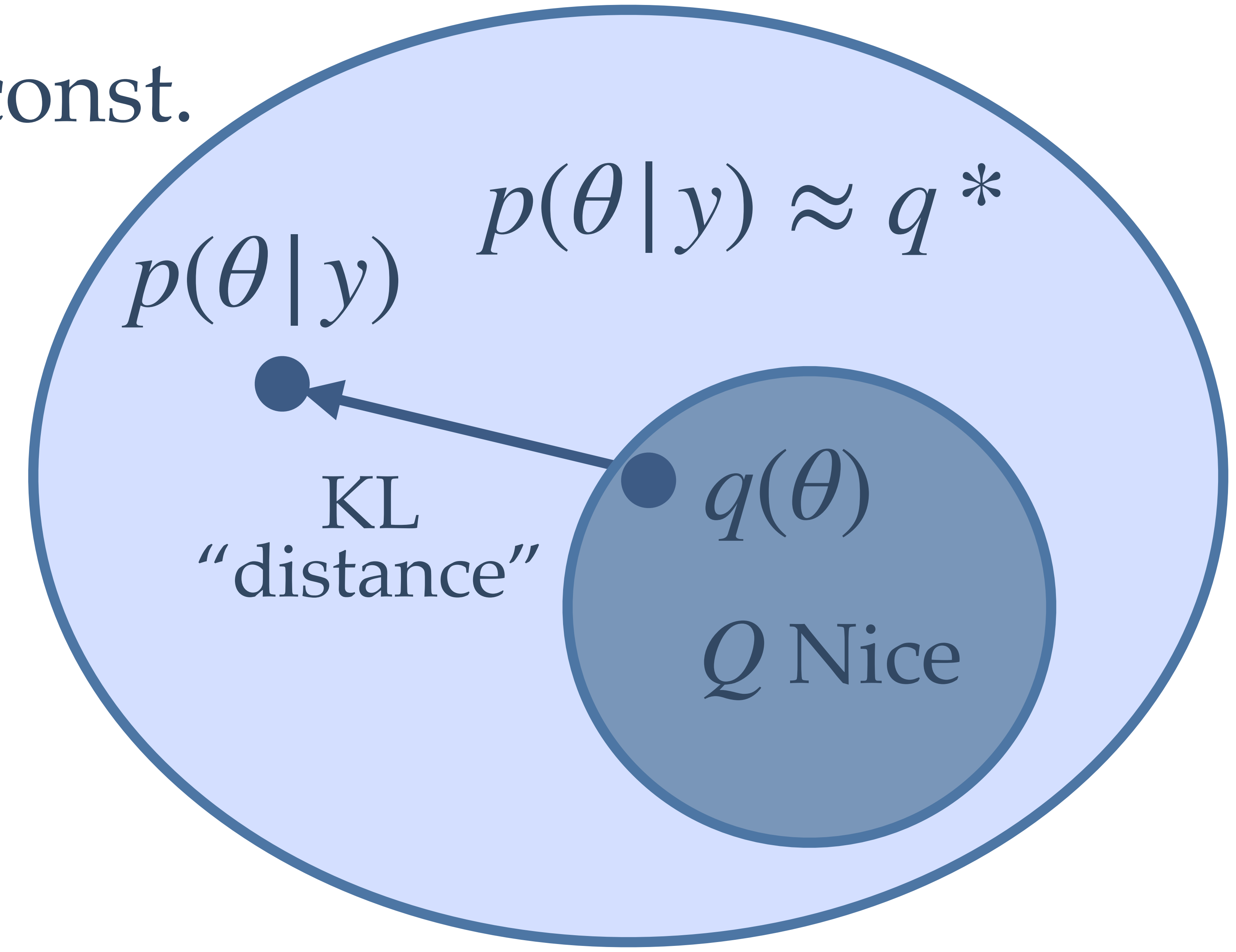
Maximizing the ELBO is equivalent to minimizing KL:

$$q^*_j = argmin_{q(\theta_j)} KL(q_j(\theta_j))\,||\,\tilde{p}(\theta_j, y)) \quad \Longrightarrow \quad q_j(\theta_j) = \tilde{p}(\theta_j, y)$$

$$\log q^*_j(\theta_j) = \log \tilde{p}(\theta_j, y_j) = \mathbb{E}_{i\neq j}[\log p(\theta, y)] + \text{const.}$$

set by normalizing $q^*_j(\theta_j)$

$$q^*_j(\theta_j) = \frac{\exp(\mathbb{E}_{i\neq j}[\log p(\theta, y)])}{\int \exp(\mathbb{E}_{i\neq j}[\log p(\theta, y)])d\theta_j}$$



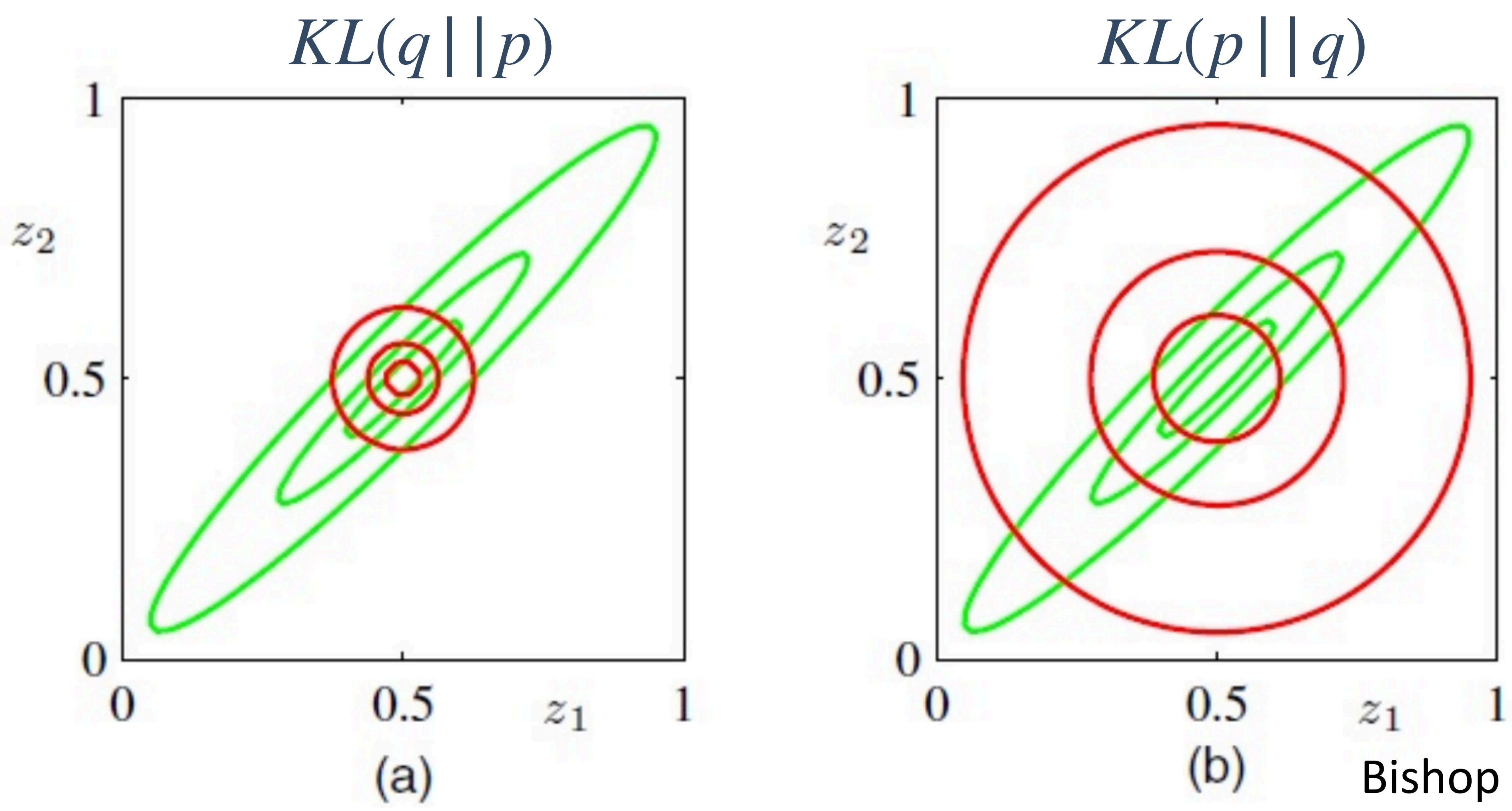
This represents the conditions for the maximum of the ELBO, given the factorization assumption. The problem is that the expression for $q^*_j(\theta_j)$ depends on expectations of all other $q_i(\theta_i)$. We need to iterate through them. Convergence is guaranteed because ELBO is convex with respect to each factor $q_i(\theta_i)$

> for $j \in \{1,....,m\}$ do   until ELBO has converged
>
> $$q_j(\theta_j) \propto \exp\{\mathbb{E}_{i\neq j}[\log(p(\theta_j|\theta_{i\neq j}, y)]\}$$
>
> compute ELBO(q)

- Coordinate descent
- Stochastic variational inference (SVI) [Hoffman et al 2013]
- Automatic differentiation variational inference (ADVI) [Kucukelbir et al 2015, 2017]

Approximate Bayesian Inference: Mean Field Variational Bayes

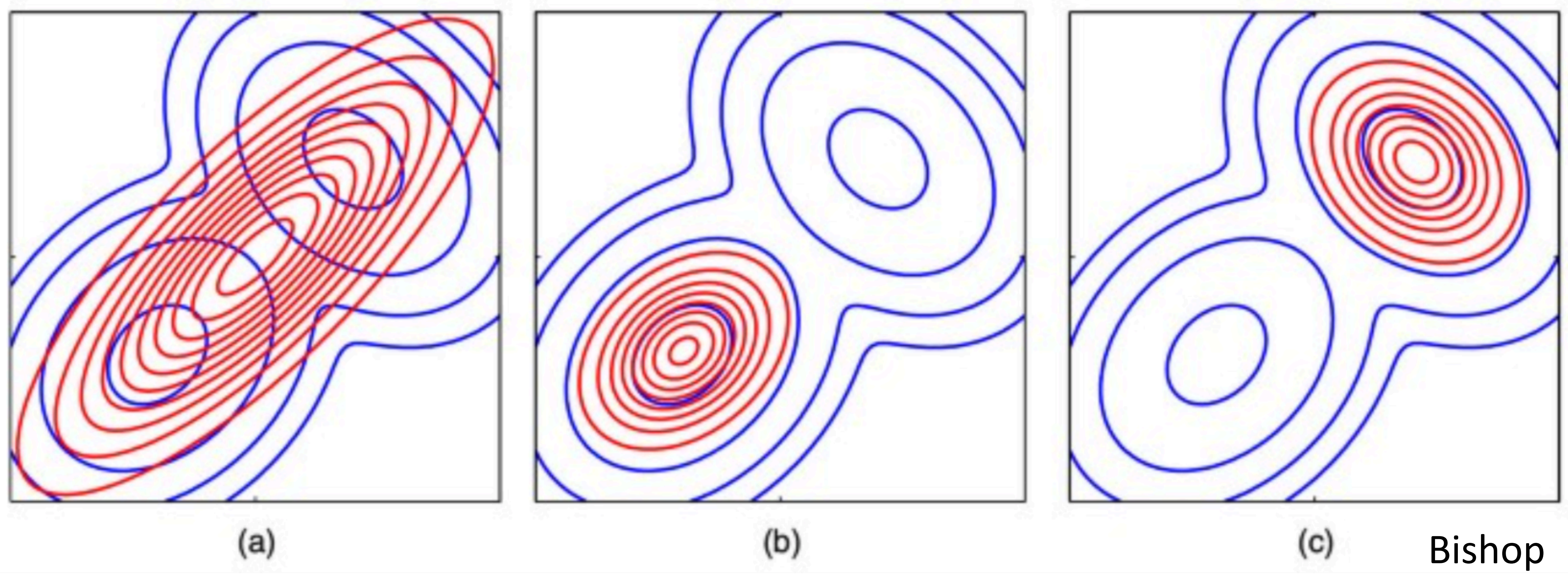
Green:
1, 2, and 3 standard deviations for a correlated Gaussian distribution



Red:
Same levels of an approximate distribution given by the product of two independent univariate Gaussians obtained by:

a) minimizing $KL(q||p)$ divergence
b) minimizing $KL(p||q)$ divergence

Blue:
Bimodal distribution given by a mixture of two Gaussians



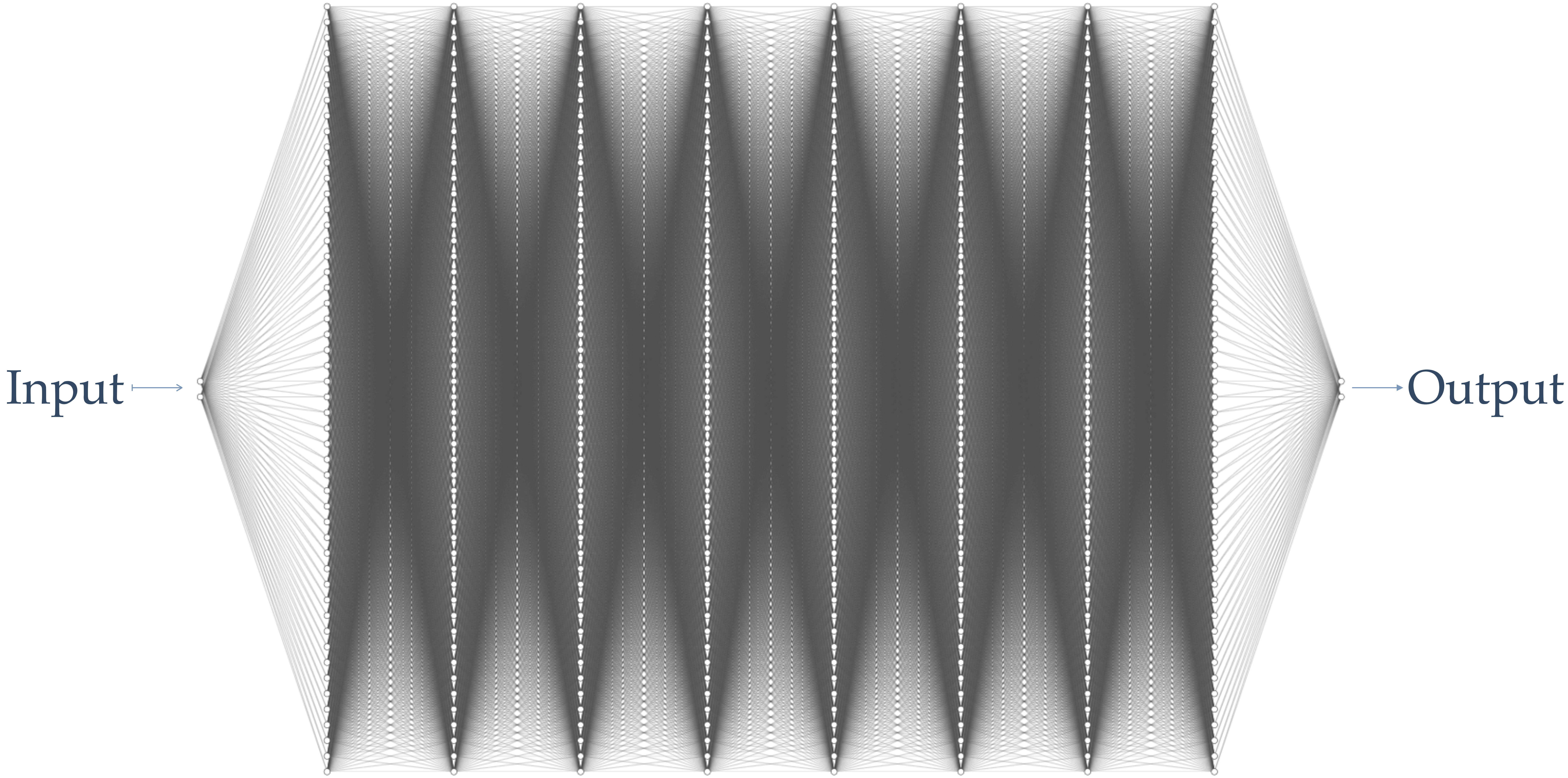
Red:

a) Best approximation by a single Gaussian by minimizing the KL divergence
b) same as (a) but numerically minimizing KL divergence
c) same as in (b) but another local minimum of the KL divergence

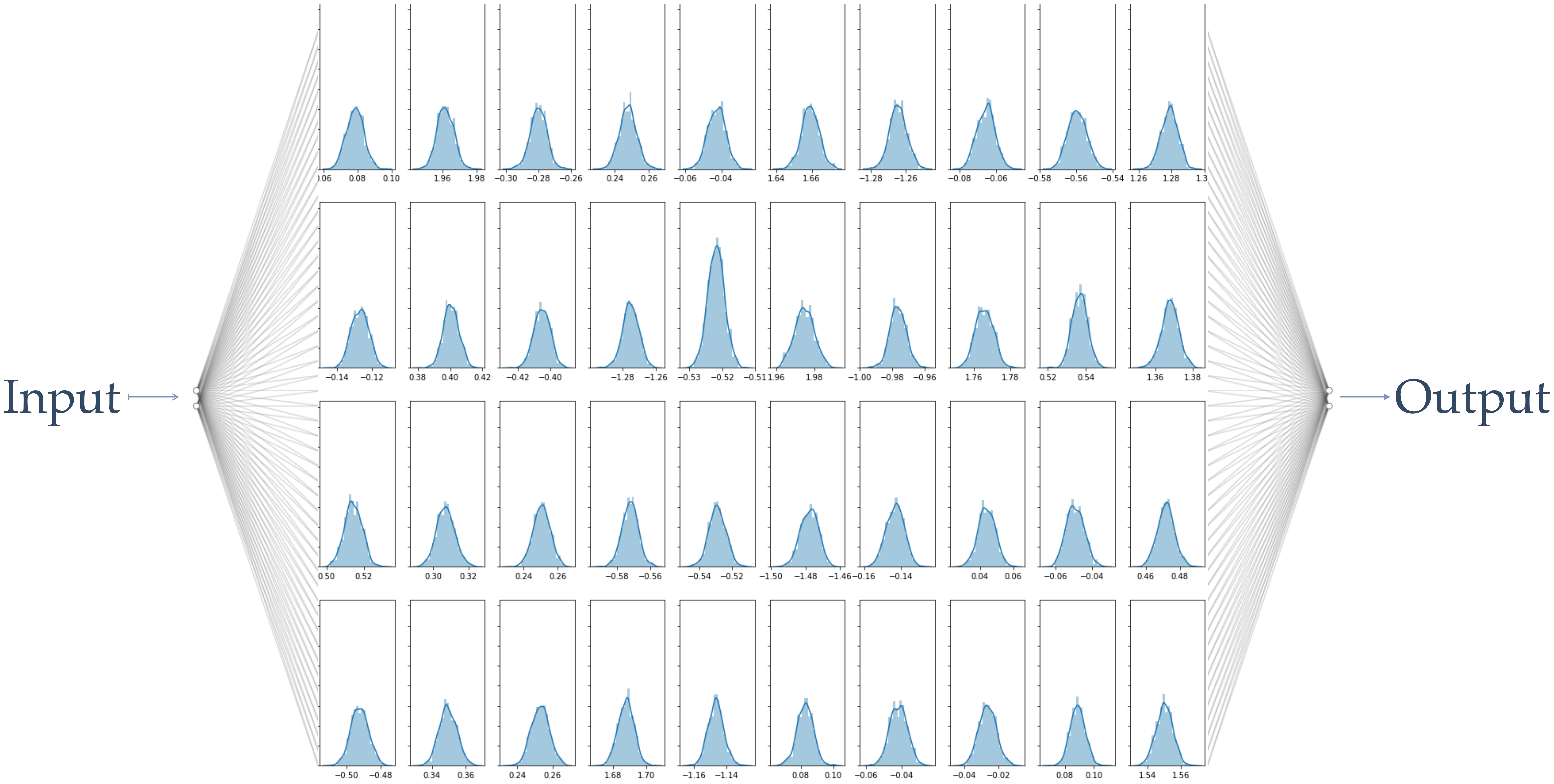
So far

1. Bayesian Inference ✓
2. Markov Chain Monte Carlo ✓
3. Variational Inference ✓
4. Bayesian Neural Networks ✓
5. Bayes by backprop and flipout ✓
6. Mean Field Variational Bayes ✓
7. Application to Neural Networks
8. Drop Out as a Bayesian Approximation
9. Bootstrap for Inference
10. Measure performance of methods on uncertainty quantification

Bayesian Inference for Neural Networks



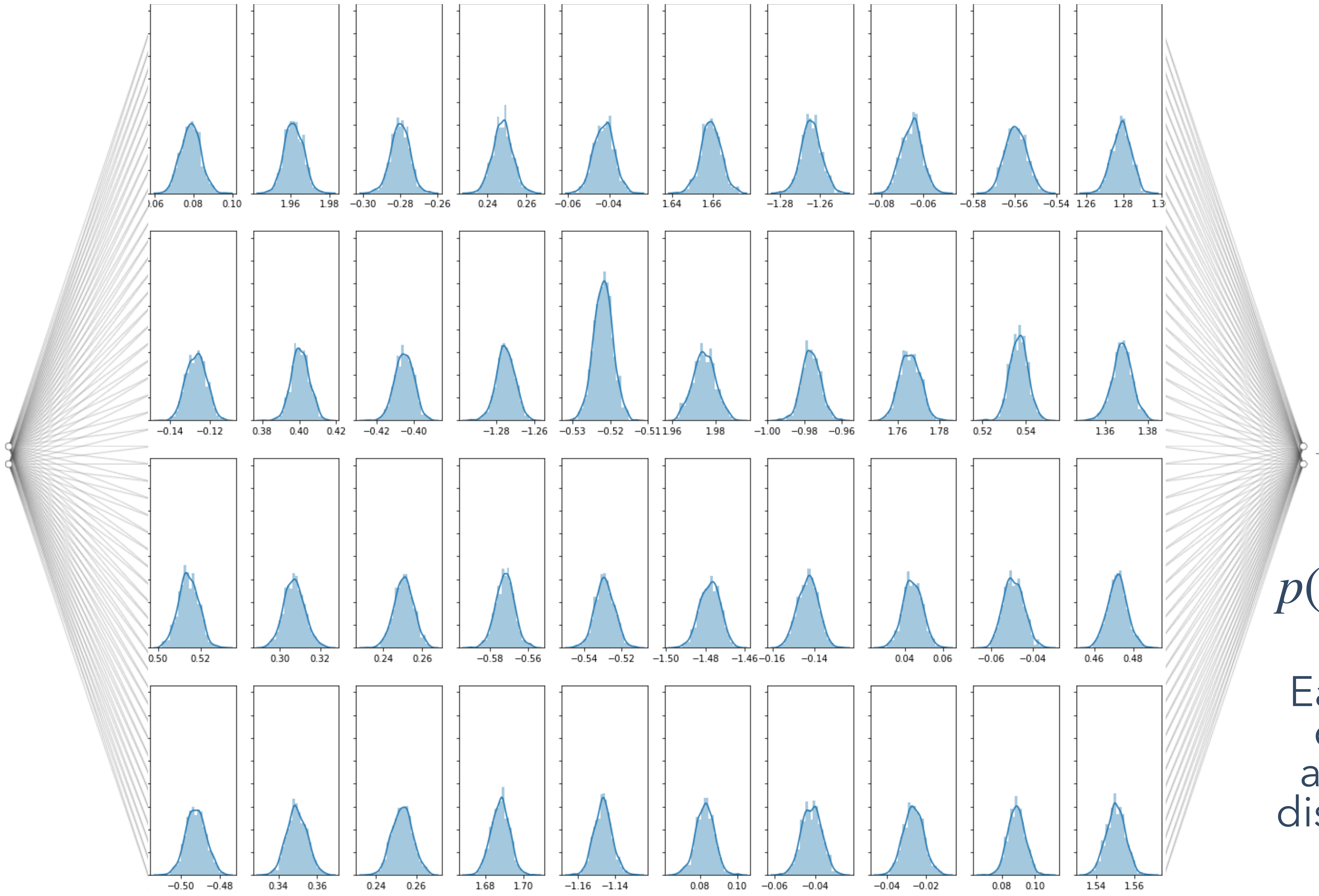
Bayesian Inference for Neural Networks



Bayesian Inference for Neural Networks

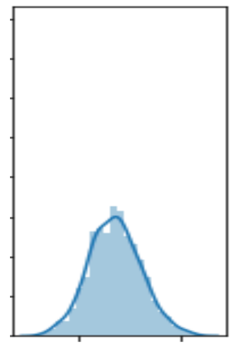
Input

$\hat{x}$



$p(\omega \,|\, y)$

Output



$\hat{y}$

$$p(\hat{y} \,|\, \hat{x}) = \mathbb{E}_{p(\omega|y)}[p(\hat{y} \,|\, \hat{x}, \omega)]$$

Each possible configuration of the weights, weighted according to the posterior distribution - infinite number of NNs

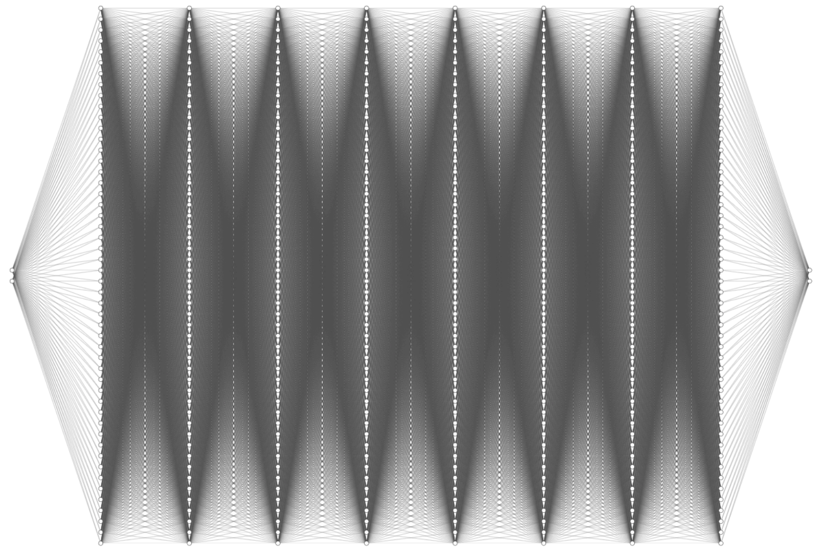
BI for NN calculates the posterior distribution of the weights given the training data

Bayesian Neural Networks

Priors $p(\theta)$ $\longrightarrow$ $\dfrac{p(\theta)\,p(y\,|\,\theta)}{p(y)_{\text{intractable}}} = p(\theta\,|\,y)$

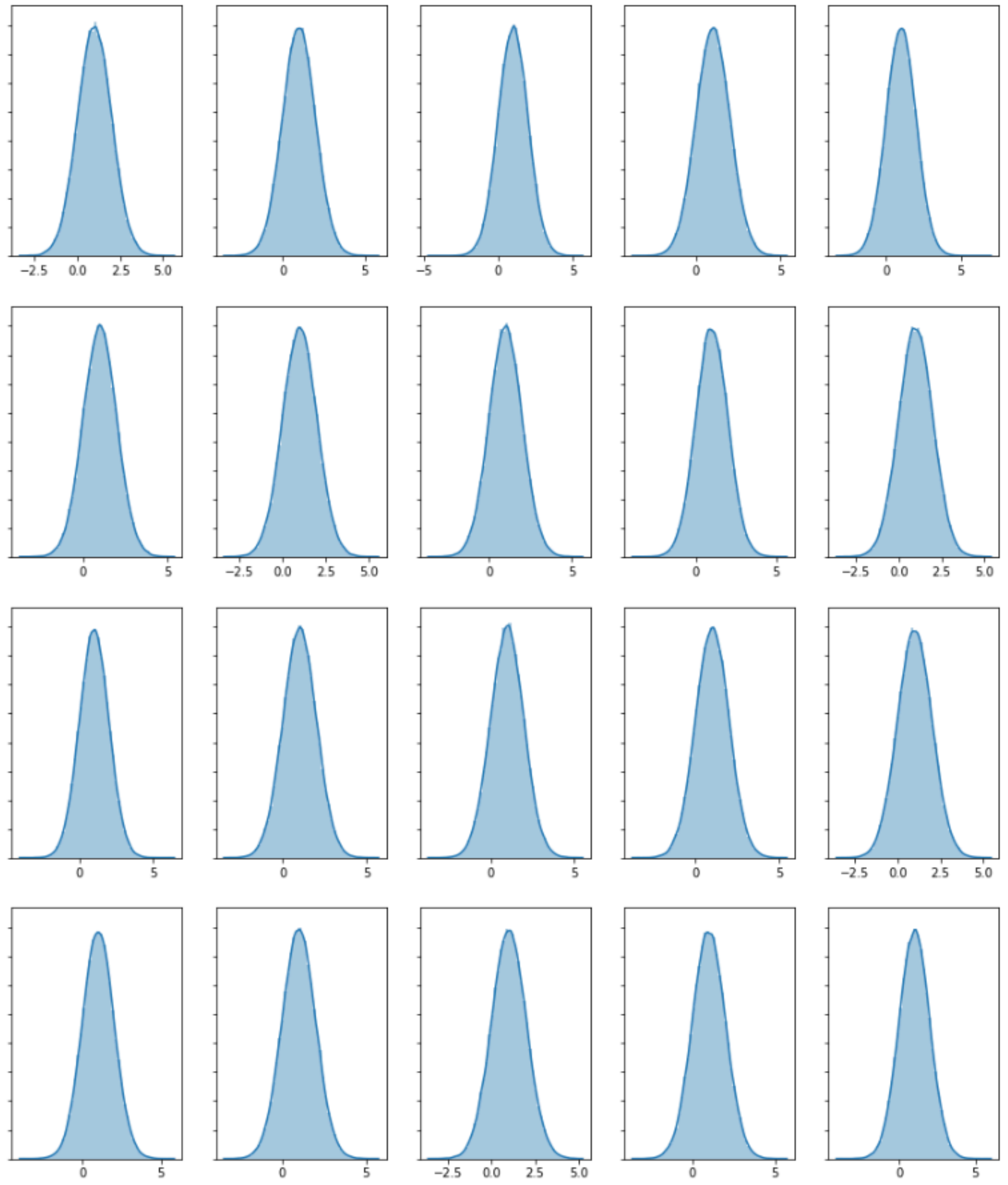
Means

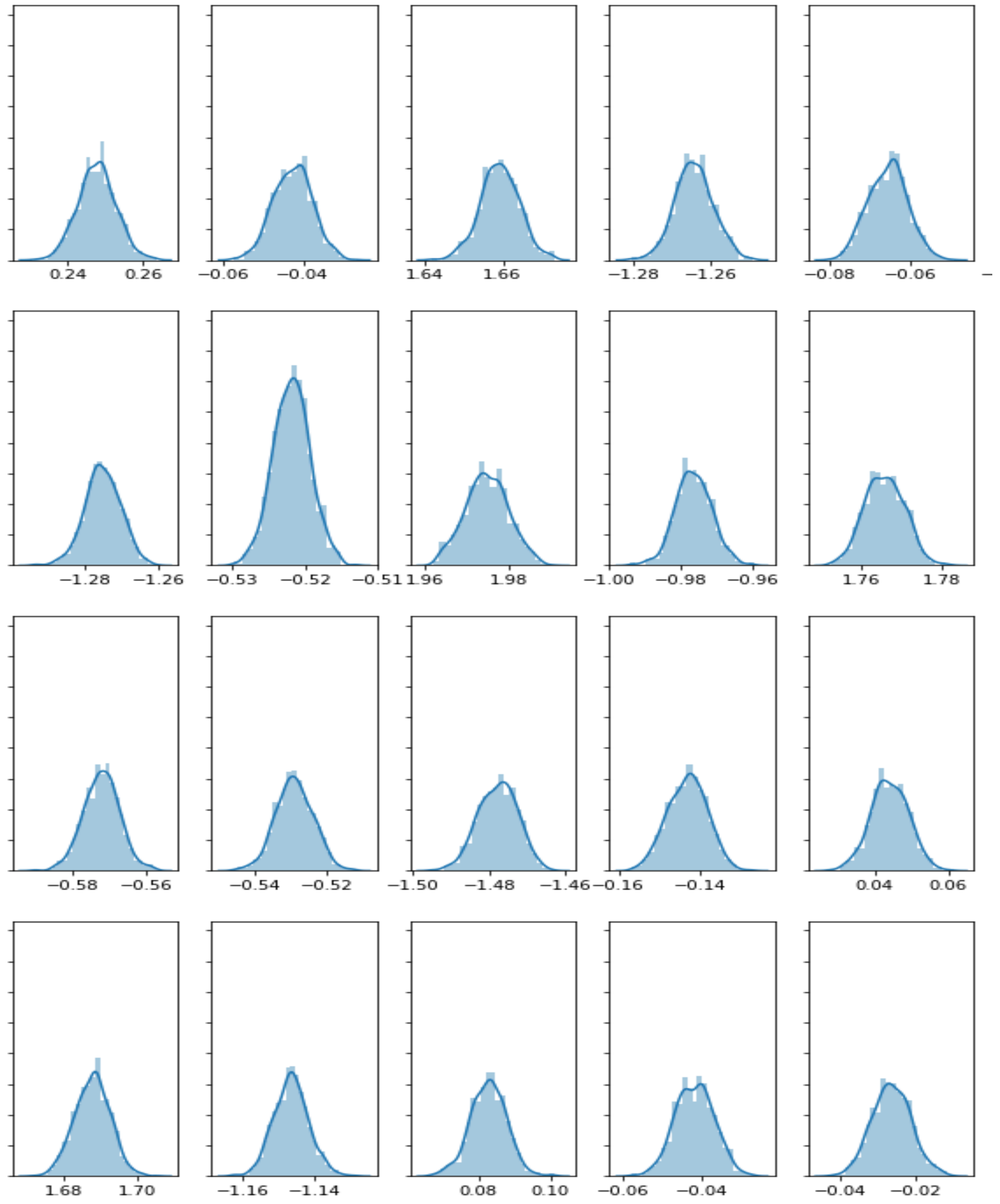
FCNN



&

Scale





Bayesian Neural Networks

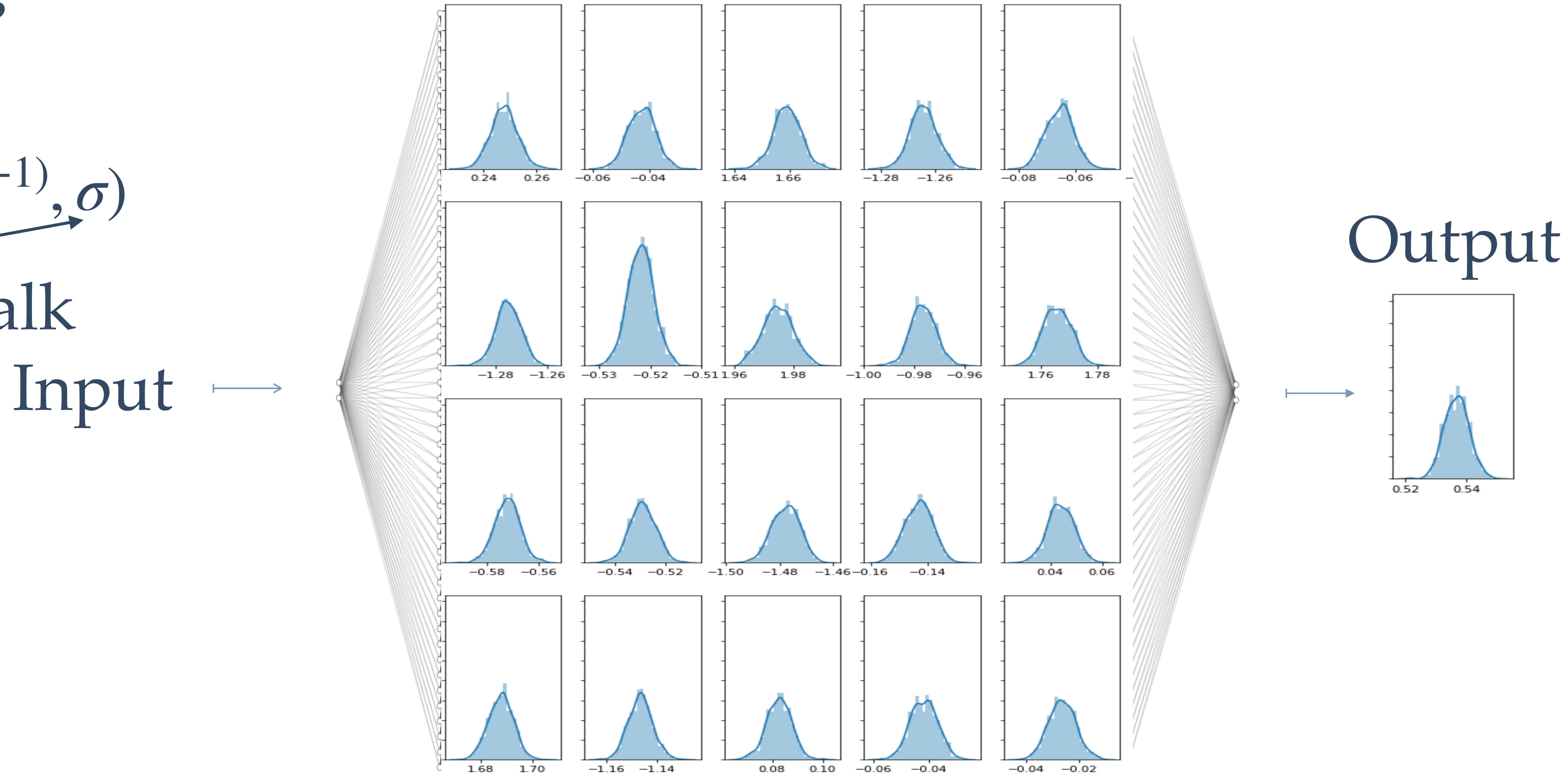
MCMC: Eventually accurate $p(\theta) \times p(y|\theta) \propto p(\theta|y)$

Only forward passes

$$\theta = (\mu, \sigma)$$

$$\mu^j \sim N(\mu^{(j-1)}, \sigma)$$

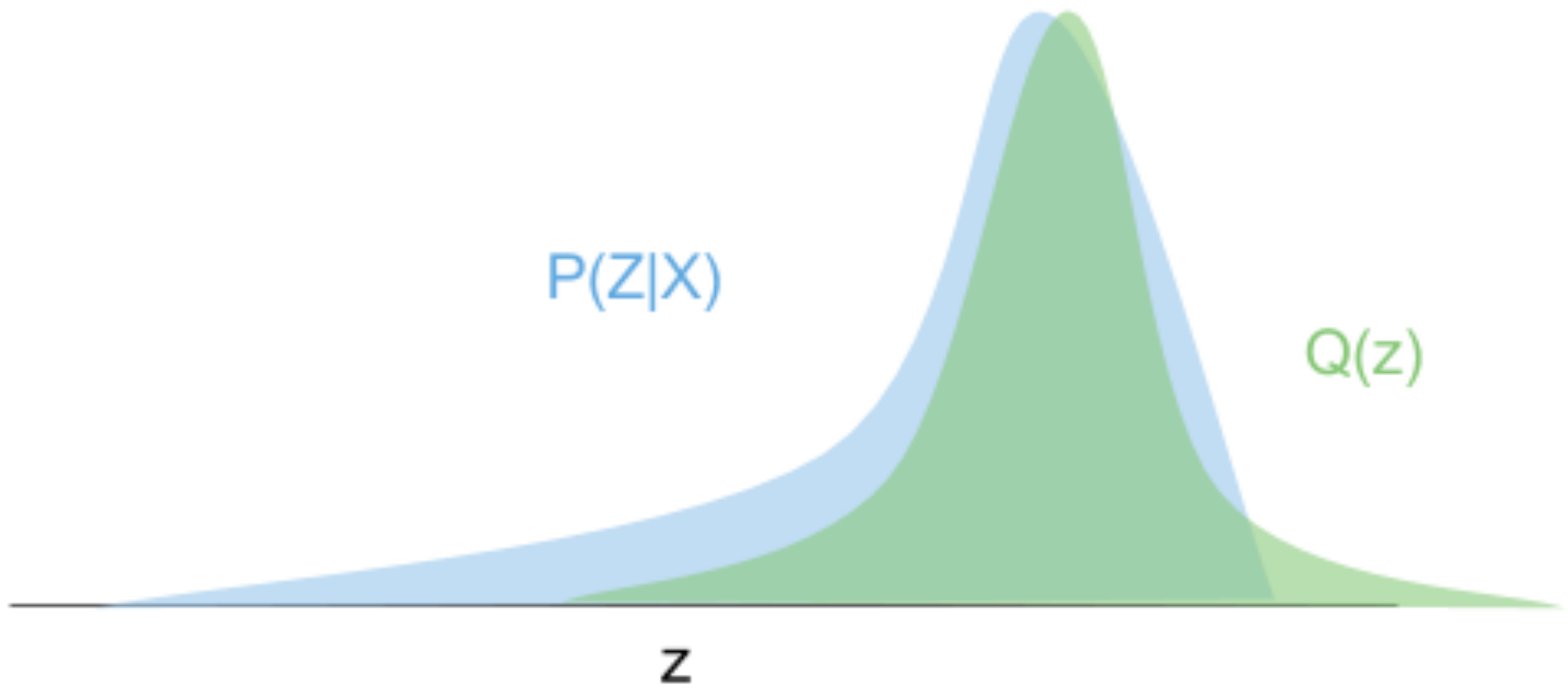
Step of random walk



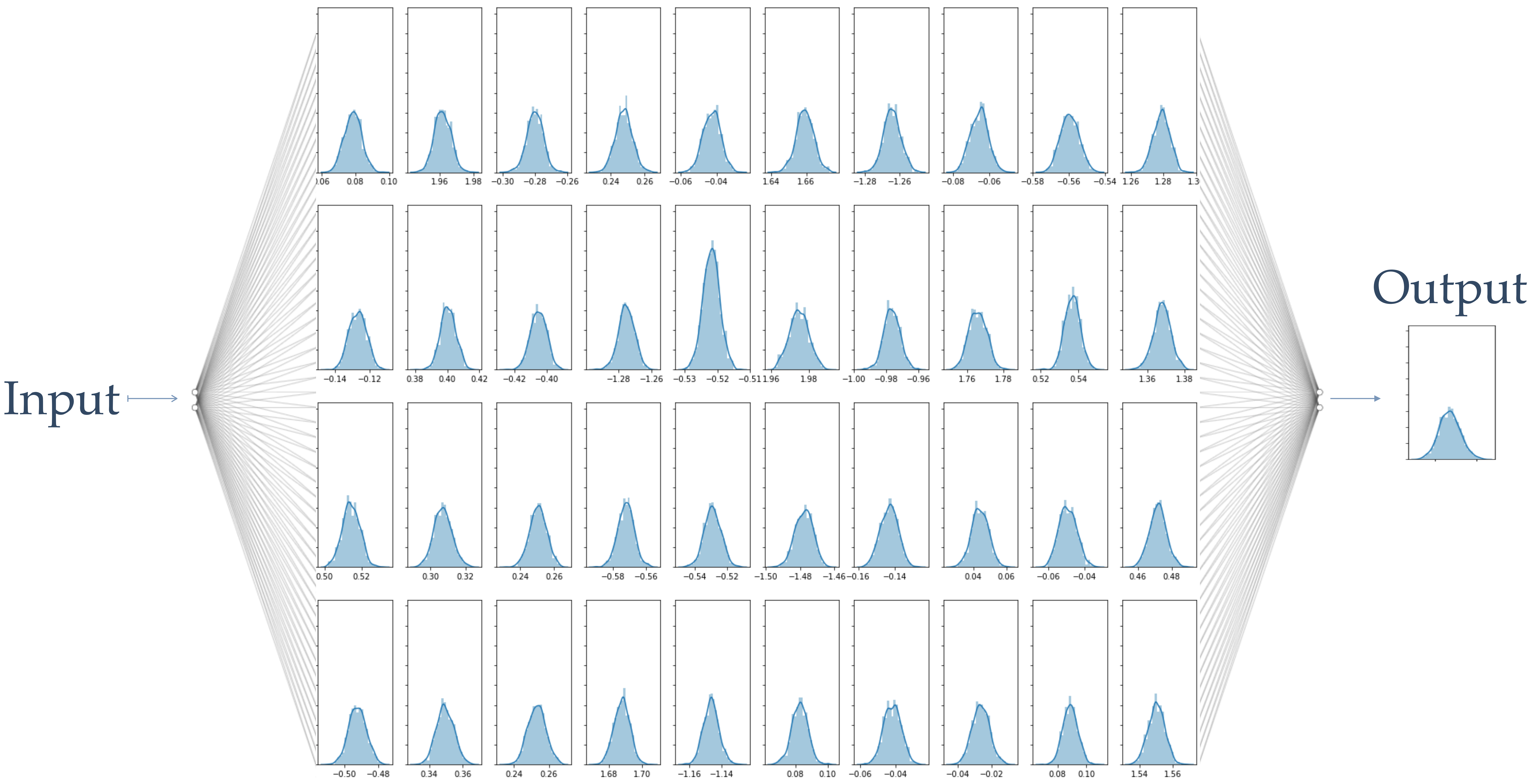
Approximate Bayesian Inference: Variational Bayes

$$p(\theta \mid y) = \frac{p(y \mid \theta)\, p(\theta)}{\int p(y, \theta) d\theta} \approx q^*$$

Optimization approach ➡ Q a family of "nice" distributions

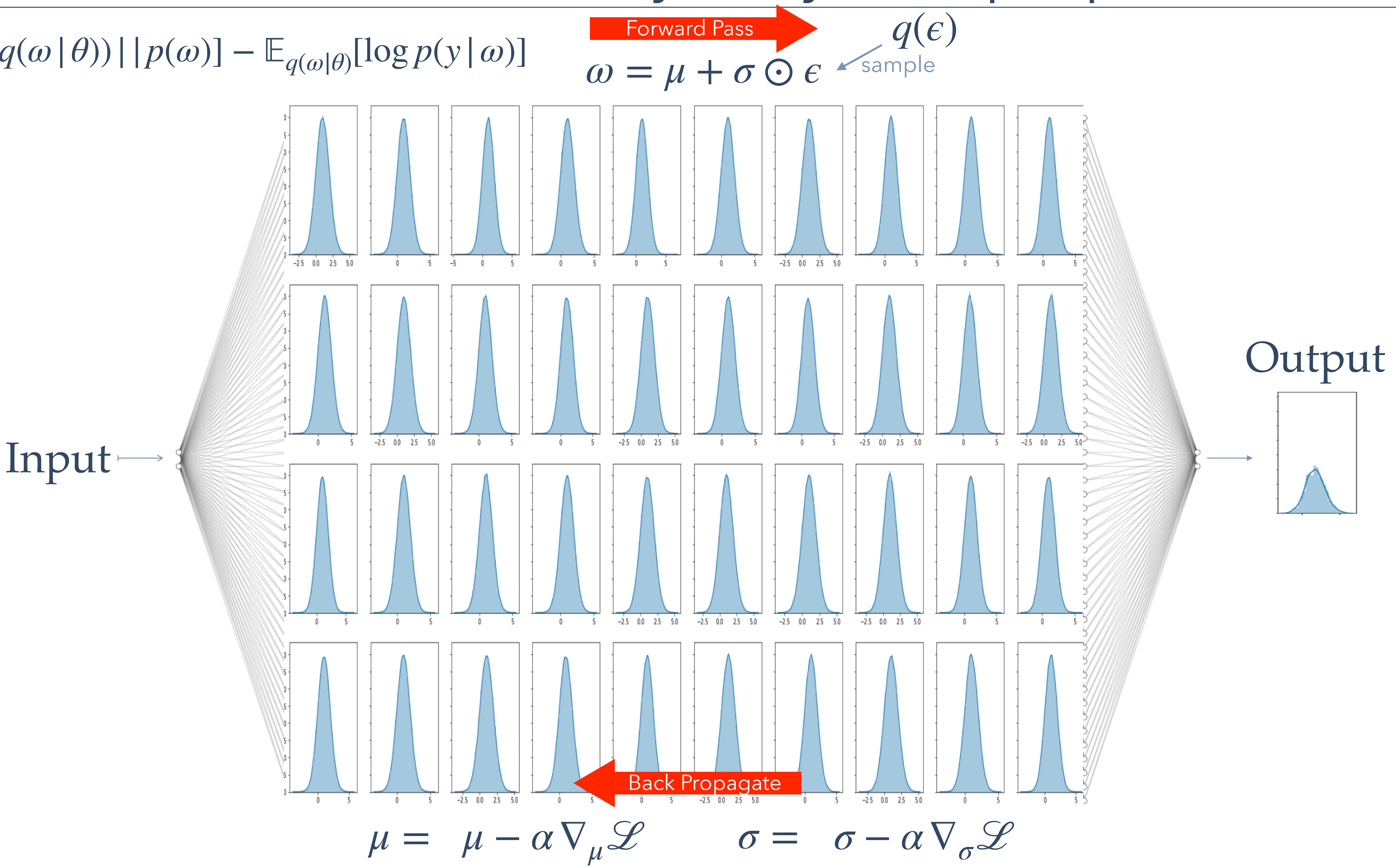


Bayesian Neural Network



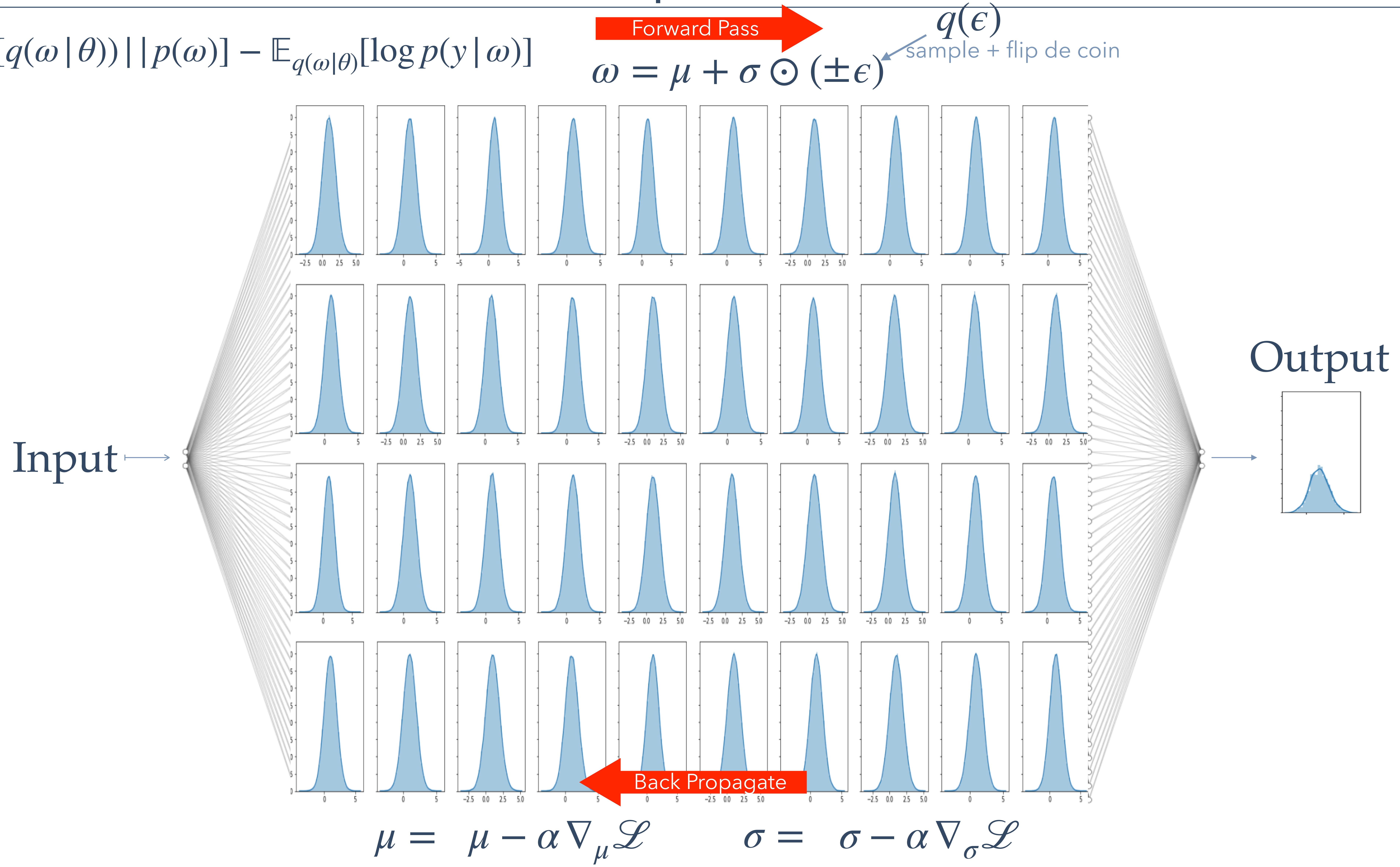
Bayesian Neural Network: Bayes by Backprop

$$\mathscr{L}(y\,|\,\theta) = KL[q(\omega\,|\,\theta)\,||\,p(\omega)] - \mathbb{E}_{q(\omega|\theta)}[\log p(y\,|\,\omega)]$$



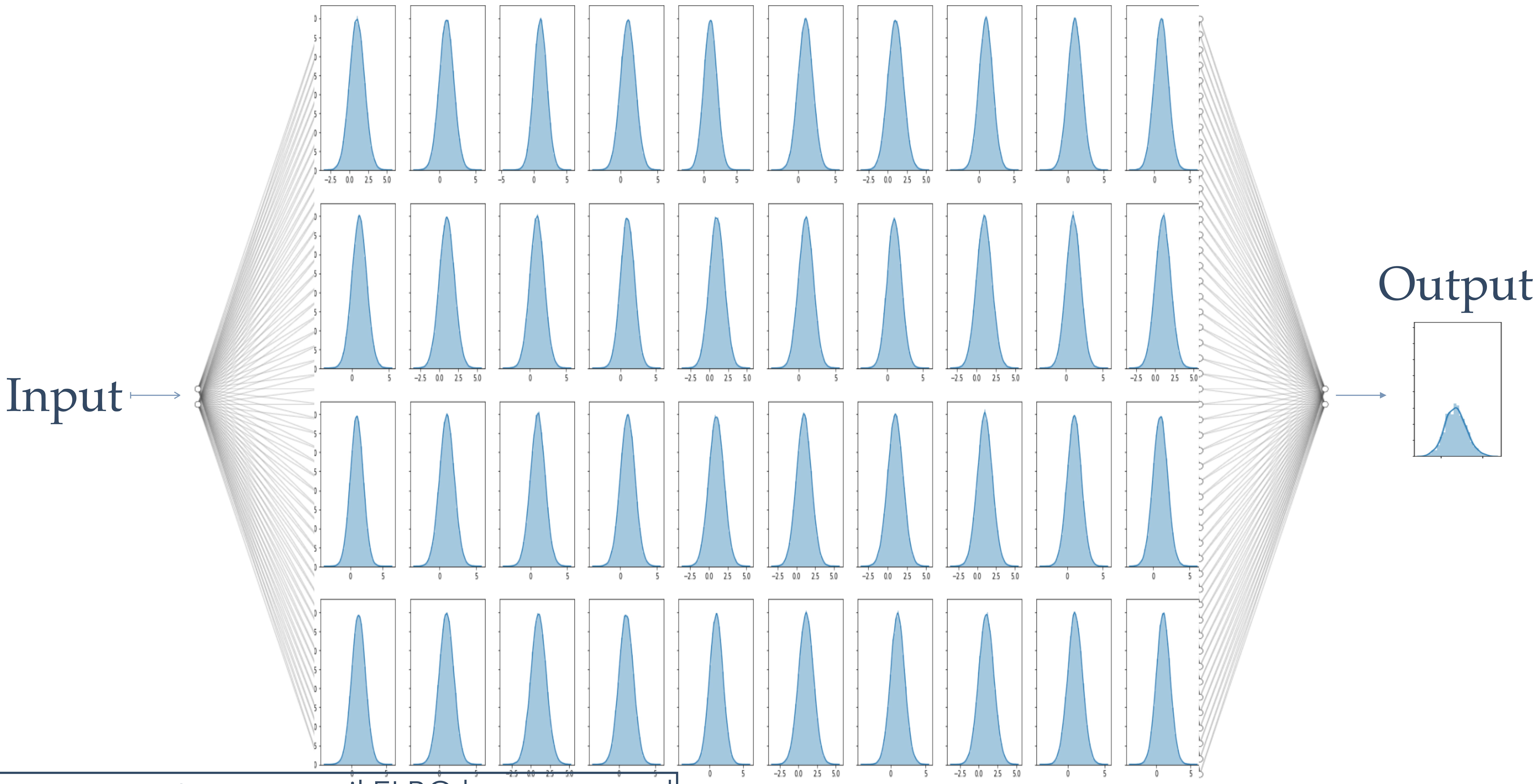
Bayesian Neural Network: Flipout

$$\mathscr{L}(y|\theta) = KL[q(\omega|\theta)||p(\omega)] - \mathbb{E}_{q(\omega|\theta)}[\log p(y|\omega)]$$



Bayesian Neural Network: MFVB

$$p(\theta|y) \approx q^* = argmax_{q \in Q} ELBO$$



for $j \in \{1,....,m\}$ do

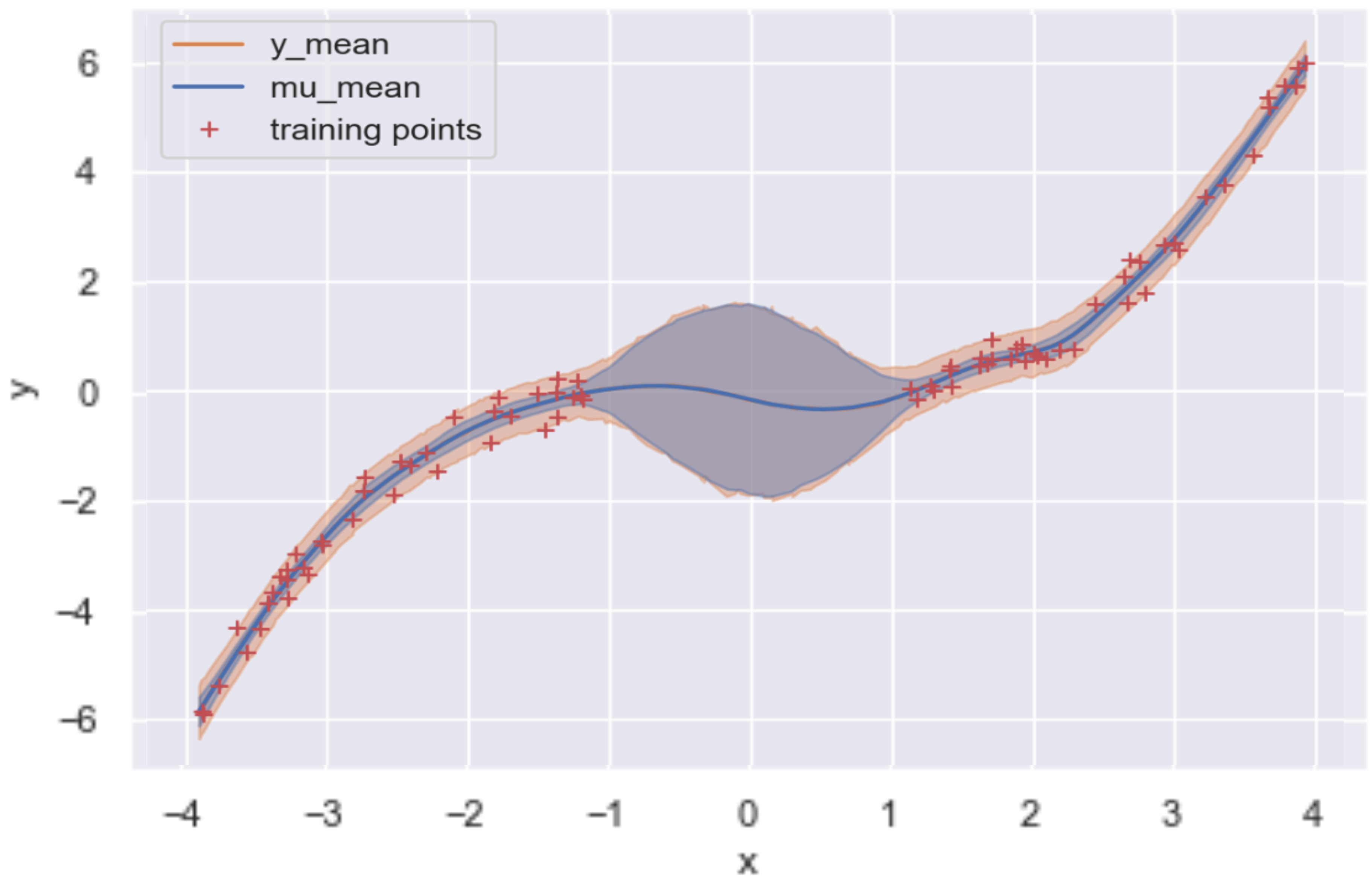
$$q_j(\theta_j) \propto \exp\{\mathbb{E}_{i \neq j}[\log(p(\theta_j | \theta_{i \neq j}, y)]\}$$

compute ELBO(q)

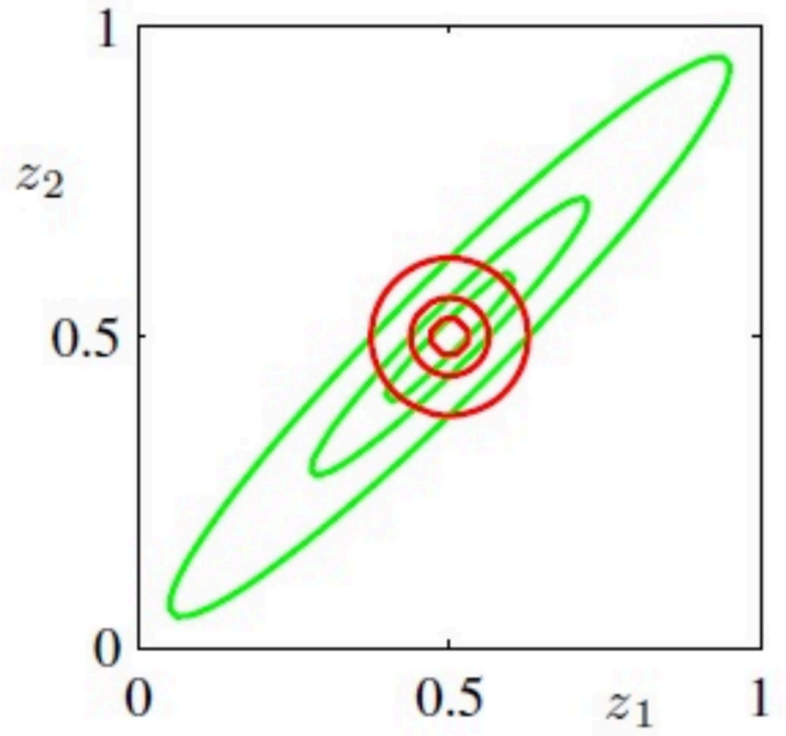
until ELBO has converged

Variational Bayesian Inference: The problem

Variational Bayesian Inference: The right solution (MCMC)

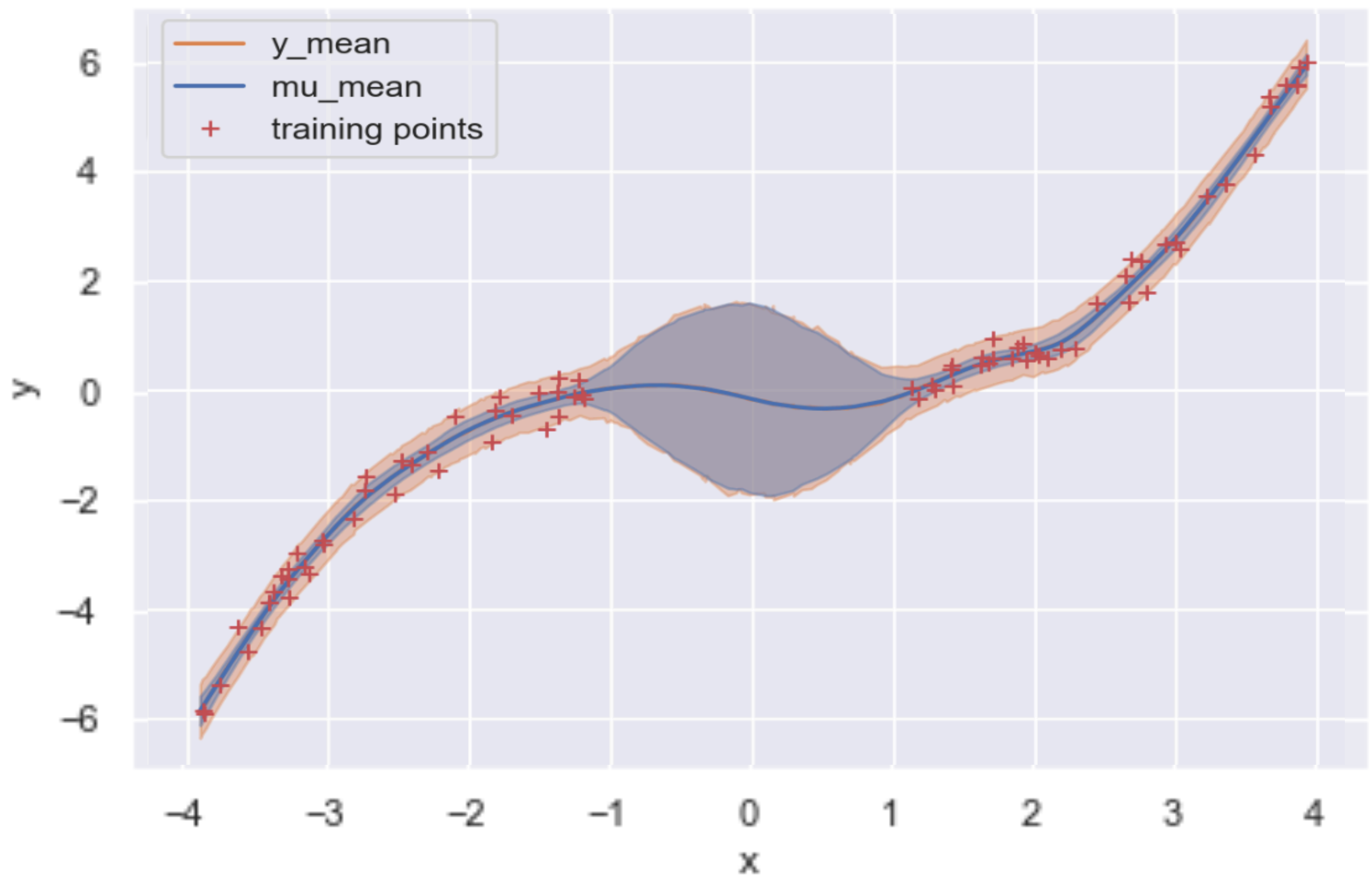


Variational Bayesian Inference

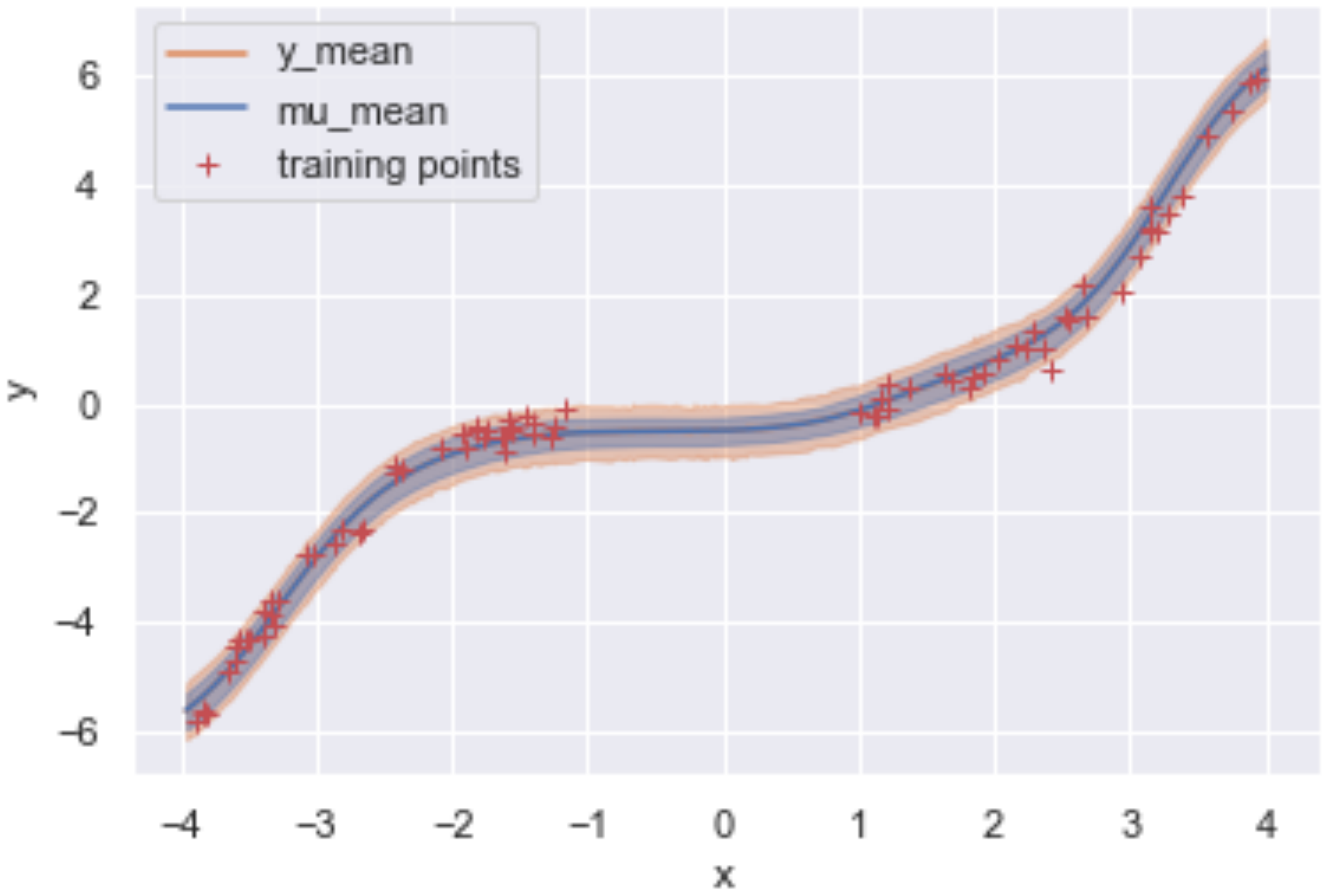


(a)

MCMC



MFVB

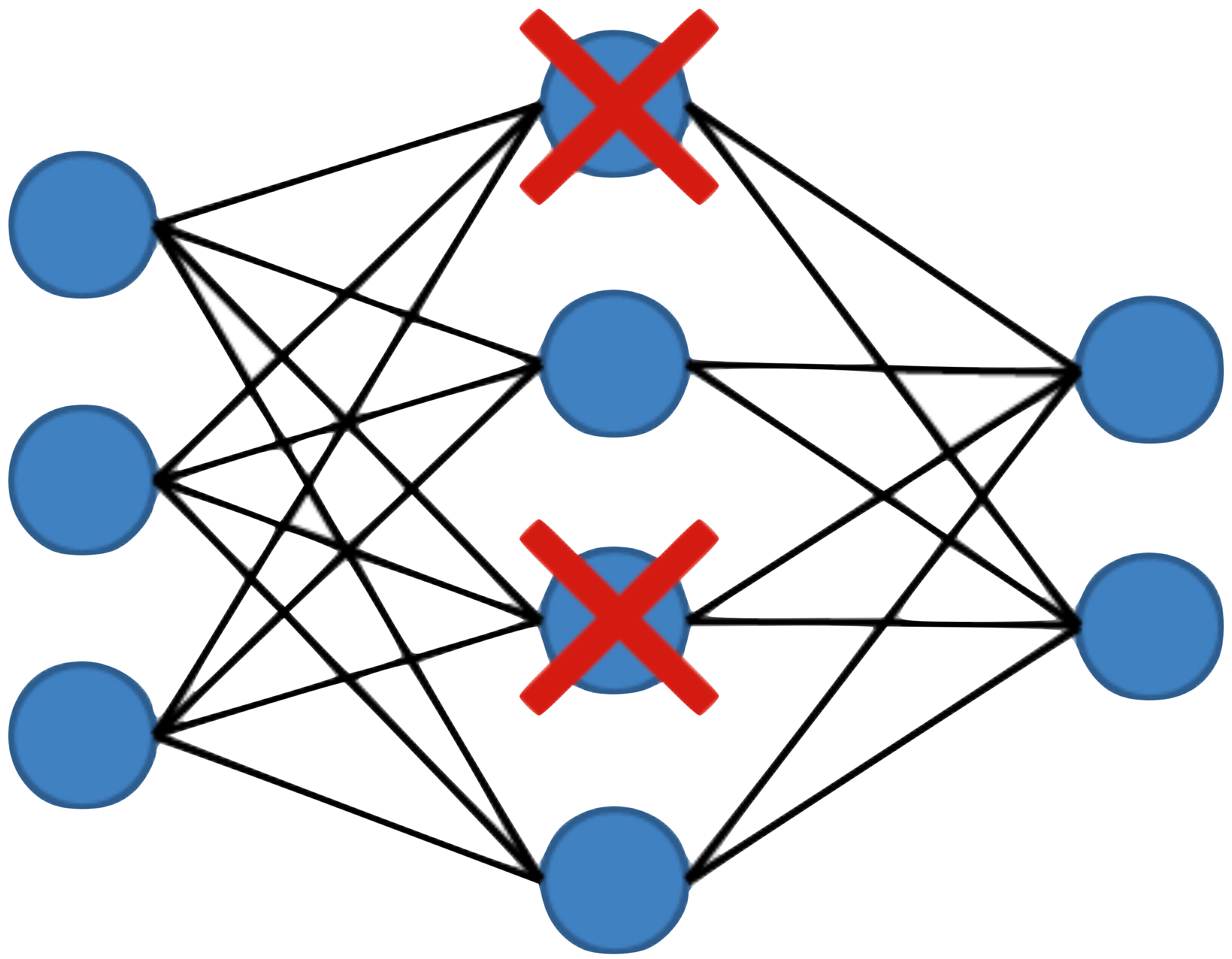


Dropout

arXiv:1506.02142

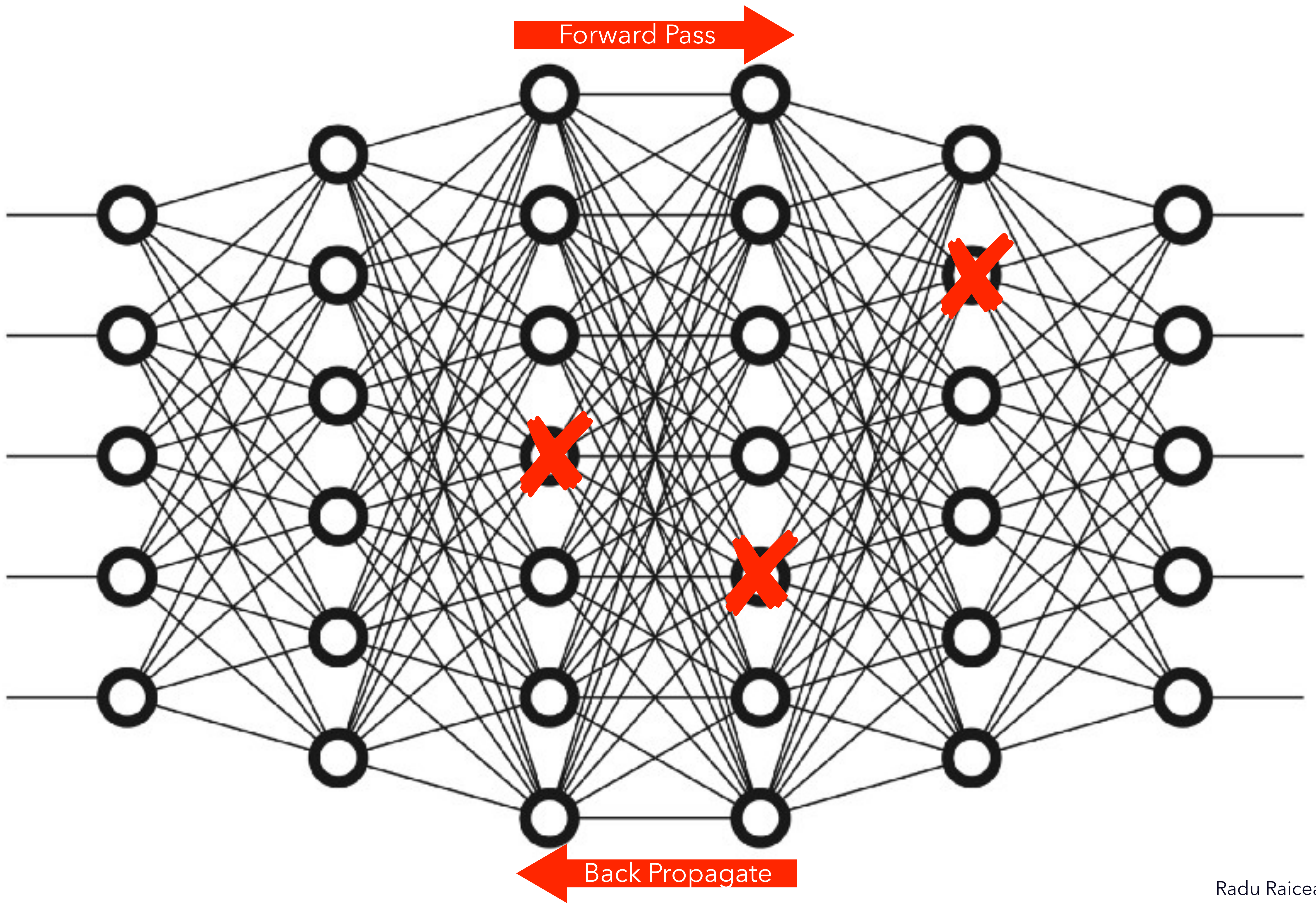
Dropout as a Bayesian Approximation:
Representing Model Uncertainty in Deep Learning

Yarin Gal YG279@CAM.AC.UK
Zoubin Ghahramani ZG201@CAM.AC.UK
University of Cambridge



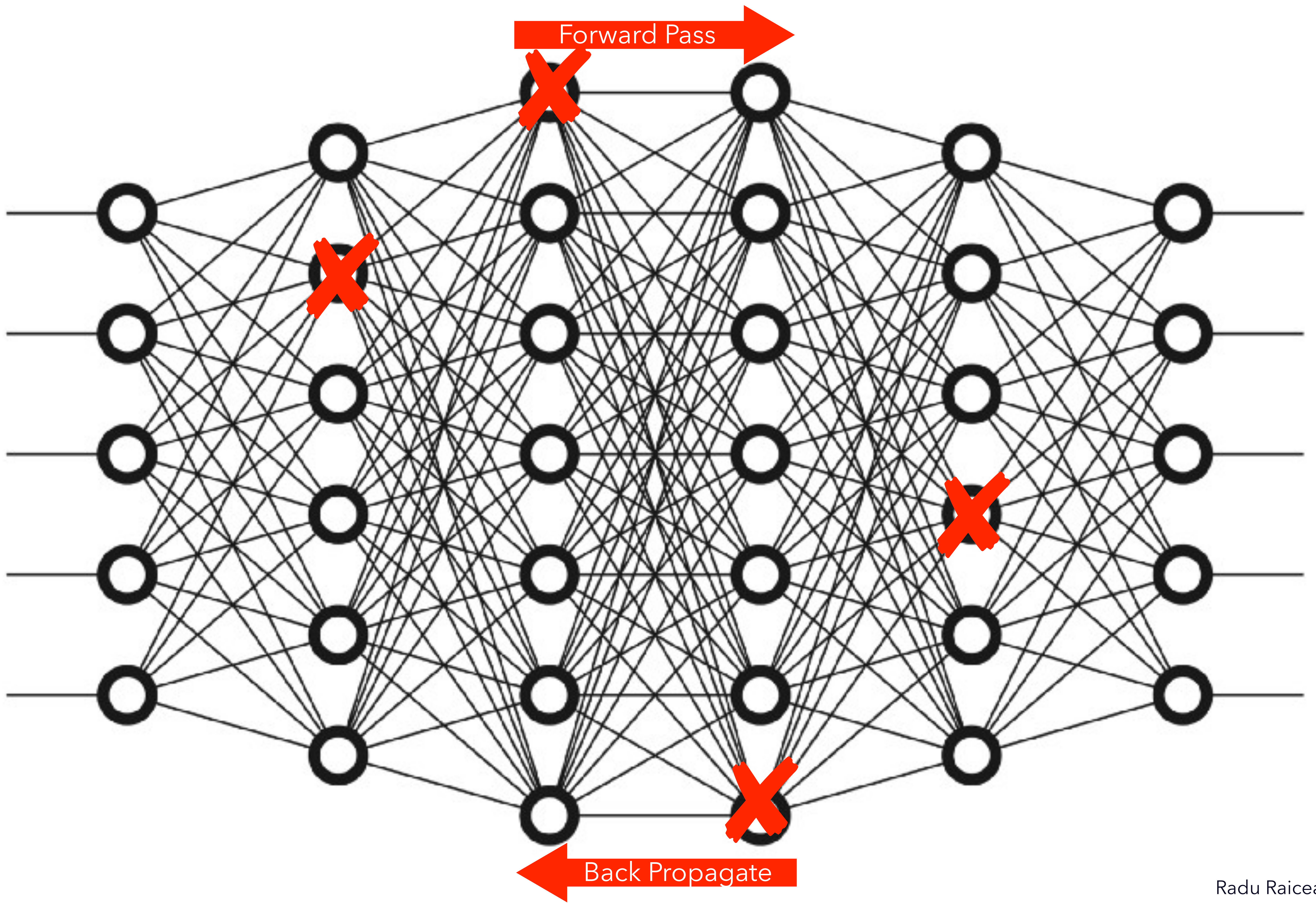
They show that a NN with arbitrary depth and non-linearities, with dropout applied before every weight layer, is mathematically equivalent to an approximation to the probabilistic deep Gaussian process.

Dropout: train



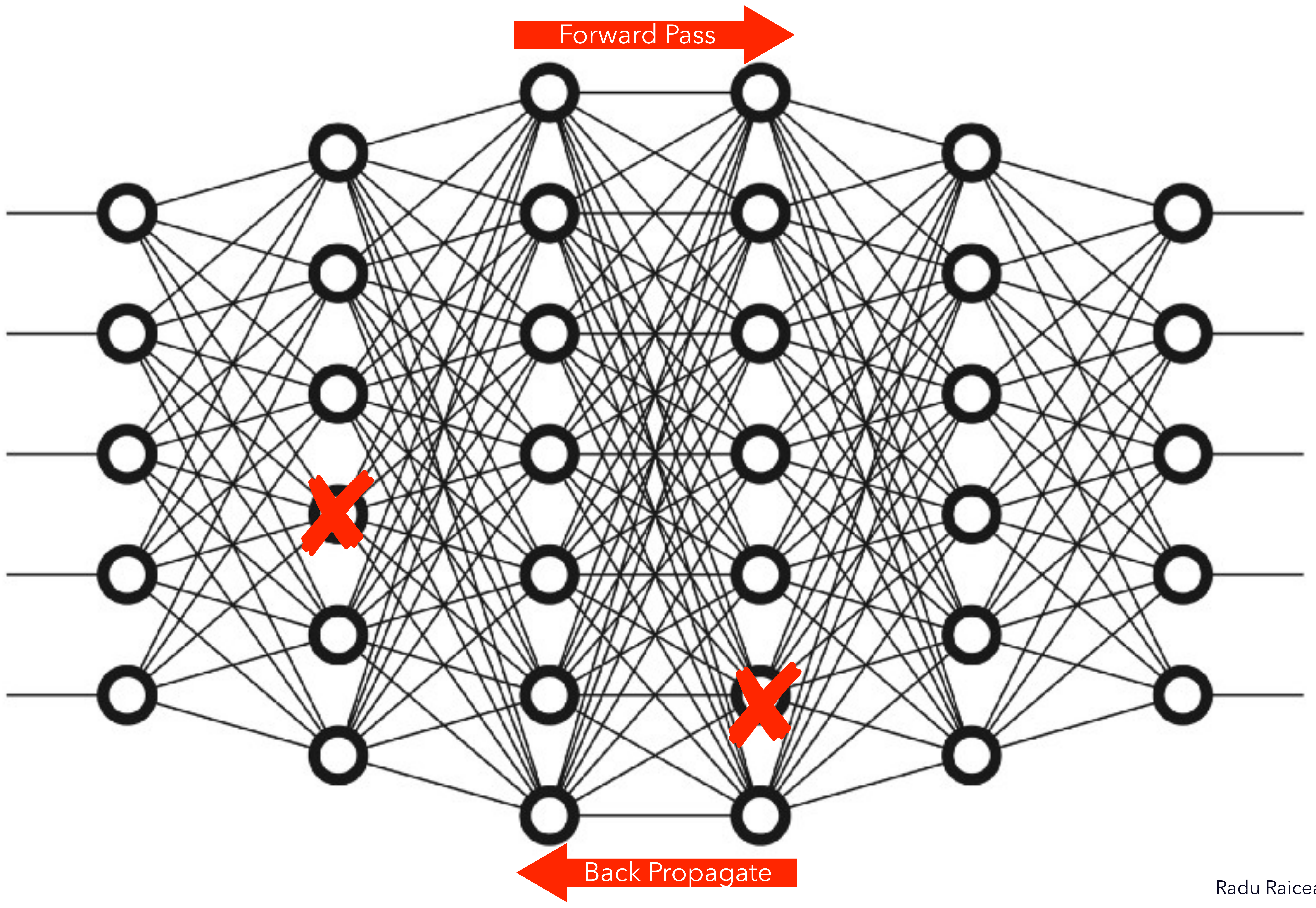
Radu Raicea

Dropout: train



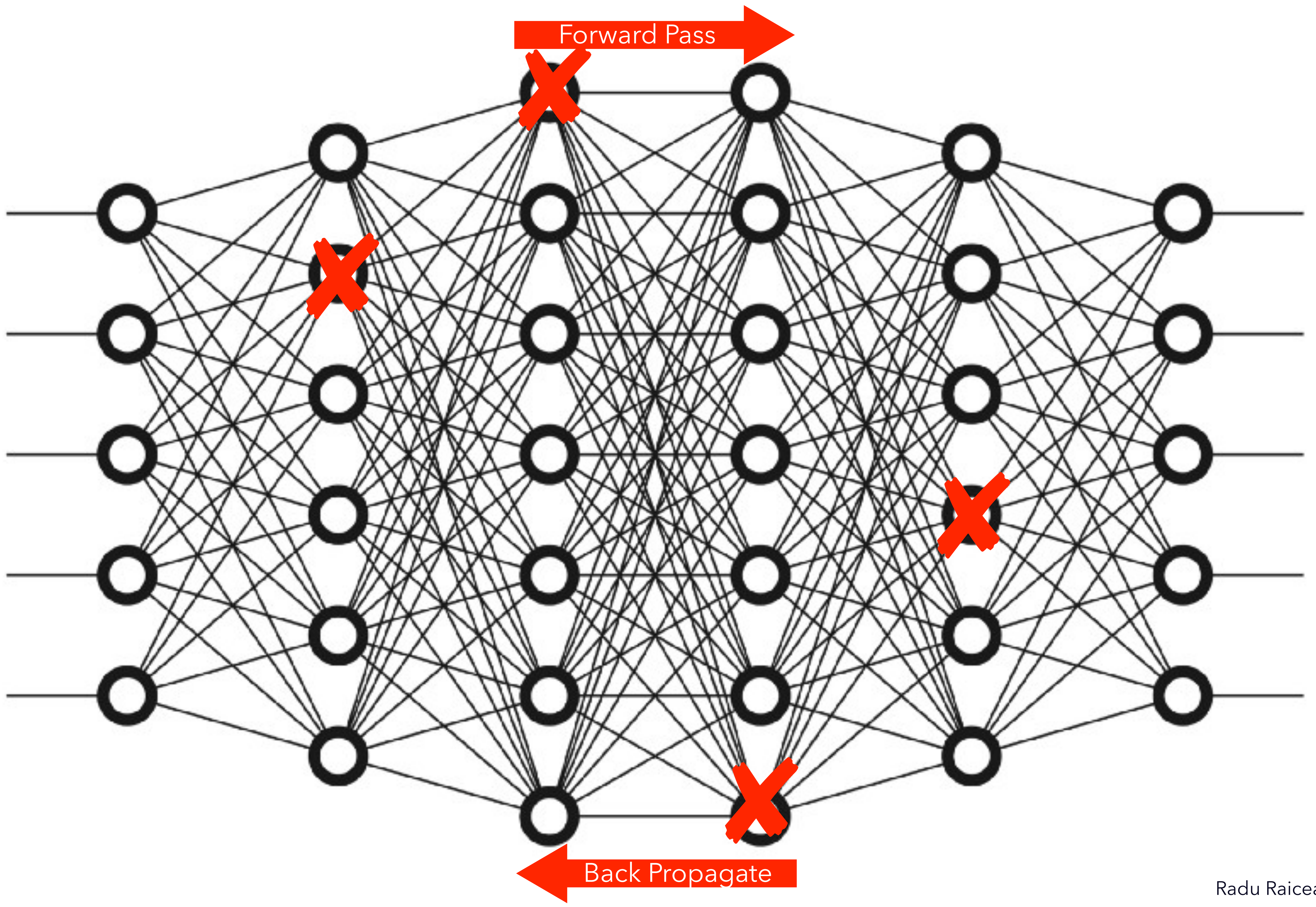
Radu Raicea

Dropout: train



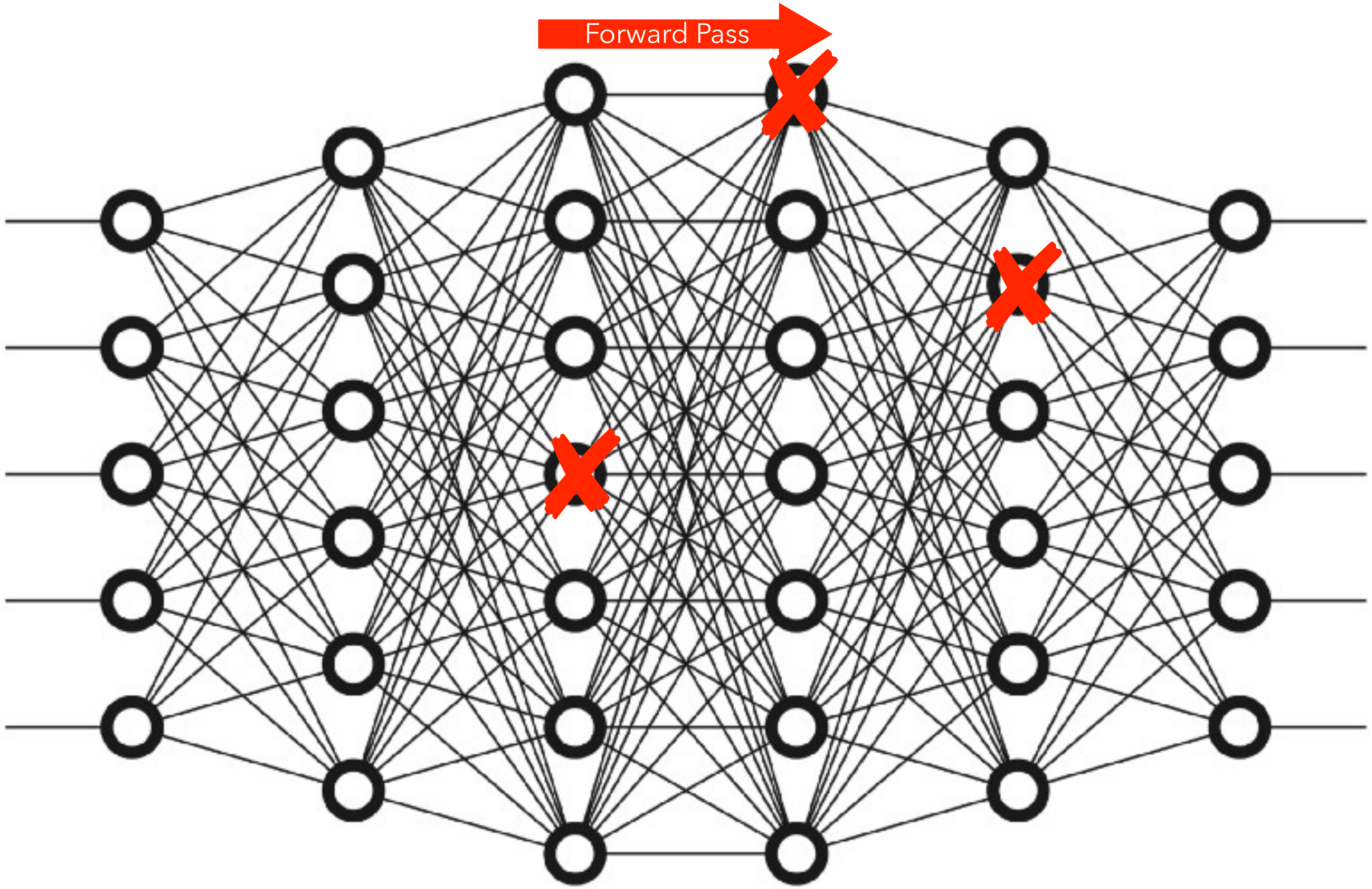
Radu Raicea

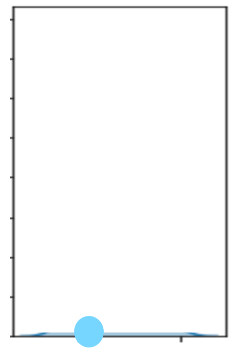
Dropout: train



Radu Raicea

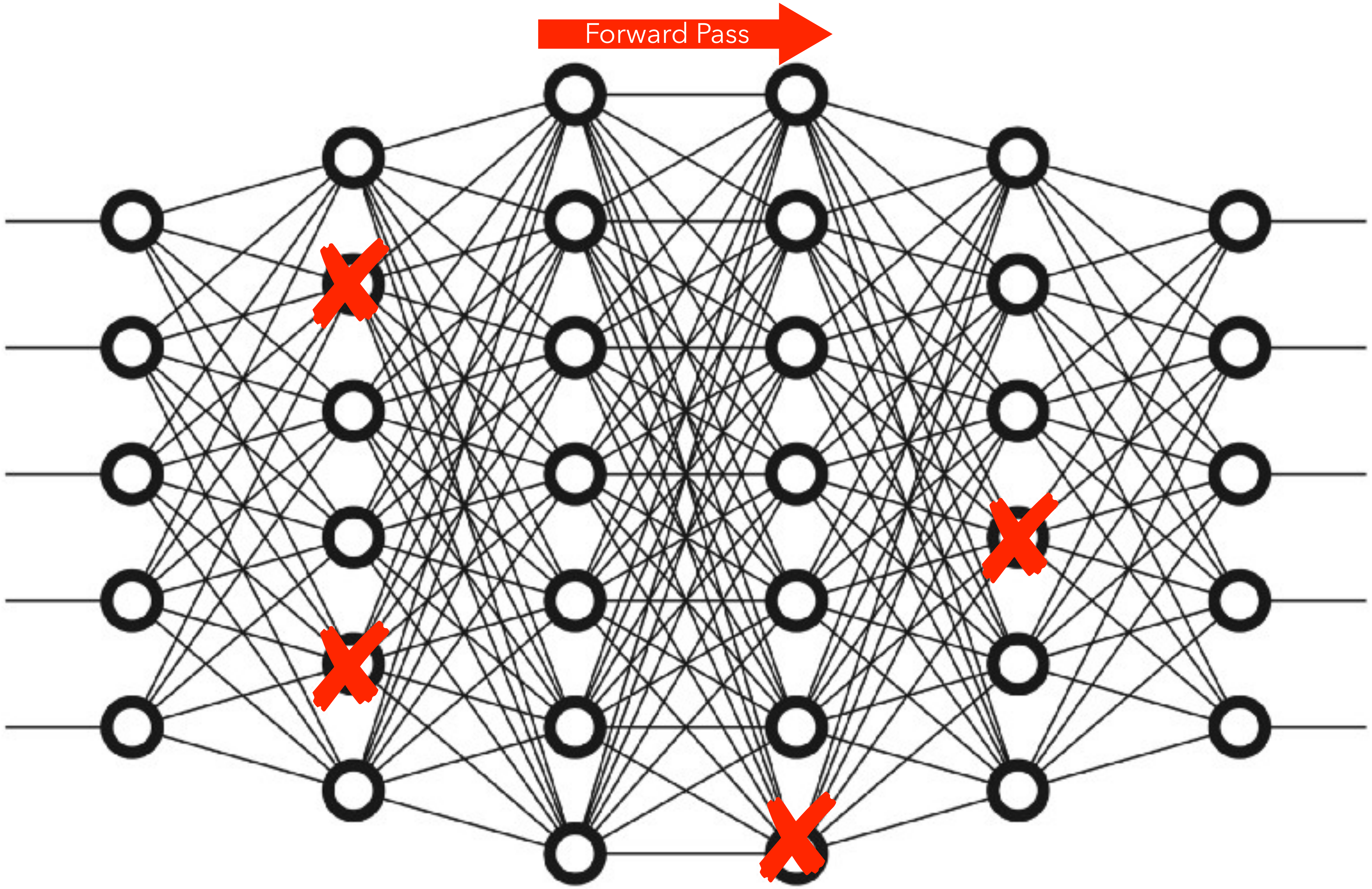
Dropout: evaluate

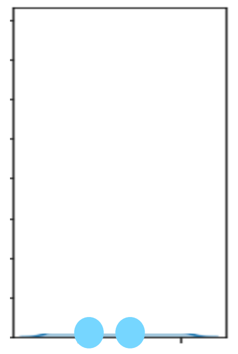




Radu Raicea

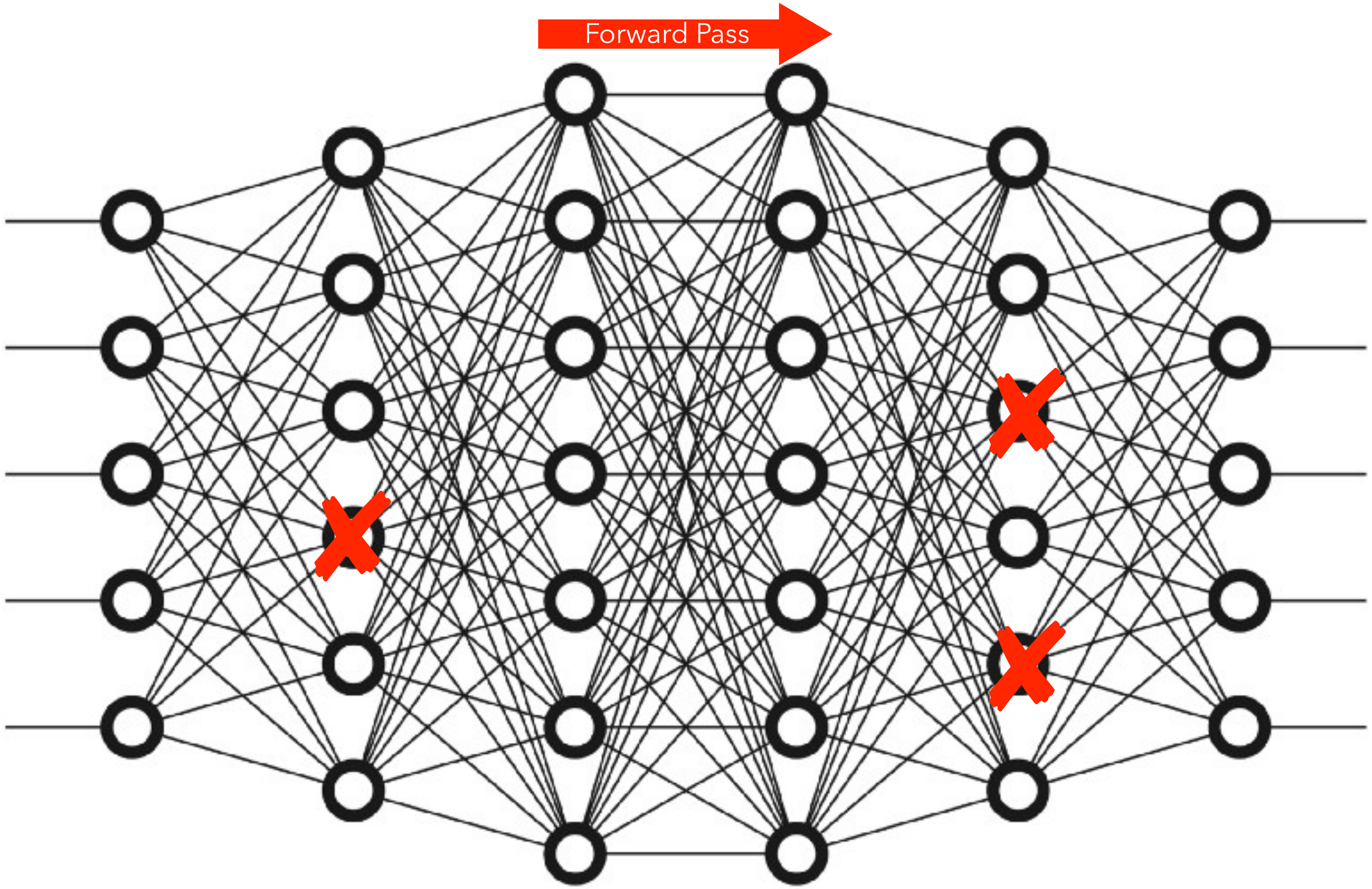
Dropout: evaluate

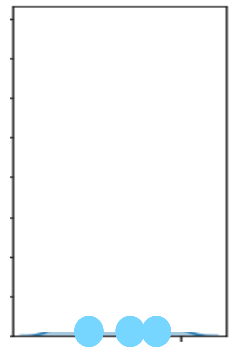




Radu Raicea

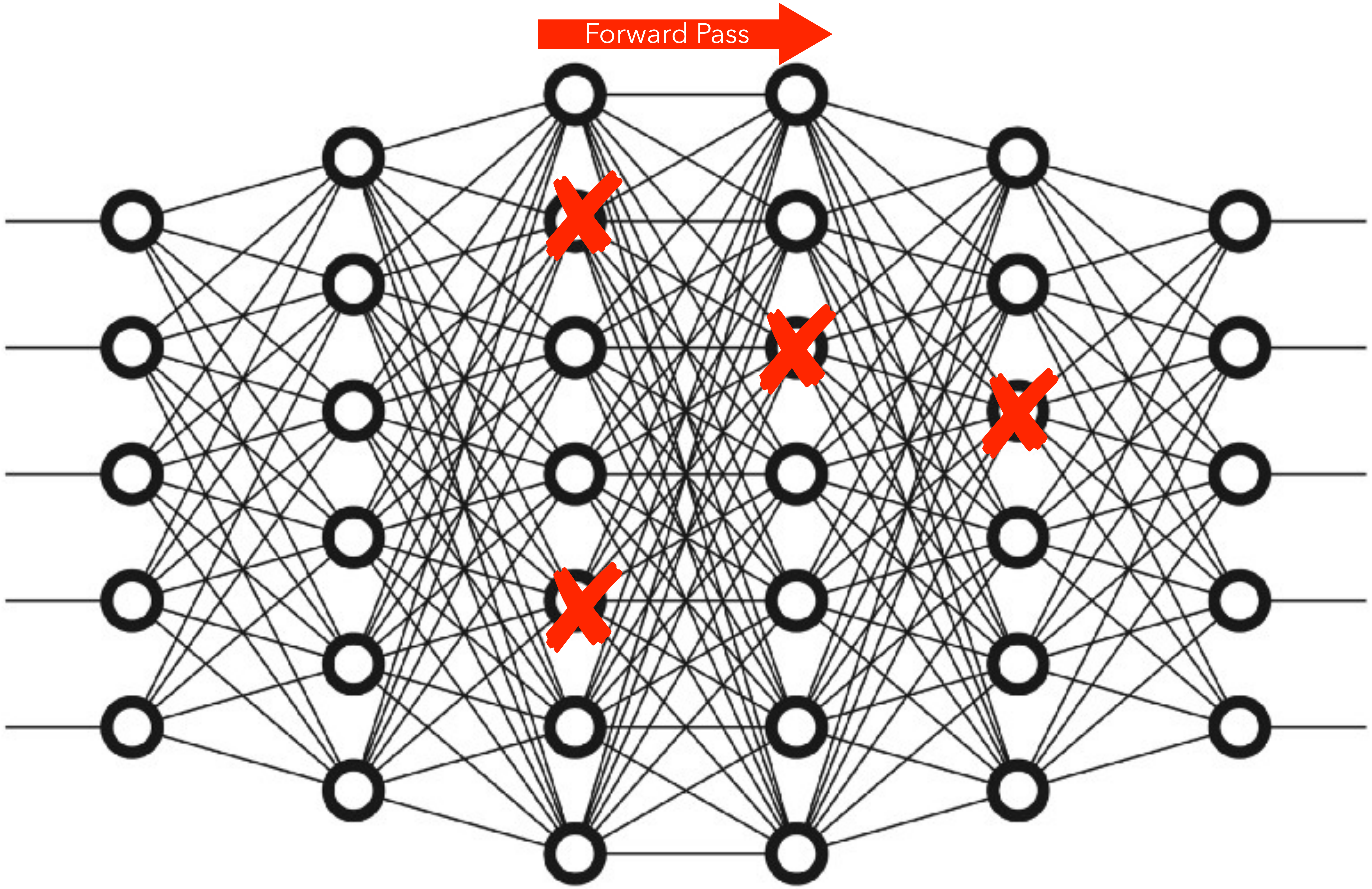
Dropout: evaluate

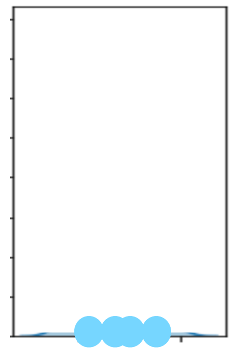




Radu Raicea

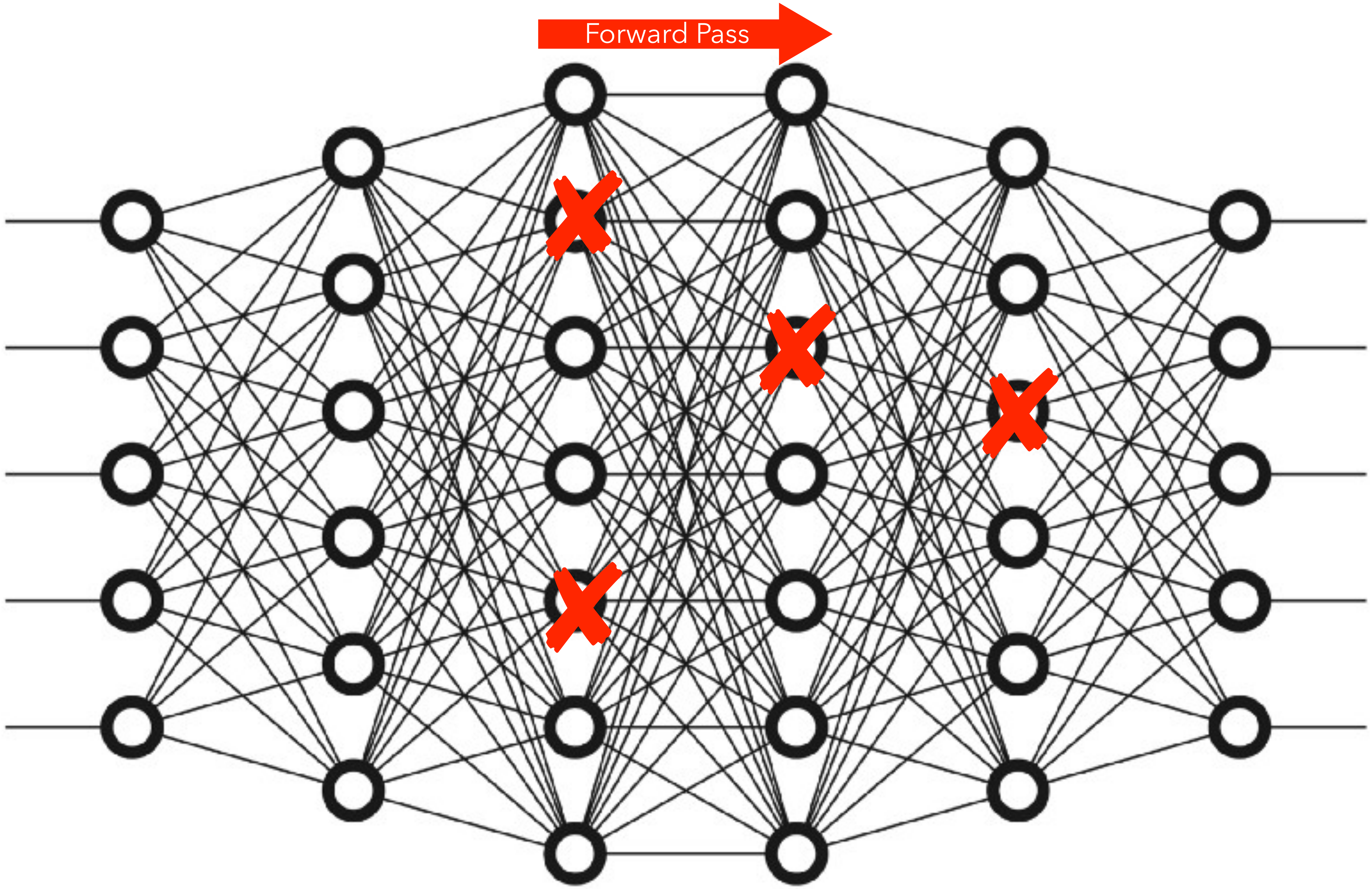
Dropout: evaluate

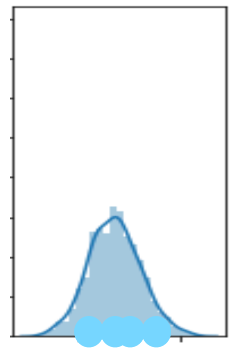




Radu Raicea

Dropout: evaluate



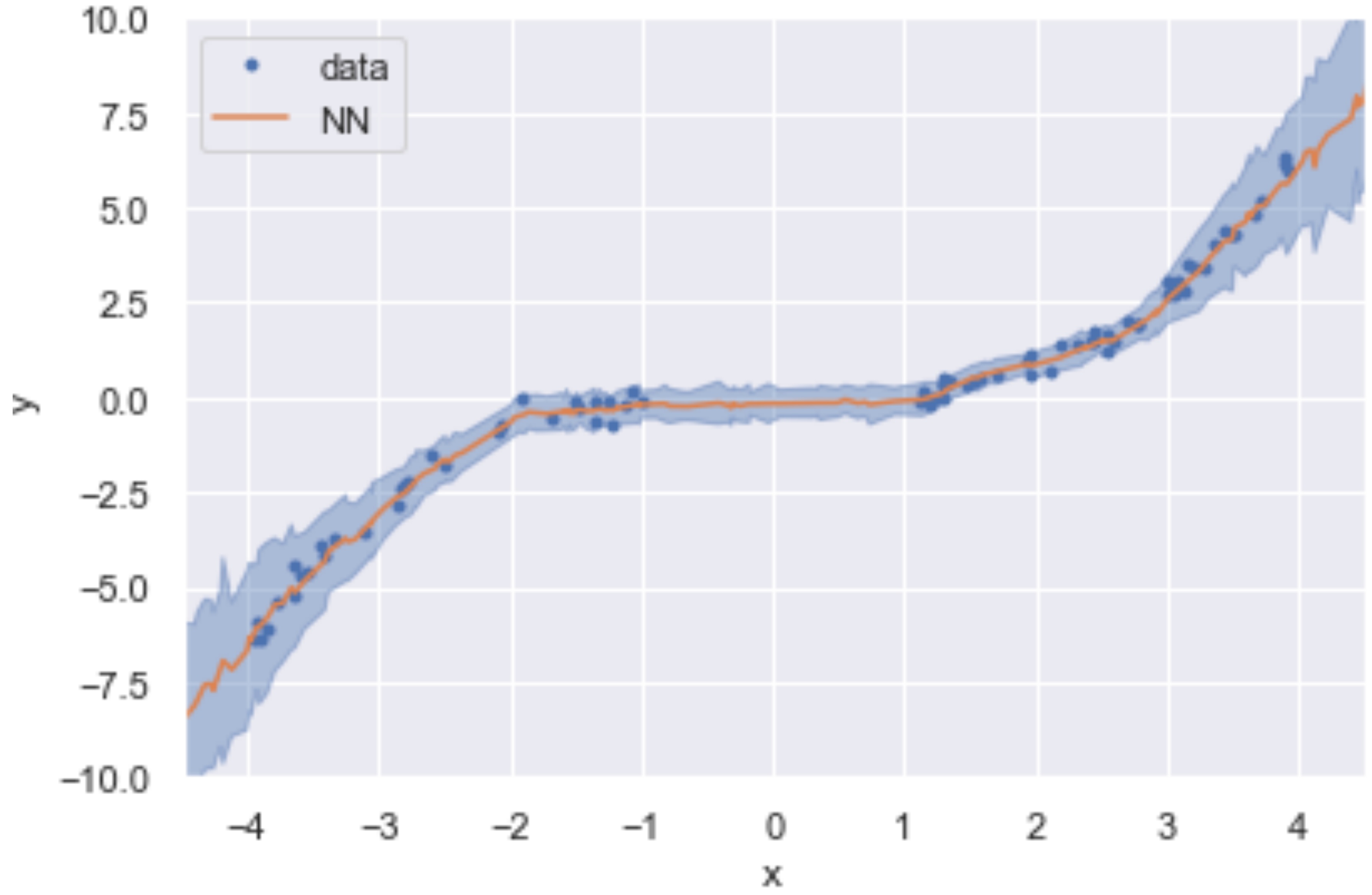


Radu Raicea

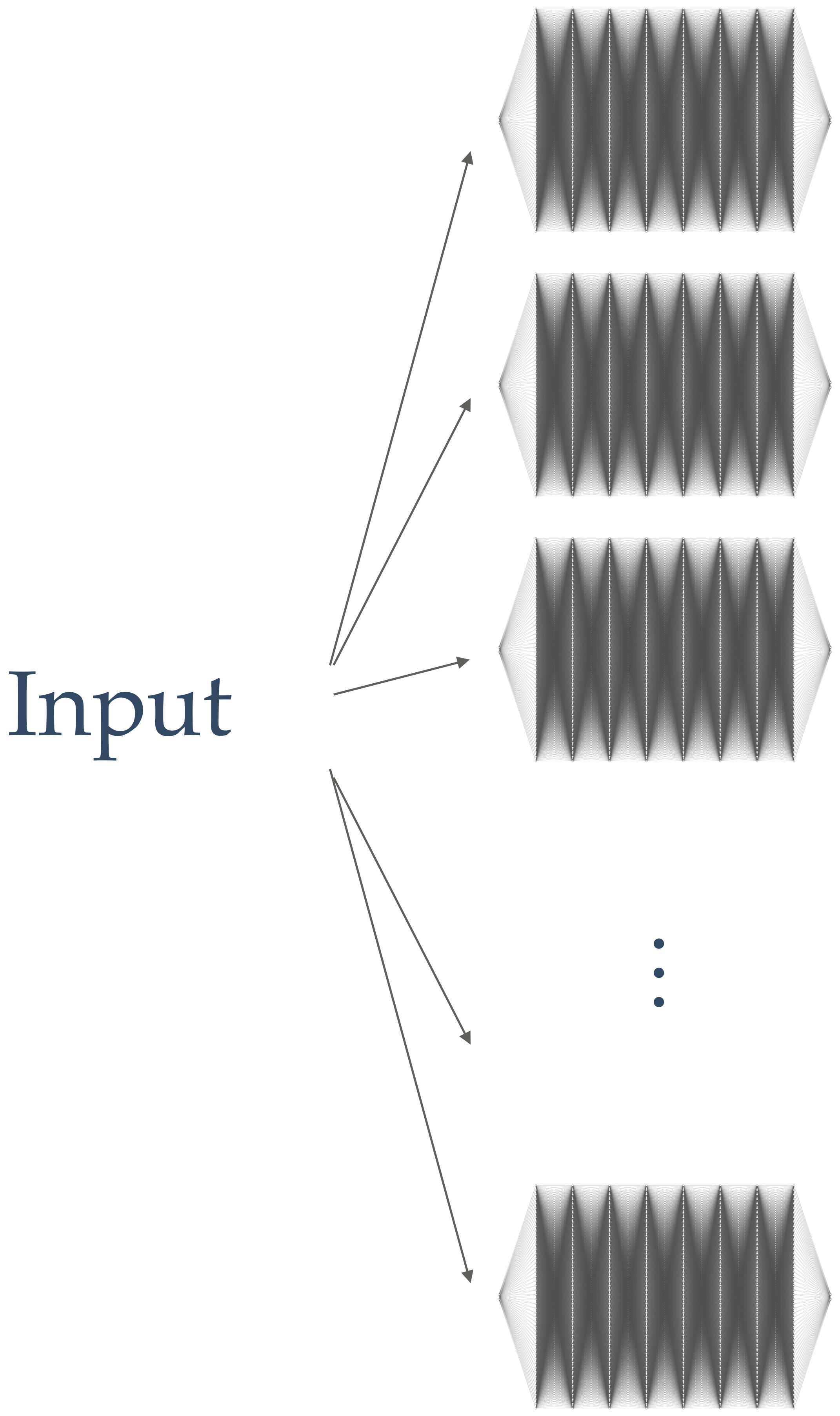
Dropout

MCMC

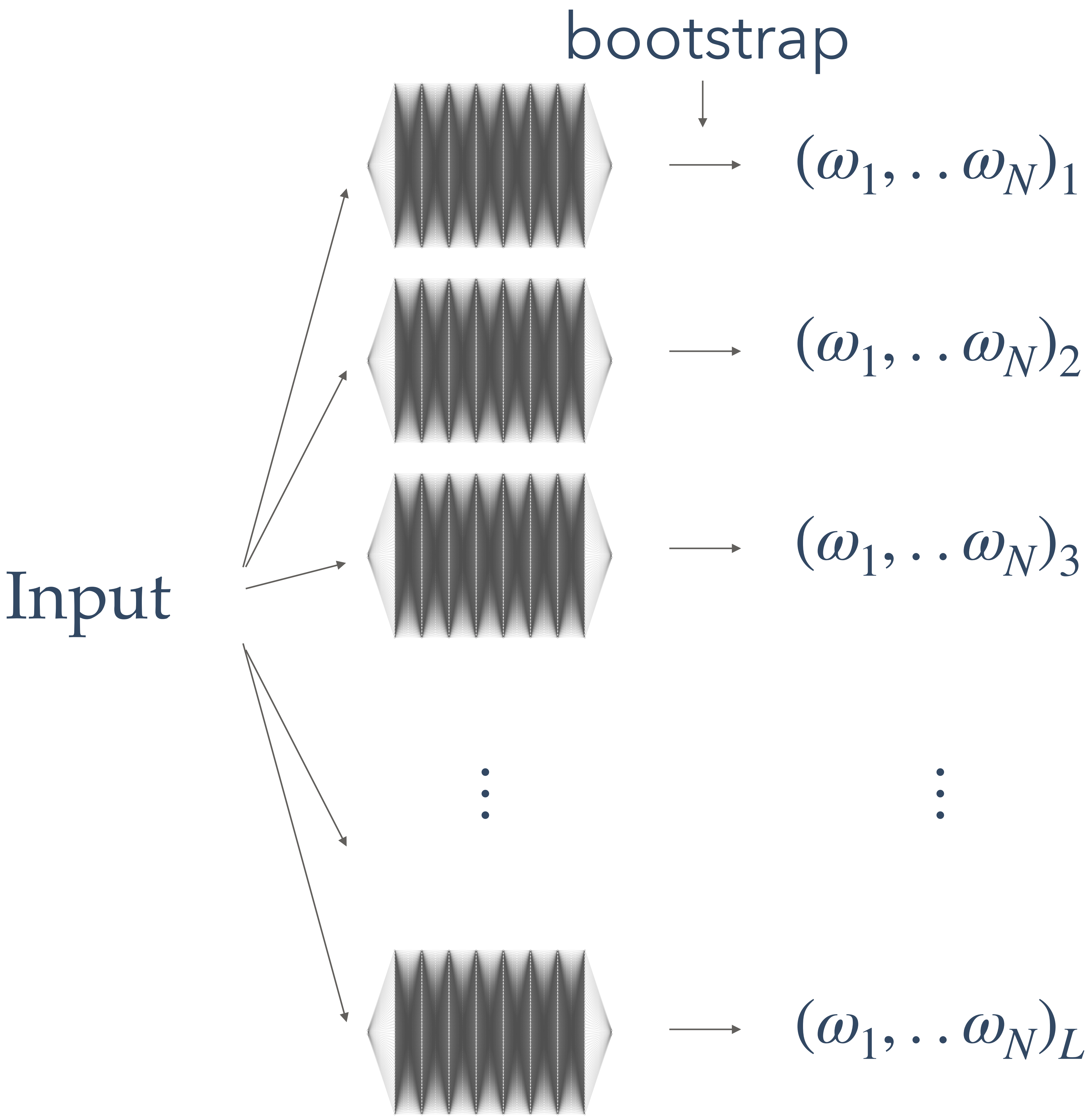
Dropout



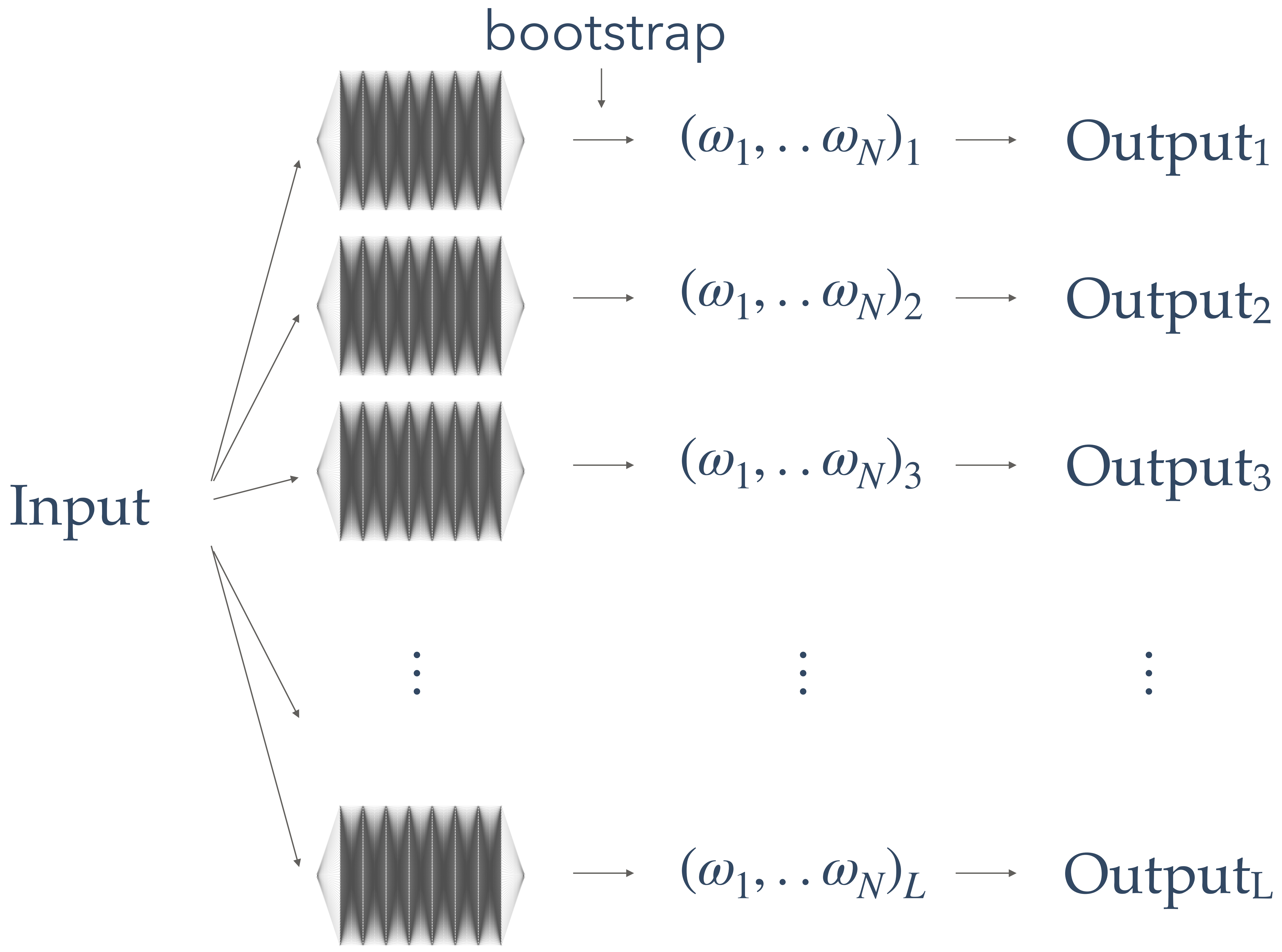
Bootstrap: L models



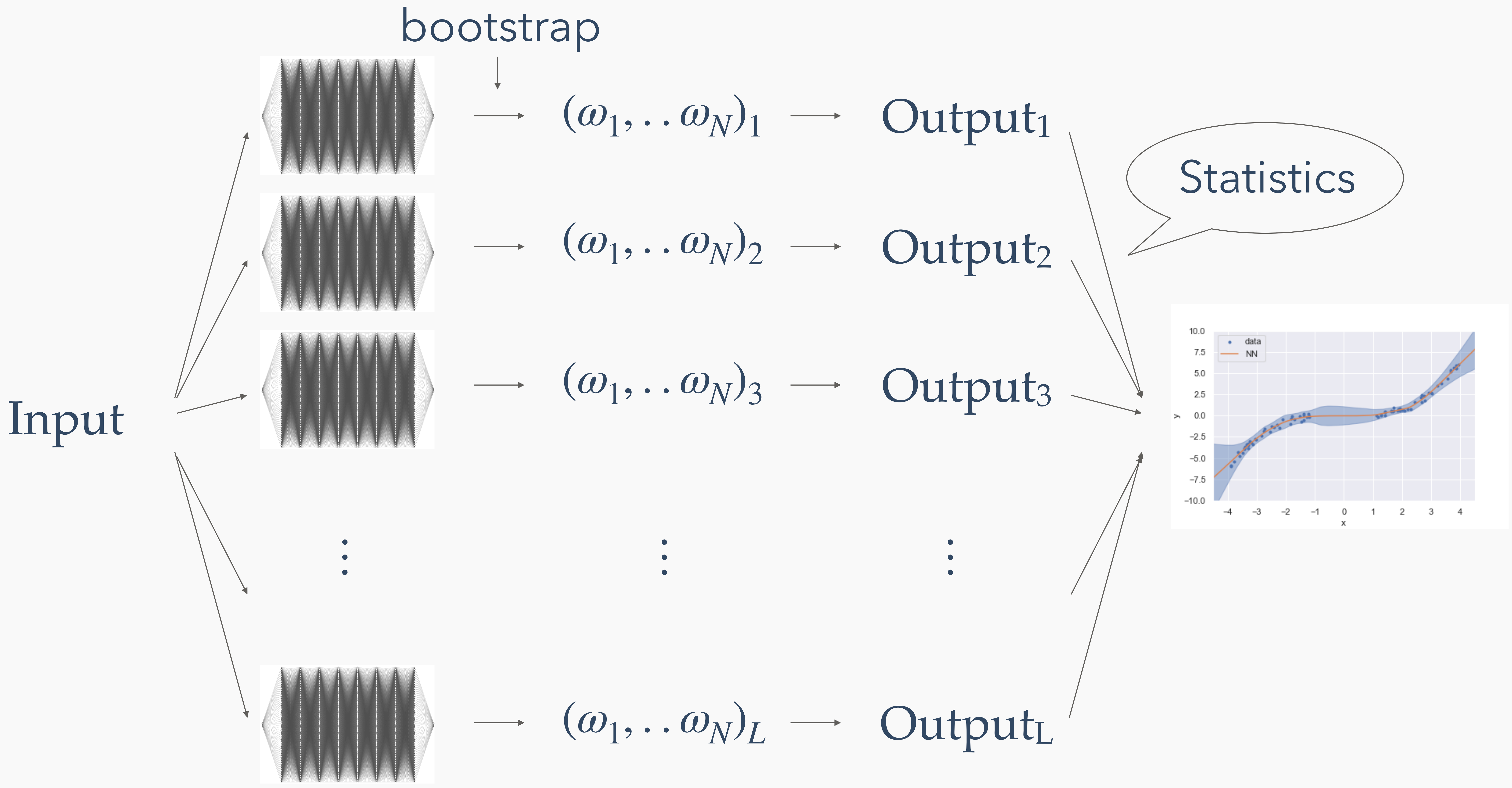
Bootstrap: L models



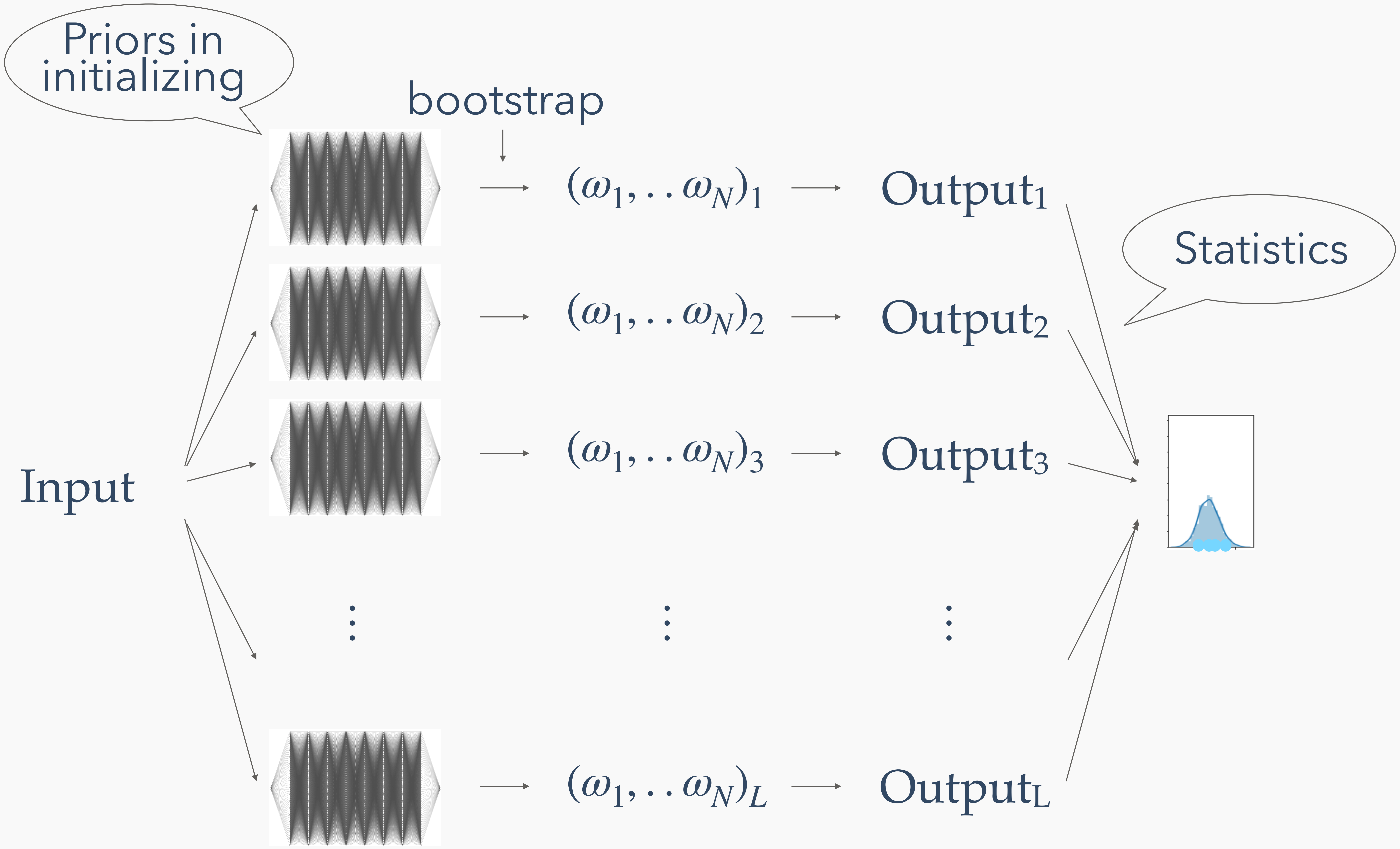
Bootstrap: L models



Bootstrap: L models



Bootstrap: L models

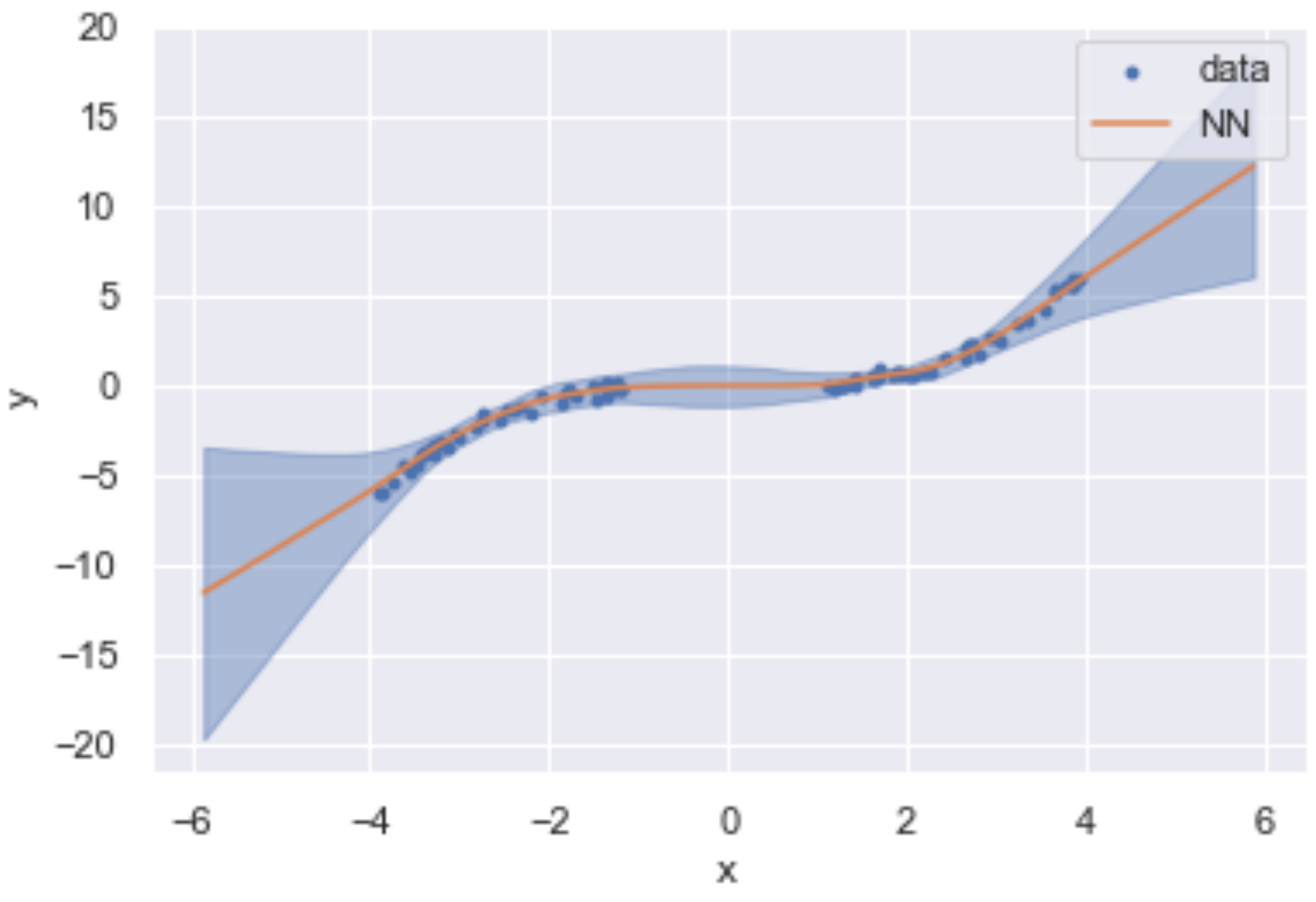


Bootstrap: L models

MCMC

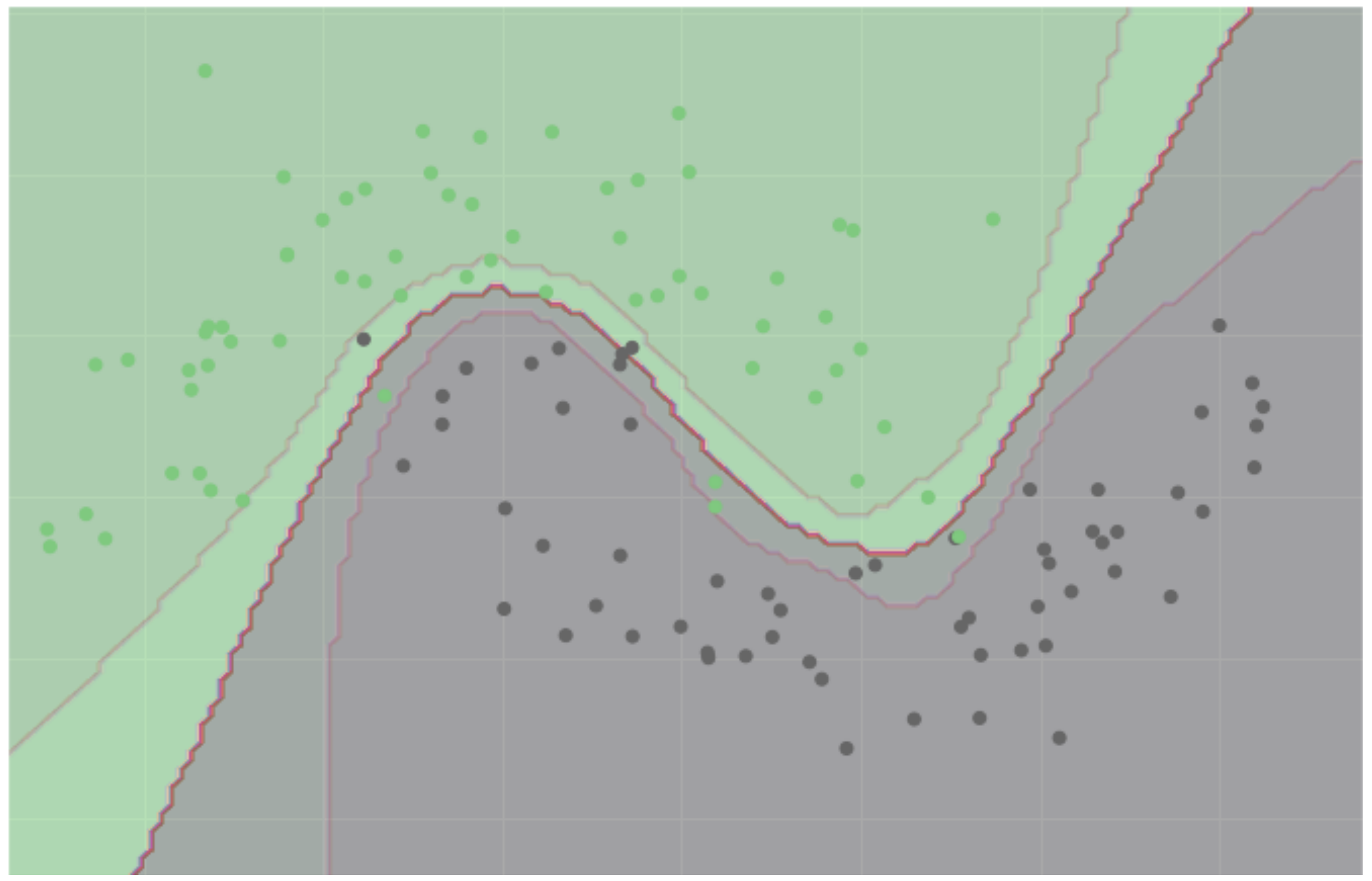
Bootstrap

Regression



Model Mean
95% models

Classification



Comparing Models of Uncertainty Quantification: Reading material

Quality of Uncertainty Quantification for Bayesian Neural Network Inference

Jiayu Yao [*1] **Weiwei Pan** [*1] **Soumya Ghosh** [2] **Finale Doshi-Velez** [1]

https://arxiv.org/pdf/1906.09686.pdf

"Frequently in literature, high test log likelihood is used as evidence that the inference procedure has more faithfully captured the true posterior. However, here we argue that while test log likelihood may be a good criteria for model selection, it is not a reliable criteria for determining how well an approximate posterior aligns with the true posterior."

They compare 10 commonly used approximate inference procedures for Bayesian NNs. They find that approximate Bayesian inference methods typically do not capture true posteriors, and that non-Bayesian methods often do not capture the desired predictive uncertainty.

We need more careful metrics for evaluating the performance of methods on uncertainty quantification

The End

Questions?