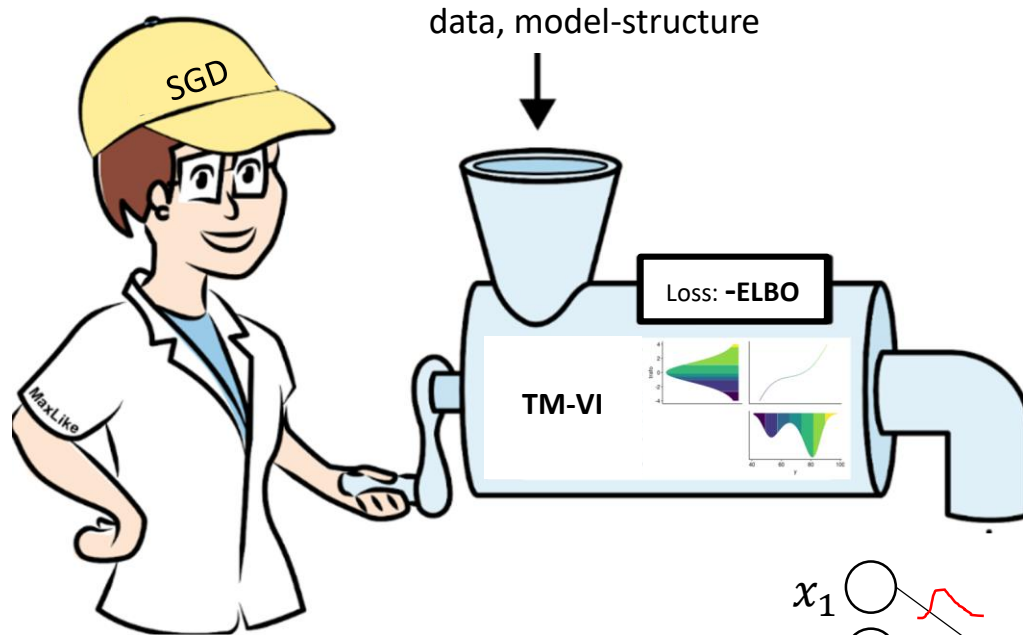


# Bayes for dummies

## Transformation Models for Flexible Posteriors in Variational Bayes



### Goal:

Get a fitted Bayesian model with **flexible posteriors** for the model parameters

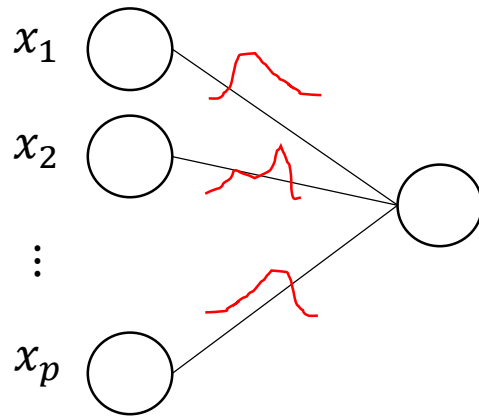
<https://arxiv.org/abs/2106.00528>

Prelude

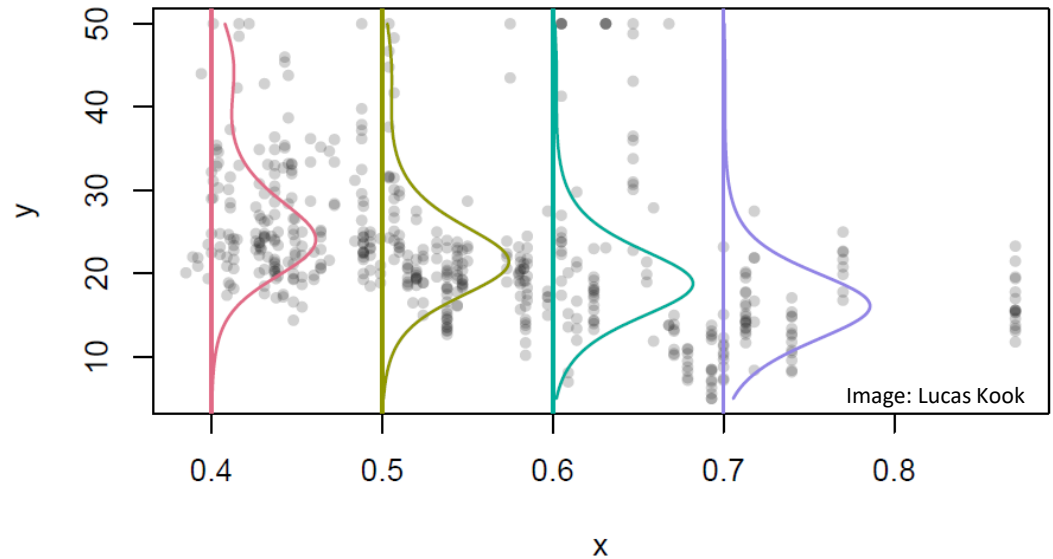
Transformation Models

# Modeling complex distributions

complex posteriors



complex conditional probability distributions



Sometimes we need complex distributions and don't know the distribution family.

# How to model complex distributions?

- Use a mixture model (e.g. mixture Gaussians)
- Use a transformation model!

# The idea of transformation models (TM)

The heart of a TM is a **bijective transformation function**  $h_\lambda$  that transforms between a **simple distribution**  $p(z) = N(0,1)$  and a potentially complex **distributional**  ${}^{TM}q_\lambda(w)$

“change of variable” formula 
$${}^{TM}q_\lambda(w) = p(z) \cdot \left| \frac{\partial h_\lambda(z)}{\partial z} \right|^{-1}$$

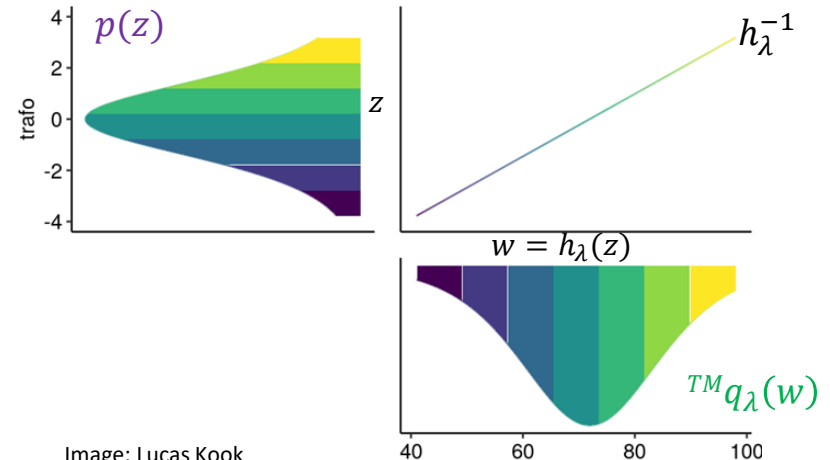
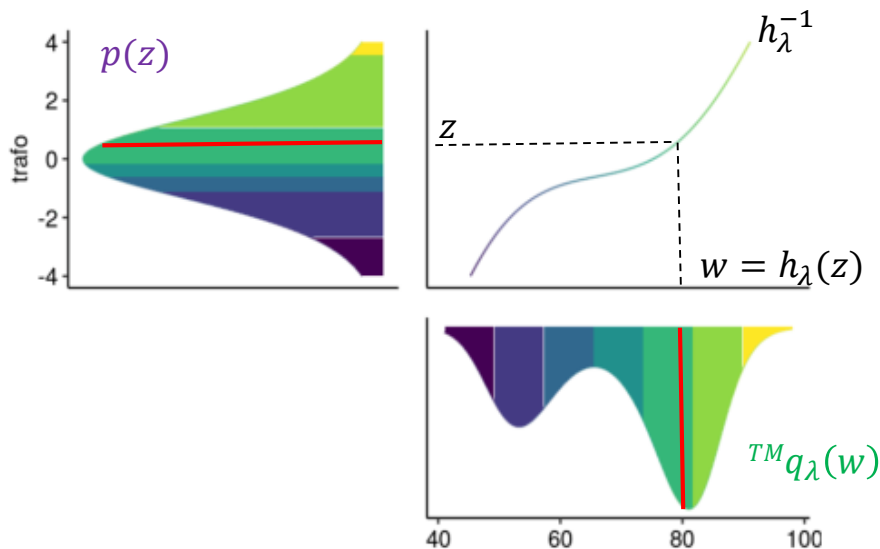


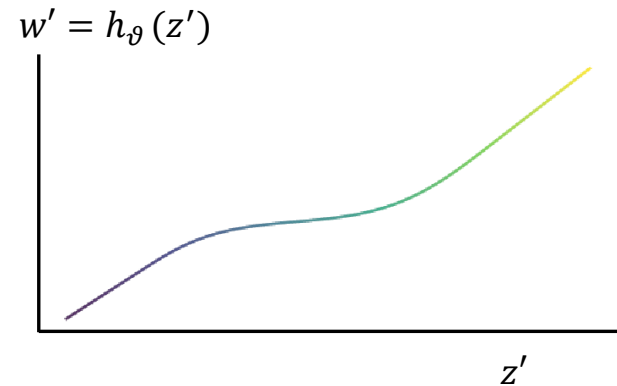
Image: Lucas Kook

Fitting a complex distributions requires a complex transformation function  $h_\lambda$

# Bernstein-polynomial

$$w' = h_{\vartheta}(z') = \sum_{k=1}^M \frac{\vartheta_k}{M+1} \text{Be}_k(z')$$

$$z' \in [0,1]$$



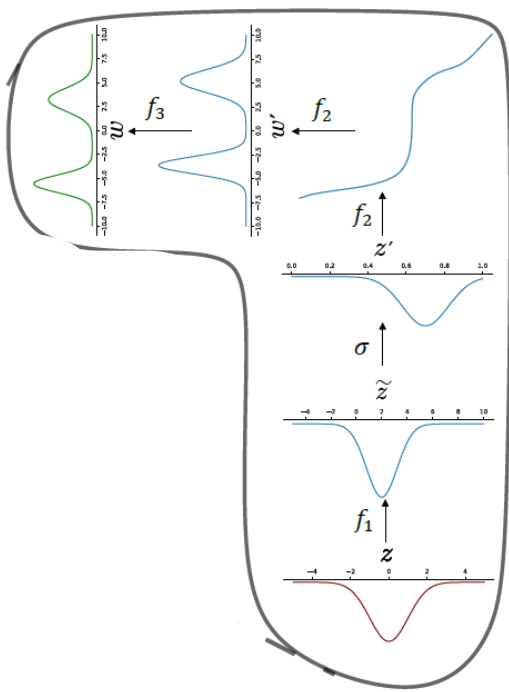
A Bernstein polynomial has nice properties:

- It can **approximate every function** on the support  $[0; 1]$
- It's flexibility can be controlled by the order  $M$
- It is bijective, i.e. **monotone increasing**, if parameters  $\vartheta_1 \leq \vartheta_2 \leq \dots \leq \vartheta_M$

# Constructing a transformation function

$$f_3(w') = \alpha \cdot w' + \beta$$

$${}^{TM}q_{\lambda}(w) = p(z) \cdot \left| \frac{\partial h_{\lambda}(z)}{\partial z} \right|^{-1}$$



$$f_2(z') = \sum_{i=0}^M \text{Be}_i(z') \frac{\vartheta_i}{M+1}$$

$$f_1(\tilde{z}) = a \cdot z + b$$

$$p(z) = N(0,1)$$

$h_{\lambda} = f_3 \circ f_2 \circ \sigma \circ f_1$  has M+5 parameters:  $\lambda = (a, b, \alpha, \beta, \vartheta_0, \dots, \vartheta_M)$

**Back to Bayes!**

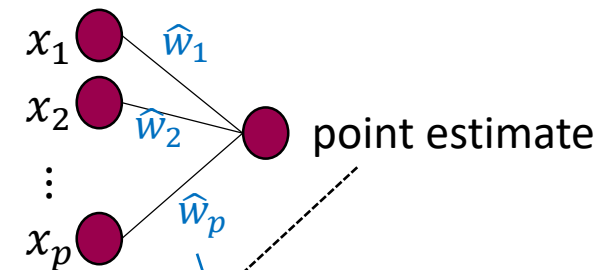


# Bayes is often used for probabilistic modeling

Bayesian models allow to capture parameter uncertainty and outcome uncertainty.

Example: Linear regression :

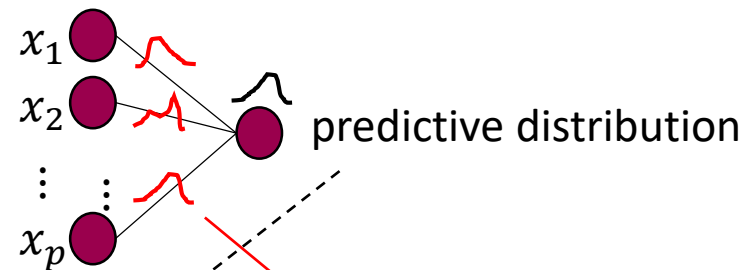
non-probabilistic model



$$\hat{\mu}(x) = \sum_{i=1}^p \hat{w}_i \cdot x_i$$

point estimates for  
the model parameters

probabilistic Bayesian model



$$p(\mu|x, D) = \int p(\mu|x, \mathbf{w}) \cdot p(\mathbf{w}|D) d\mathbf{w}$$

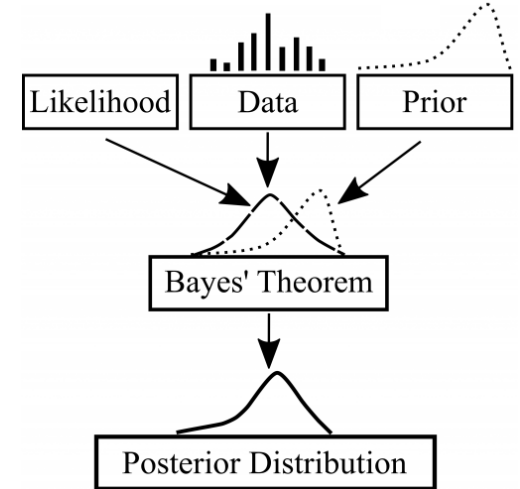
posterior of the  
model parameters

# Compute posteriors via Bayes' theorem

posterior      likelihood      prior

$$p(w|D) = \frac{p(D|w) \cdot p(w)}{p(D)} = \frac{p(D|w)p(w)}{\int p(D|w)p(w)dw} \sim p(D|w)p(w)$$

normalizing denominator  
**this is the most difficult part!**



Source: <https://towardsdatascience.com/bayesian-statistics-for-data-science-45397ec79c94>

# Compute posteriors via Bayes' theorem

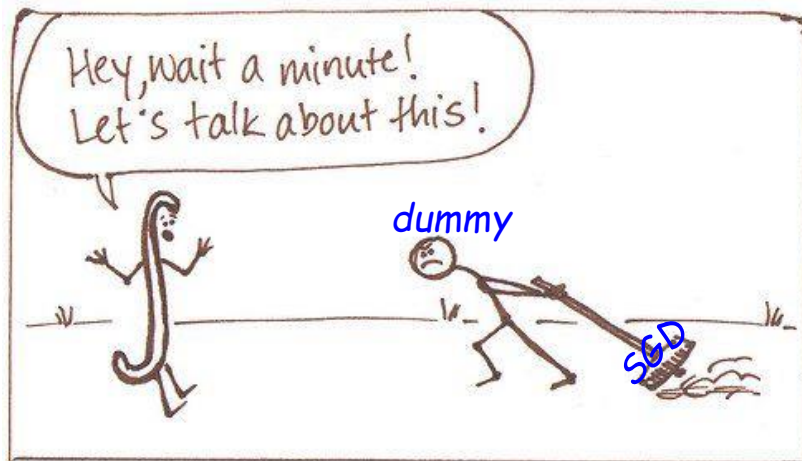
$$p(w|D) = \frac{p(D|w) \cdot p(w)}{p(D)} = \frac{p(D|w)p(w)}{\int p(D|w)p(w)dw} \sim p(D|w)p(w)$$

posterior

likelihood

prior

normalizing denominator  
**this is the most difficult part!**



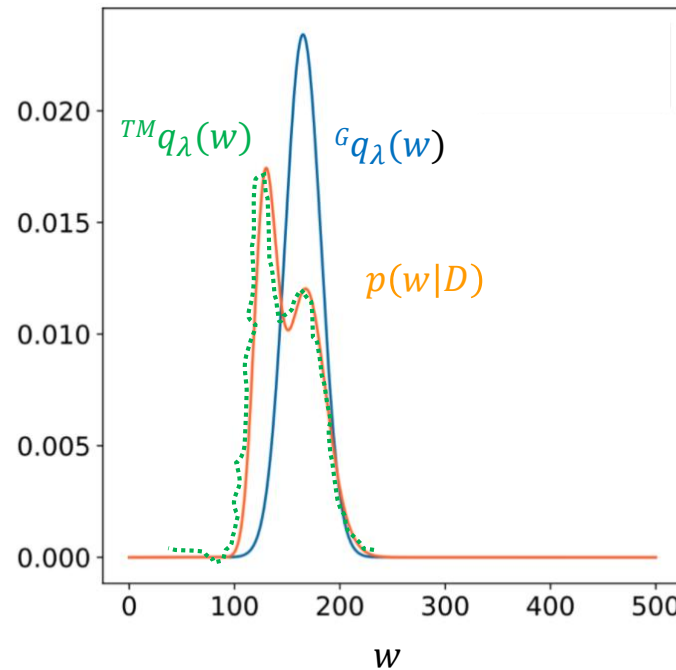
<https://brownsharpie.courtneygibbons.org/comic/integrand/>

- Bayes theorem is dead
- Long live TM-VI

# The idea of Variational Inference (VI)

Approximate **posterior**  $p(w|D)$  by a variational distribution  $q_\lambda(w)$

- **Gaussian-VI**: Use a **Gaussian** as variational distribution  $^Gq_\lambda(w)$
- **TM-VI**: Use a transformation model to get a **flexible**  $^{TM}q_\lambda(w)$



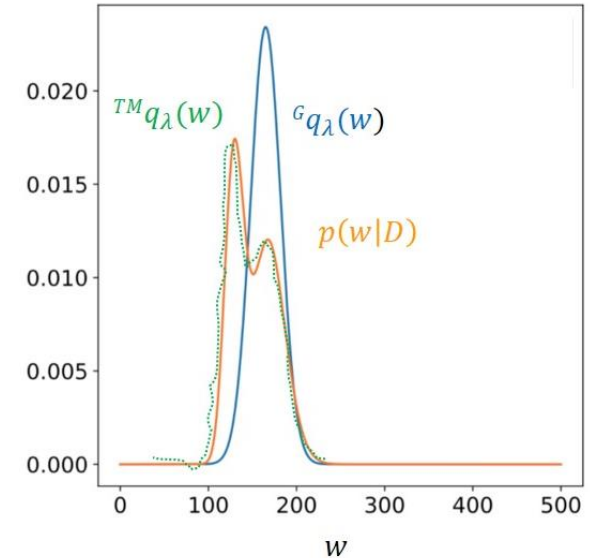
Gaussian-VI is not flexible enough to approximate complex posteriors.

# Variational inference is an optimization problem

Find the best  $q_\lambda$  by optimizing parameters  $\lambda$  so that

- Kullback-Leibler divergence between variational distribution and posterior distribution is minimized

$$\begin{aligned} KL(q_\lambda(w) || p(w|D)) &= E_{w \sim q_\lambda} \left( \log \left( \frac{q_\lambda(w)}{p(w|D)} \right) \right) \\ &= \log(D) - \underbrace{\left( E_{w \sim q_\lambda} (\log(p(D|w))) - E_{w \sim q_\lambda} \left( \log \left( \frac{q_\lambda(w)}{p(w)} \right) \right) \right)}_{\text{ELBO}(\lambda)} \end{aligned}$$

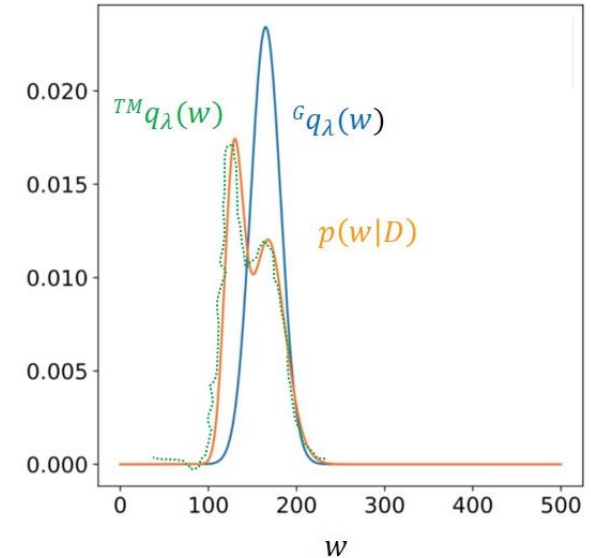


# Variational inference is an optimization problem

Task: tune variational parameters  $\lambda$  to

- minimize the Kullback-Leibler divergence between variational distribution and posterior distribution

$$\begin{aligned} KL(q_\lambda(w) || p(w|D)) &= E_{w \sim q_\lambda} \left( \log \left( \frac{q_\lambda(w)}{p(w|D)} \right) \right) \\ &= \log(D) - \underbrace{\left( E_{w \sim q_\lambda} (\log(p(D|w))) - E_{w \sim q_\lambda} \left( \log \left( \frac{q_\lambda(w)}{p(w)} \right) \right) \right)}_{\text{ELBO}(\lambda)} \end{aligned}$$



- $\Leftrightarrow$  maximize the evidence lower bound (ELBO)  $\Leftrightarrow$  minimize loss =  $-\text{ELBO}(\lambda)$

0) Initialize  $\lambda$

1) sample  $w_t \sim q_\lambda$

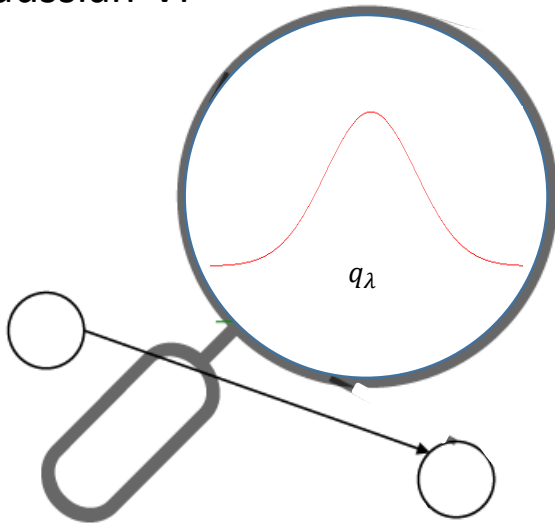
2) loss =  $-\text{ELBO}(\lambda) \approx -\left( \frac{1}{T} \sum_t \log(p(D|w_t)) - \frac{1}{T} \sum_t \log \left( \frac{q_\lambda(w_t)}{p(w_t)} \right) \right) \xrightarrow{\text{SGD}} \lambda_{\text{update}}$

# Gaussian-VI versus TM-VI

$$q_{\lambda}(w) = N(\mu, \sigma)$$

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

Gaussian-VI



0) Initialize  $\lambda$

1)  $w_{\text{sample}} \sim q_{\lambda}$

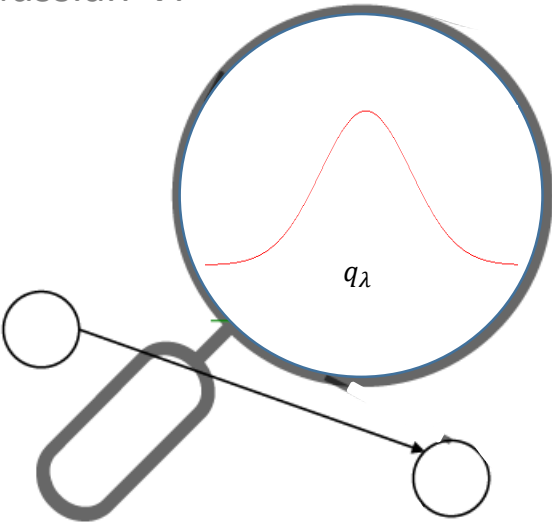
2)  $\text{loss} = -\text{ELBO}(\lambda) \xRightarrow{\text{SGD}} \lambda_{\text{update}}$

# Gauss-VI versus TM-VI

$$^Gq_{\lambda}(w) = N(\mu, \sigma)$$

$$^G\lambda = (\mu, \sigma)$$

Gaussian-VI



0) Initialize  $\lambda$

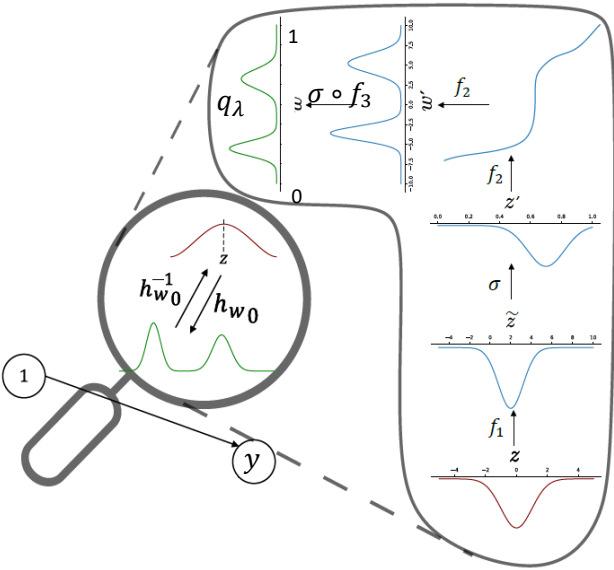
1)  $w_{\text{sample}} \sim q_{\lambda}$

2)  $\text{loss} = -\text{ELBO}(\lambda) \xrightarrow{\text{SGD}} \lambda_{\text{update}}$

$$^Gq_{\lambda}(w) = p(z) \cdot \left| \frac{\partial h_{\lambda}(z)}{\partial z} \right|^{-1}$$

$$^{\text{TM}}\lambda = (a, b, \alpha, \beta, \vartheta_0, \dots, \vartheta_M)$$

TM-VI



0) Initialize  $\lambda$

1)  $z_{\text{sample}} \sim N(0,1) \Rightarrow w_{\text{sample}} = h_{\lambda}(z_{\text{sample}})$

2)  $\text{loss} = -\text{ELBO}(\lambda) \xrightarrow{\text{SGD}} \lambda_{\text{update}}$



# Single parameter models

# Bernoulli experiment as one-parameter-model

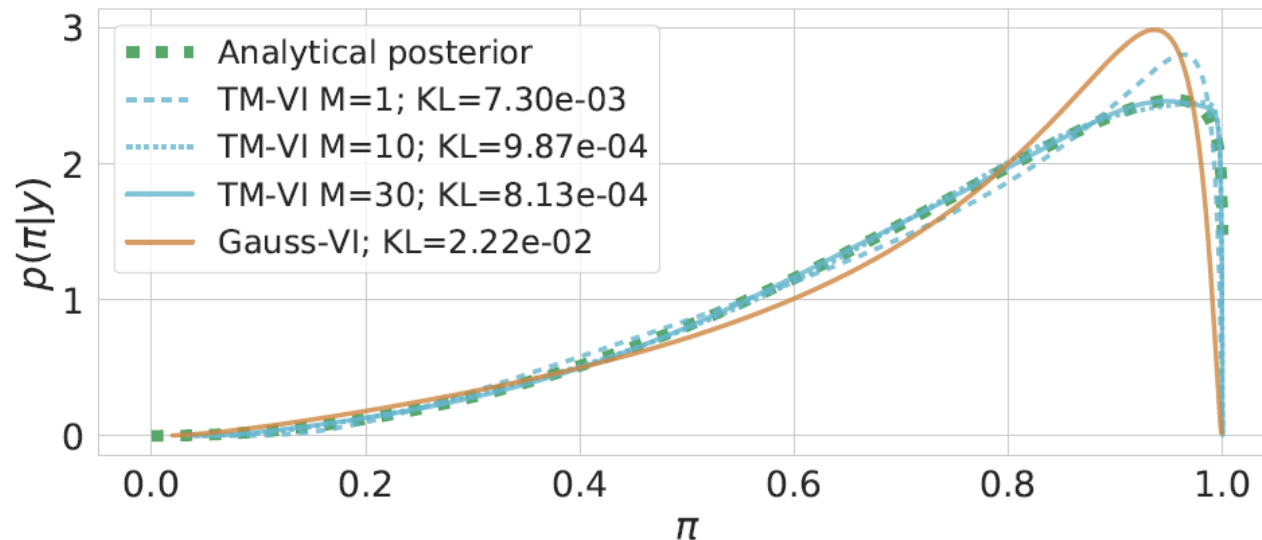
Bernoulli model  $y \sim \text{Ber}(\pi)$ ; two observations  $D = (y_1 = 1, y_2 = 1)$ .

## Exact analytical posterior:

Prior:  $p(\pi) = \text{Beta}(\alpha = 1.1, \beta = 1.1)$

Likelihood:  $p(D|\pi) = \pi \cdot \pi = \pi^2$

Posterior:  $p(\pi|D) = \text{Beta}(\alpha + \sum y_i, \beta + n - \sum y_i)$   
 $= \text{Beta}(3.3, 1.1)$



# Bernoulli experiment as one-parameter-model

Bernoulli model  $y \sim \text{Ber}(\pi)$ ; two observations  $D = (y_1 = 1, y_2 = 1)$ .

## Exact analytical posterior:

Prior:  $p(\pi) = \text{Beta}(\alpha = 1.1, \beta = 1.1)$

Likelihood:  $p(D|\pi) = \pi \cdot \pi = \pi^2$

Posterior:  $p(\pi|D) = \text{Beta}(\alpha + \sum y_i, \beta + n - \sum y_i)$   
 $= \text{Beta}(3.3, 1.1)$

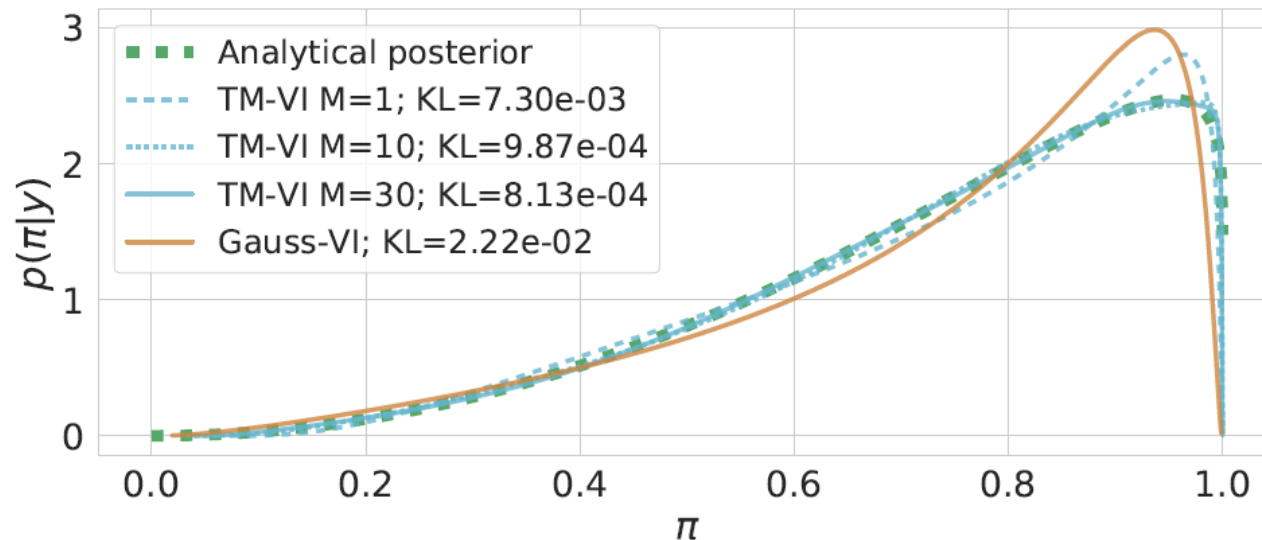
## VI-approximated posterior:

Gauss-VI :  ${}^Gq_\lambda(\pi) = \text{sigmoid}(N(\mu, \sigma))$ ,  $\lambda = (\mu, \sigma)$

TM-VI :  ${}^{\text{TM}}q_\lambda(\pi) = p(z) \cdot \left| \frac{\partial h_\lambda(z)}{\partial z} \right|^{-1}$

$h_\lambda = \sigma \circ f_3 \circ f_2 \circ \sigma \circ f_1$ ,  $\lambda = (a, b, \alpha, \beta, \vartheta_0, \dots, \vartheta_M)$

$\text{loss} = -\text{ELBO}(\lambda) \xrightarrow{\text{SGD}} \lambda_{\text{opt}} \Rightarrow q_{\lambda_{\text{opt}}}$

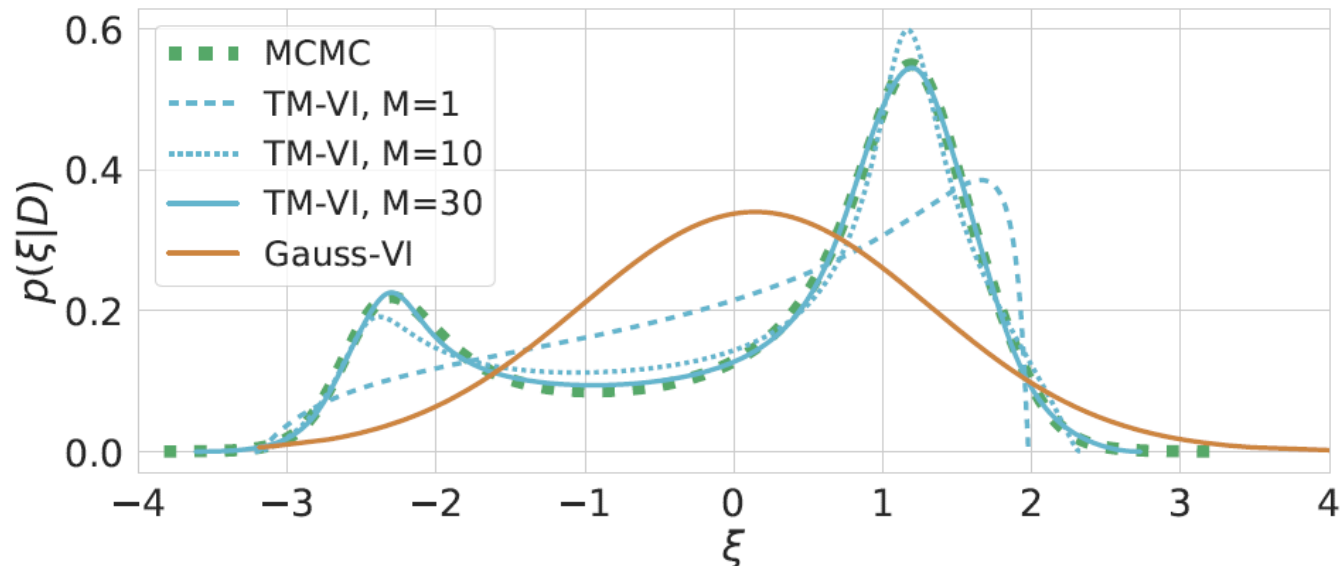


# Cauchy experiment as one-parameter-model

Cauchy model  $y \sim \text{Cauchy}(\xi; \gamma)$ ; 6 observations sampled from a mixture-Cauchy

## Exact posterior via MCMC (Stan):

```
data{
  int<lower=0> N;
  real<lower=0> gamma;
  vector[N] y;
}
parameters{
  real xi;
}
model{
  y ~ cauchy(xi, gamma); // likelihood
  xi ~ normal(0, 1);     // prior
}
```



# Cauchy experiment as one-parameter-model

Cauchy model  $y \sim \text{Cauchy}(\xi; \gamma)$ ; 6 observations sampled from a mixture-Cauchy

## Exact posterior via MCMC (Stan):

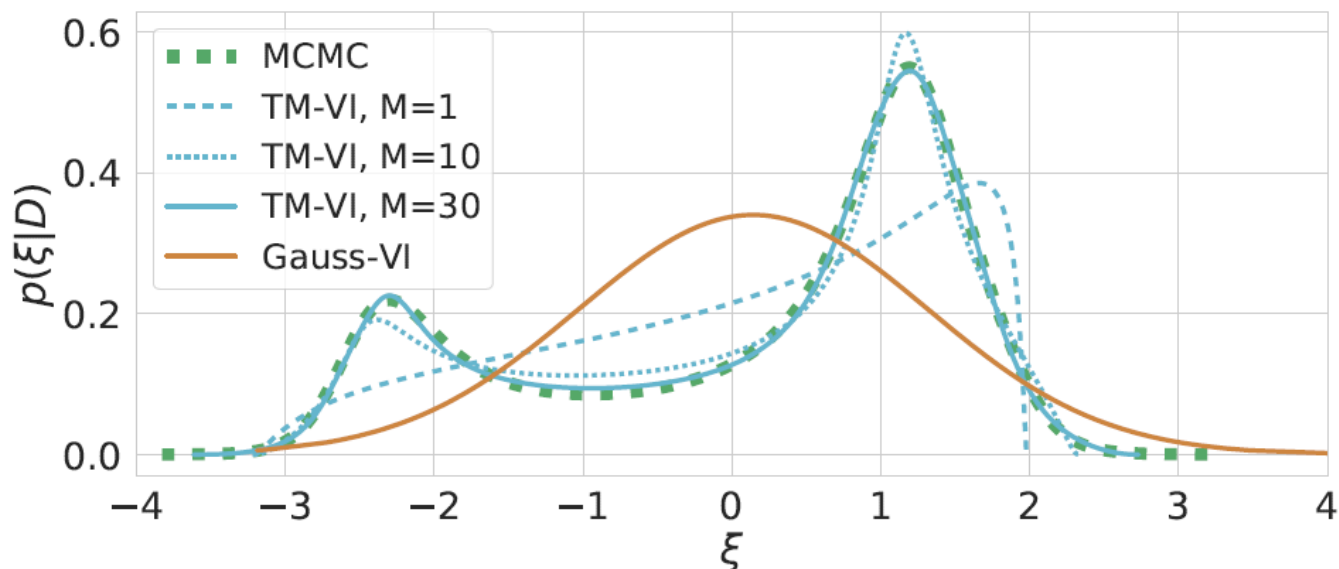
```
data{
  int<lower=0> N;
  real<lower=0> gamma;
  vector[N] y;
}
parameters{
  real xi;
}
model{
  y ~ cauchy(xi, gamma); // likelihood
  xi ~ normal(0, 1);     // prior
}
```

## VI-approximated posterior:

Gauss-VI :  $^G q_\lambda(\pi) = N(\mu, \sigma)$ ,  $\lambda = (\mu, \sigma)$

TM-VI :  $^M q_\lambda(\pi) = p(z) \cdot \left| \frac{\partial h_\lambda(z)}{\partial z} \right|^{-1}$   
 $h_\lambda = f_3 \circ f_2 \circ \sigma \circ f_1$ ,  $\lambda = (a, b, \alpha, \beta, \vartheta_0, \dots, \vartheta_M)$

$\text{loss} = -\text{ELBO}(\lambda) \xrightarrow{SGD} \lambda_{opt} \Rightarrow q_{\lambda_{opt}}$



# Multi-parameter models

# Mean-field approximation for multi-parameter-models

In **mean-field VI** we assume that we can model all variational distributions independently.

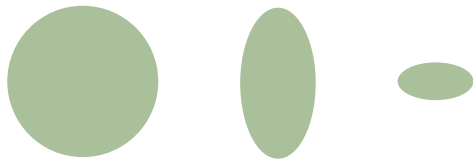
Hence the joint variational distribution is given by a product of marginal distributions:

$$q_{\lambda}(\mathbf{w}) = \prod_{k=1}^p q_{\lambda_k}(w_k)$$

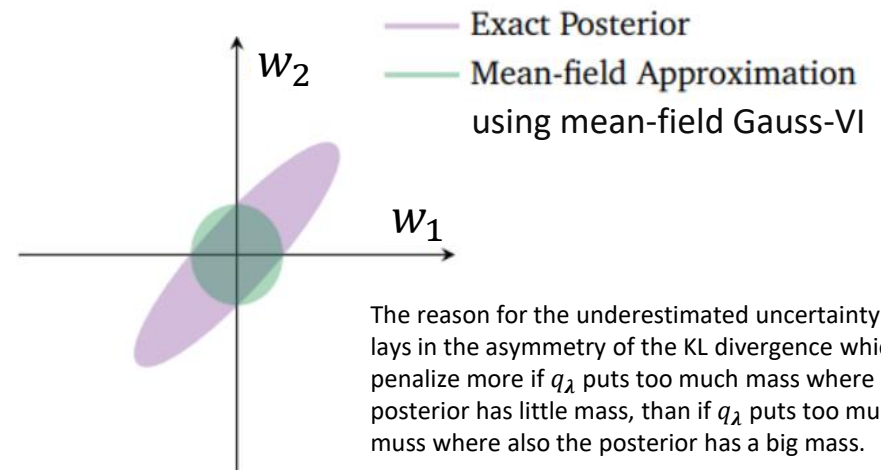
**Pros:** no need to model dependencies  
→ less parameters are needed

**Cons:** dependencies are ignored and  
uncertainty of posterior is underestimated

Possible bivariate Gaussians with mean-field



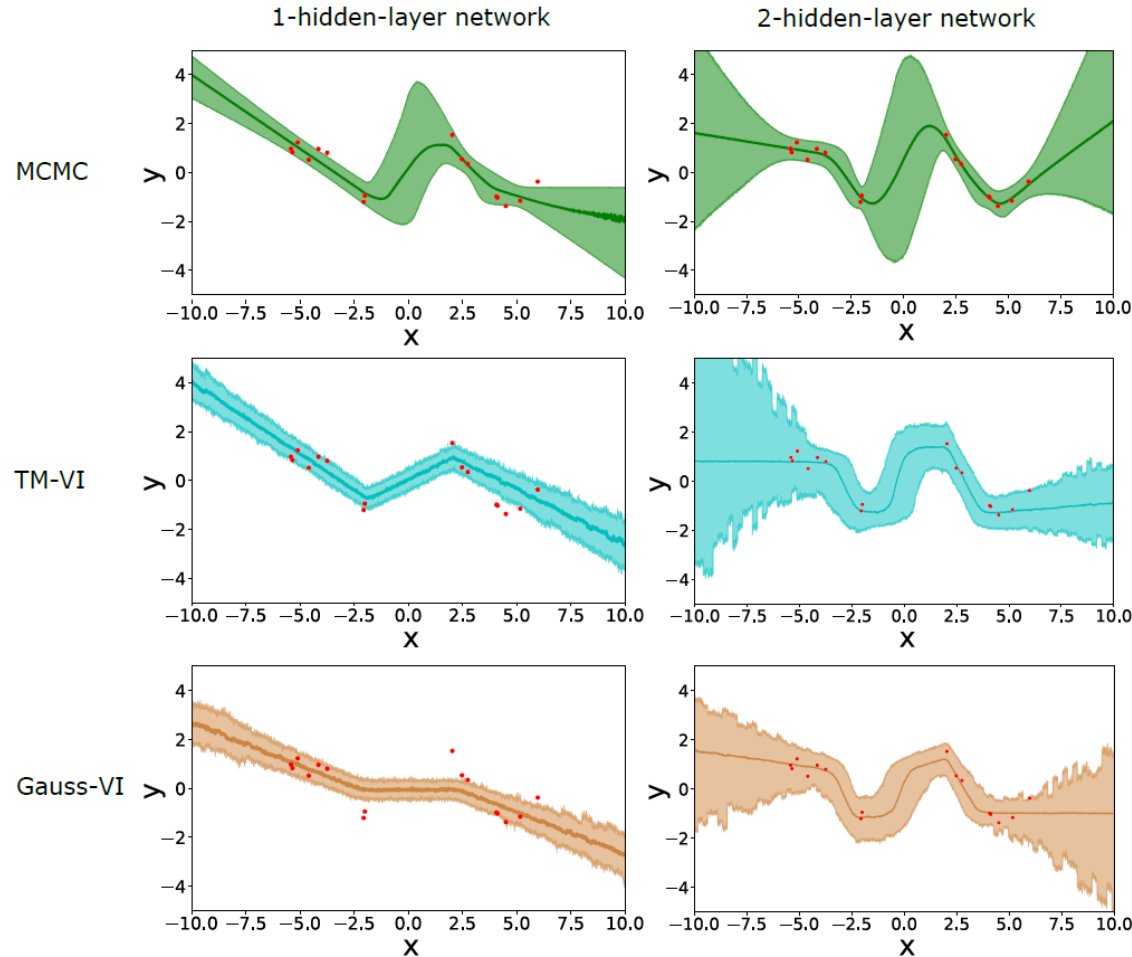
Impossible bivariate Gaussians with mean-field



<https://arxiv.org/abs/1601.00670>

# Mean-field VI for multi-parameter NN

We use Bayesian NNs to estimate the conditional mean  $\mu(x)$  of  $(y|x) \sim N(\mu(x), \sigma)$



Both VI-approaches underestimate the uncertainty. TM-VI can't leverage in mean-field.

Note: For Gaussian-VI it is known that mean-field does not hurt in deep NN <https://arxiv.org/abs/2002.03704>



## Conclusion and outlook

- VI allows for approximating posterior by an optimization process
- Gaussian-VI approximate posteriors by a Gaussian
- TM-VI allows for flexible posterior approximations
- In single parameter model TM-VI yields very accurate posterior approximations and outperforms Gaussian-VI by far if posteriors are complex
- In multi-parameter models mean-field VI is usually used
- Ignoring dependencies in mean-field VI destroys advantages of TM-VI

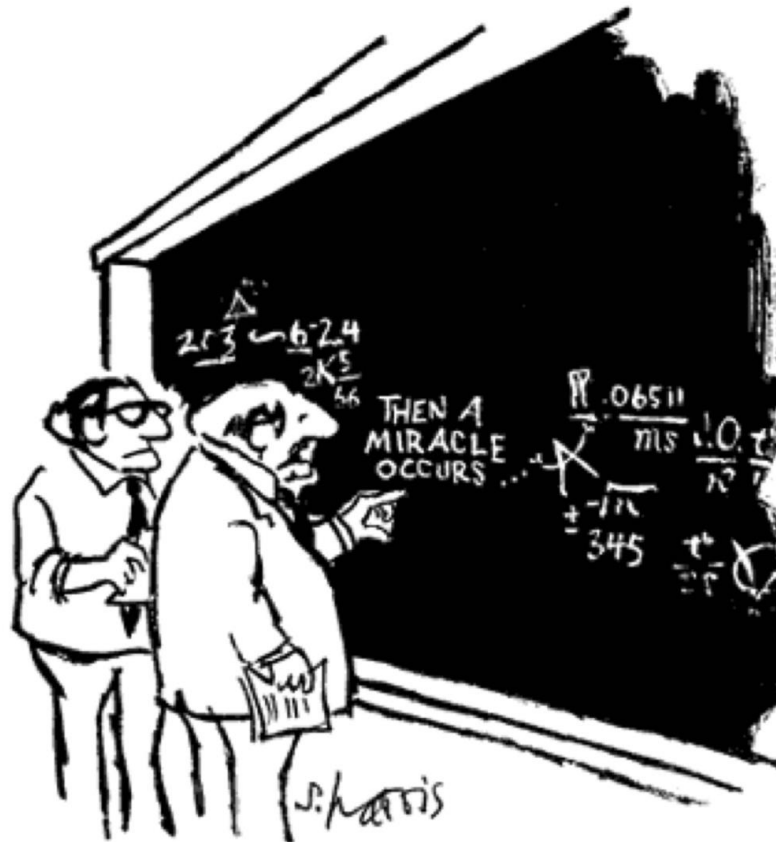
### Outlook:

- Go for TM-VI in few-parameter models w/o mean-field approximation
- Go for TM-VI in semi-structured models w and w/o mean-field approximation
- Setup TM-VI-Bayesian transformations models

# Appendix

## Calculating distance to an unknown function

- We need to calculate KL-Divergence between
  - $p(\theta|D)$  the **unknown posterior**
  - $q_\lambda(\theta)$  and the variational approximation
- Is this possible?




"I think you should be more explicit here in step two."

## Be more explicit about step two

$$KL[q_\lambda(\theta) \| p(\theta|D)] = \int q_\lambda(\theta) \log \frac{q_\lambda(\theta)}{p(\theta|D)} d\theta$$

We have to start with way, q first



$$p(\theta|D) = p(\theta, D)/p(D)$$

$$KL[q_\lambda(\theta) \| p(\theta|D)] = \int q_\lambda(\theta) \log \frac{q_\lambda(\theta)}{p(\theta, D)/p(D)} d\theta$$

$$\log(A \cdot B) = \log(A) + \log(B)$$

$$\log(B/A) = -\log(A/B)$$

$$KL[q_\lambda(\theta) \| p(\theta|D)] = \int q_\lambda(\theta) \log p(D) d\theta - \int q_\lambda(\theta) \log \frac{p(\theta, D)}{q_\lambda(\theta)} d\theta$$



no dependence on  $\theta$  and  $\int q_\lambda(\theta) d\theta = 1$


$$KL[q_\lambda(\theta) \| p(\theta|D)] = \log p(D) - \int q_\lambda(\theta) \log \frac{p(\theta, D)}{q_\lambda(\theta)} d\theta$$



We need to minimize

## Be more explicit about step two (cont'd)

$$\lambda^* = \operatorname{argmin}\left\{- \int q_\lambda(\theta) \log \frac{p(\theta, D)}{q_\lambda(\theta)} d\theta\right\}$$


$$p(\theta, D) = p(D|\theta) \cdot p(\theta)$$

$$\lambda^* = \operatorname{argmin}\left\{- \int q_\lambda(\theta) \log \frac{p(D|\theta) \cdot p(\theta)}{q_\lambda(\theta)} d\theta \right\}$$

$$\lambda^* = \operatorname{argmin}\left\{\int q_\lambda(\theta) \log \frac{q_\lambda(\theta)}{p(\theta)} d\theta - \int q_\lambda(\theta) \cdot \log p(D|\theta) d\theta \right\}$$

$$\lambda^* = \operatorname{argmin}\{KL[q_\lambda(\theta)||p(\theta)] - E_{\theta \sim q_\lambda}[\log(p(D|\theta))]\}$$

A miracle the unknown posterior  $p(\theta|D)$  is gone.