Taking the square

Information maximising non-linear data compression

Tom Charnock

Outline

Reinforcement learning(ish)

Statistics

Neural network

Taking the square

Lyman-α constraints (Maybe)

Reinforcement learning

Supervised learning

Used to find a specific map of an input to a known output

Unsupervised learning

Used to find hidden structure from inputs without knowing the output

Reinforcement learning

Used to find a specific function without knowing the output

Uses a *reward* function

Test model - how to summarise Gaussian noise

Unknown variance θ, Gaussian noise ($n_{\mathbf{d}} = 10$)

$$\mathbf{d} = \left\{ d_i \curvearrowleft \mathcal{N}(0, \theta) \middle| \; i = 1 \rightarrow n_{\mathbf{d}} \right\}$$

Analytic likelihood

$$-2 \ln \mathcal{L}(\mathbf{d}|\theta) = \frac{1}{\theta} \sum_{i=1}^{n_{\mathbf{d}}} d_i^2 + n_{\mathbf{d}} \ln [2\pi\theta]$$

Single sufficient statistic

$$T = \sum_{i=1}^{n_{\mathbf{d}}} d_i^2$$

Fisher information

Fisher information matrix

$$\mathbf{F}_{\alpha\beta}(\theta) = -\left\langle \frac{\partial^2 \ln \mathcal{L}(\mathbf{d}|\theta)}{\partial\theta_\alpha \partial\theta_\beta} \right\rangle \Bigg|_{\theta=\theta_{\rm fid}}$$

Analytic likelihood

$$-2\ln\mathcal{L}(\mathbf{d}|\theta) = \frac{1}{\theta}\sum_{i=1}^{n_{\mathbf{d}}} d_i^2 + n_{\mathbf{d}} \ln\left[2\pi\theta\right]$$

Fisher information with $\theta_{\rm fid} = 1$ and $n_{\mathbf{d}} = 10$

$$\begin{aligned}\mathbf{F}_{\alpha\beta}(\theta) &= \frac{n_{\mathbf{d}}}{2\theta_{\rm fid}} \\ &= 5\end{aligned}$$

Gaussian likelihoods

$$-2\ln\mathcal{L}\left(\mathbf{d}|\theta\right)=\left(\mathbf{d}-\mu(\theta)\right)^{T}\mathbf{C}^{-1}\left(\mathbf{d}-\mu(\theta)\right)+n_{\mathbf{d}}\ln|2\pi\mathbf{C}|$$

Fisher information

$$\mathbf{F}_{\alpha\beta}(\theta)=\frac{1}{2}\mathrm{Tr}\left[\frac{\partial\mu(\theta)}{\partial\theta_{\alpha}}^{T}\mathbf{C}^{-1}\frac{\partial\mu(\theta)}{\partial\theta_{\beta}}+\frac{\partial\mu(\theta)}{\partial\theta_{\beta}}^{T}\mathbf{C}^{-1}\frac{\partial\mu(\theta)}{\partial\theta_{\alpha}}\right]$$

Assuming the covariance is independent of the parameters

MOPED algorithm $\overline{d}_{\alpha}=\mathbf{r}_{\alpha}^{T}\mathbf{d}$

Each $\mathbf{r}_{\alpha}$ orthogonal such that $\overline{d}_{\alpha}$ are uncorrelated.

$$\mathbf{r}_{1}=\frac{\mathbf{C}^{-1}\boldsymbol{\mu}_{,1}}{\sqrt{\boldsymbol{\mu}^{T}{}_{,1}\,\mathbf{C}^{-1}\boldsymbol{\mu}_{,1}}}\qquad\mathbf{r}_{\alpha}=\frac{\mathbf{C}^{-1}\boldsymbol{\mu}_{,\alpha}-\sum_{\beta=1}^{\alpha-1}\left(\boldsymbol{\mu}^{T}{}_{,\alpha}\,\mathbf{r}_{\beta}\right)\mathbf{r}_{\beta}}{\sqrt{\boldsymbol{\mu}^{T}{}_{,1}\,\mathbf{C}^{-1}\boldsymbol{\mu}_{,1}-\sum_{\beta=1}^{\alpha-1}\left(\boldsymbol{\mu}^{T}{}_{,\alpha}\,\mathbf{r}_{\beta}\right)^{2}}}$$

with $,_{\alpha}\equiv\partial/\partial\theta_{\alpha}$, $\boldsymbol{\mu}=\mu(\theta)$ and $\mathbf{C}$ and $\boldsymbol{\mu}$ evaluated at fiducial parameter values.

The Fisher information is conserved under this transformation.

Less pleasant likelihoods

Unspecified function of the data $f(\mathbf{d})$

$$-2\ln\mathcal{L}\left(\mathbf{d}|\theta\right)=\left(f(\mathbf{d})-\boldsymbol{\mu}_f\right)^T\mathbf{C}_f^{-1}\left(f(\mathbf{d})-\boldsymbol{\mu}_f\right)$$

with mean and covariance over some simulations $\mathbf{s}_i$

$$\boldsymbol{\mu}_f=\frac{1}{n_\mathbf{s}}\sum_{i=1}^{n_\mathbf{s}}f(\mathbf{s}_i)$$

$$\mathbf{C}_f=\frac{1}{n_\mathbf{s}-1}\sum_{i=1}^{n_\mathbf{s}}\left(f(\mathbf{s}_i)-\boldsymbol{\mu}_f\right)\left(f(\mathbf{s}_i)-\boldsymbol{\mu}_f\right)$$

New Fisher information matrix

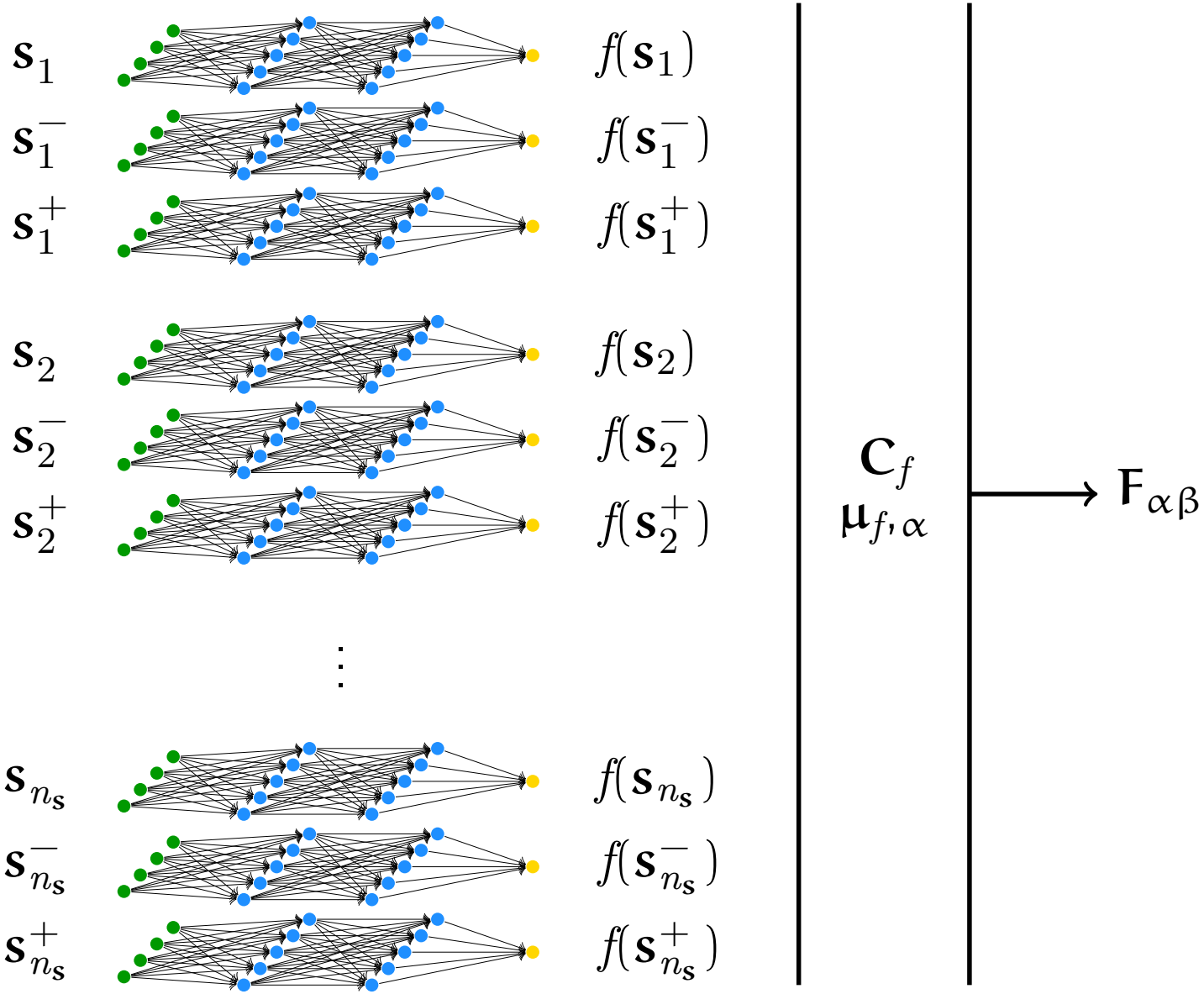
$$\mathbf{F}_{\alpha\beta}=\frac{1}{2}\mathrm{T}r\left[\boldsymbol{\mu}_f^T{}_{,\alpha}\,\mathbf{C}_f^{-1}\boldsymbol{\mu}_{f,\beta}+\boldsymbol{\mu}_f^T{}_{,\beta}\,\mathbf{C}_f^{-1}\boldsymbol{\mu}_{f,\alpha}\right]$$

Reward function

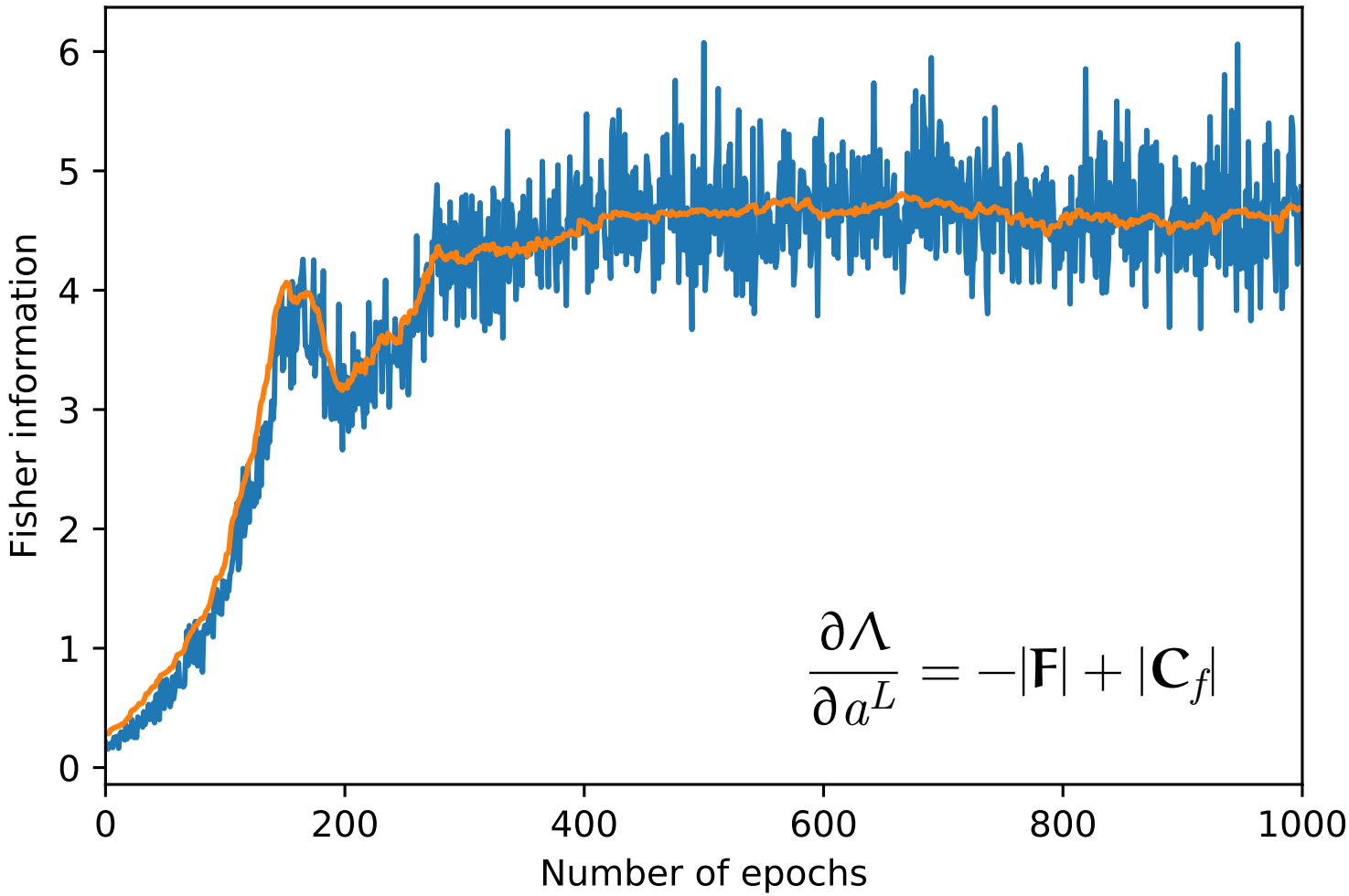
Reward function

Let's maximise the Fisher information!

Information maximising neural network



Test model - training



[96, 96], 0.5 dropout, leaky ReLU with $\alpha = 0.1$, $\eta = 0.01$, $\mathbf{b}^l_{\rm init} = 0.1$ $\boldsymbol{w}^l_{\rm init} = \mathcal{N}(0, \sqrt{2}/\kappa^{l-1})$,
1000 simulations, 200 derivatives, 2 batches, 1000 epochs

Approximate Bayesian computation (ABC)

Get summary of real data $f(\mathbf{d})$

Create simulation $\mathbf{s}_i(\theta)$ within prior range of θ

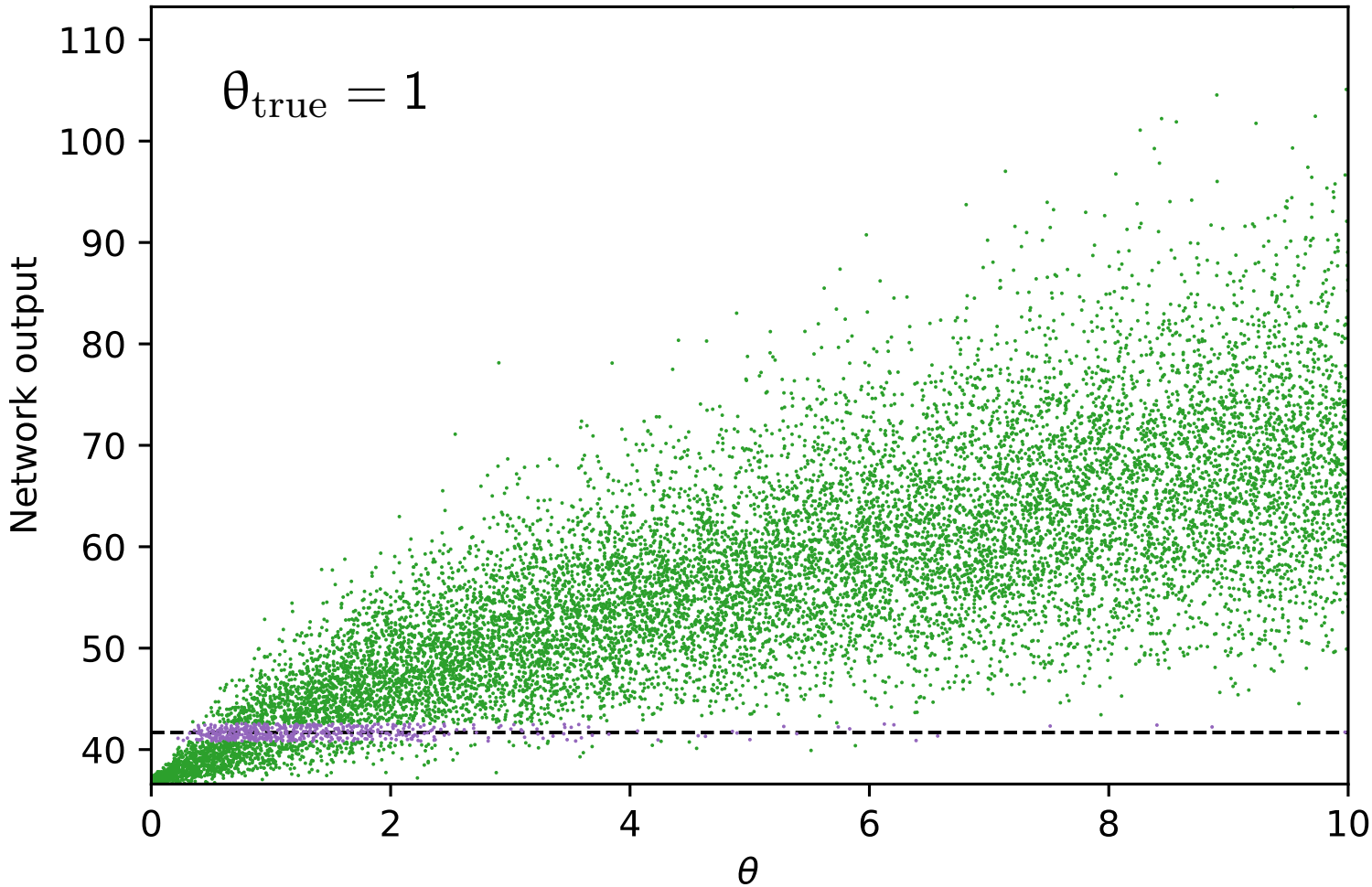
Calculate $f(\mathbf{s}_i(\theta))$

Accept point if

$$\rho = \sqrt{(f(\mathbf{s}_i(\theta)) - f(\mathbf{d}))^2} < \varepsilon$$

Actually use PMC rather than random draws from the prior

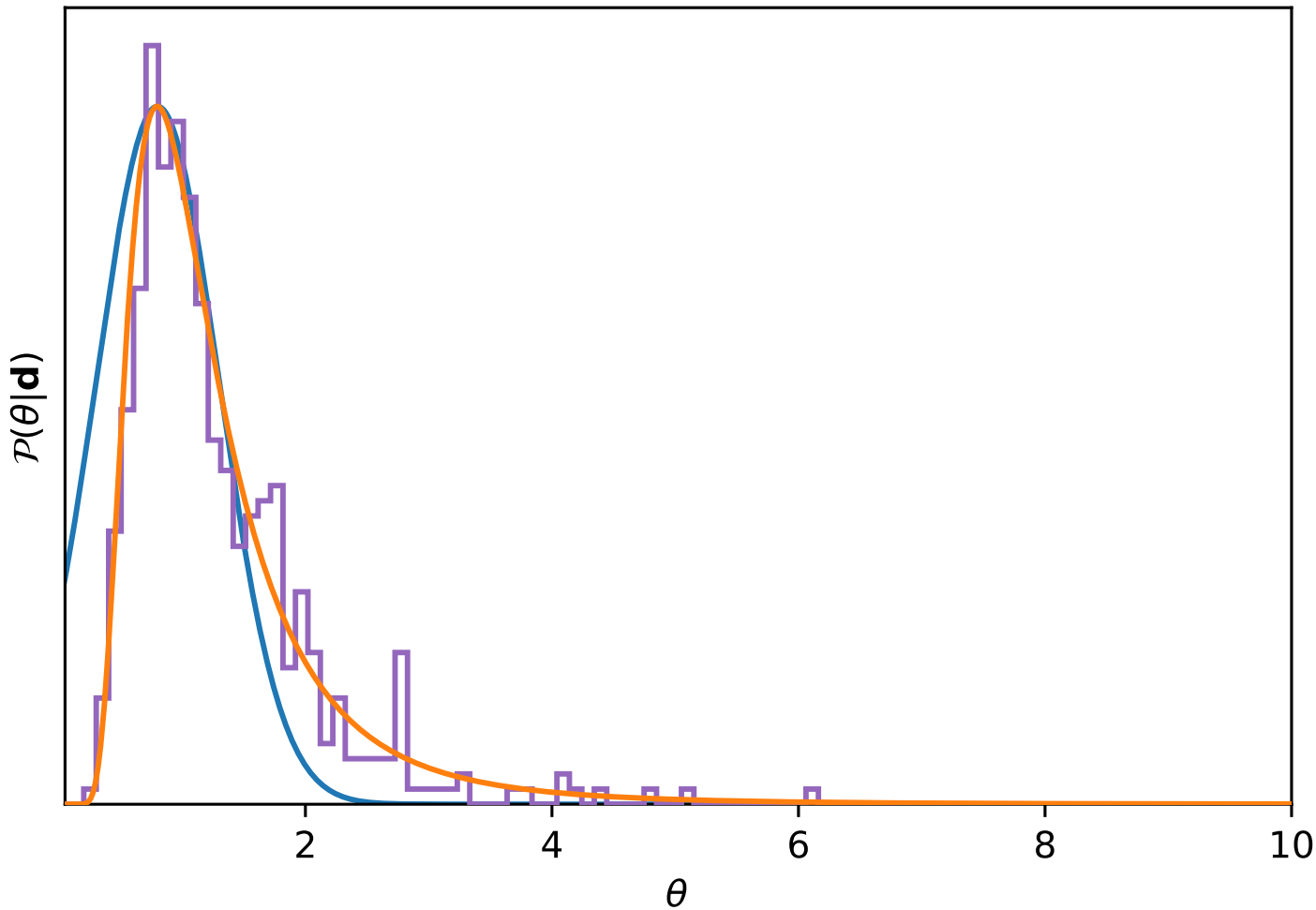
Network output



Posterior distribution for the variance, $P(\theta|\mathbf{d})$

$$\mathbf{d} = \left\{ d_i \curvearrowleft \mathcal{N}(0, \theta) \middle| \; i = 1 \rightarrow n_{\mathbf{d}} \right\}$$

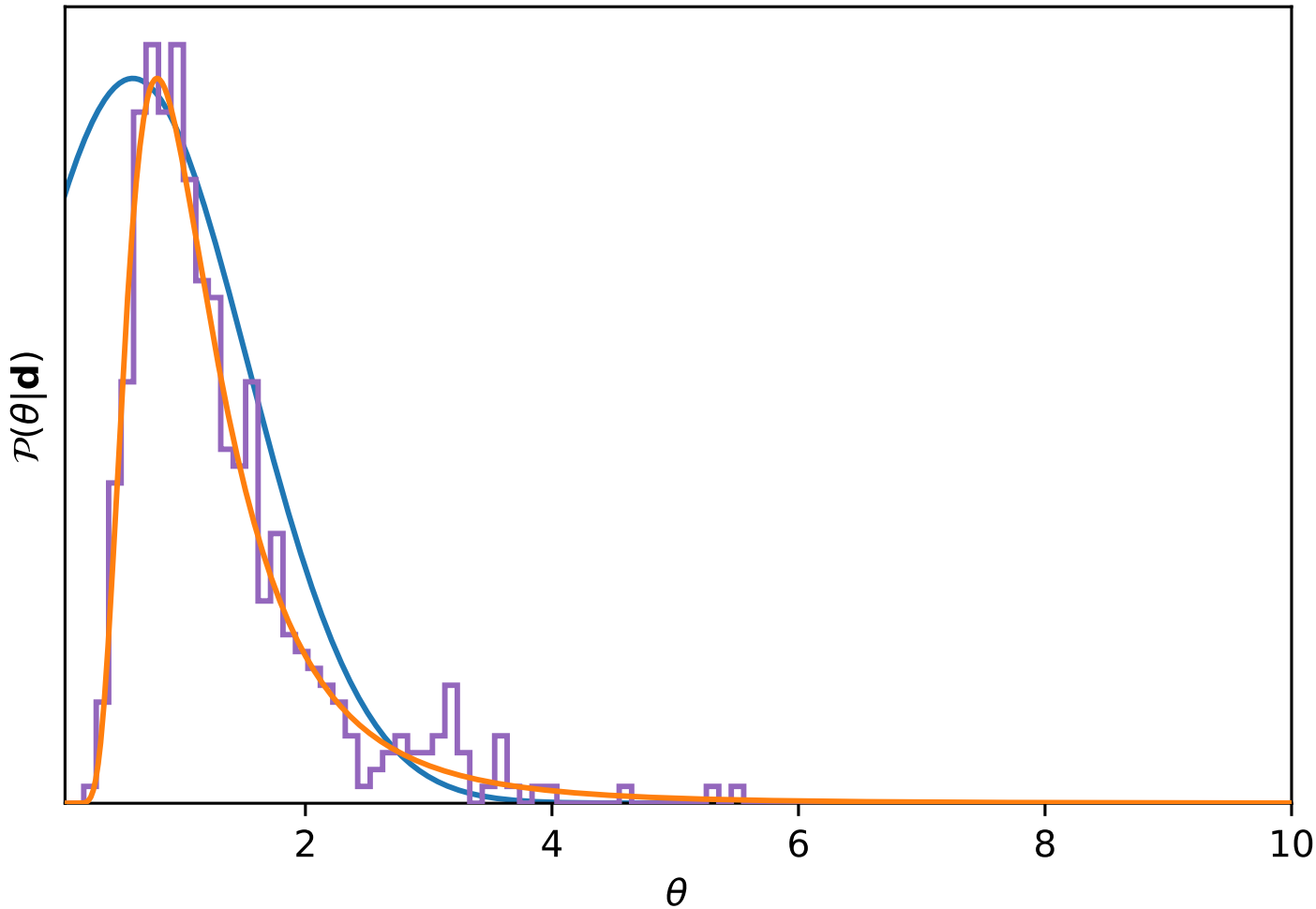
$$\theta_{\rm fid} = 1$$



Posterior distribution for the variance, $P(\theta|\mathbf{d})$

$\mathbf{d} = \left\{ d_i \curvearrowleft \mathcal{N}(0, \theta) \middle| \; i = 1 \rightarrow n_{\mathbf{d}} \right\}$

$\theta_{\rm fid} = 2$

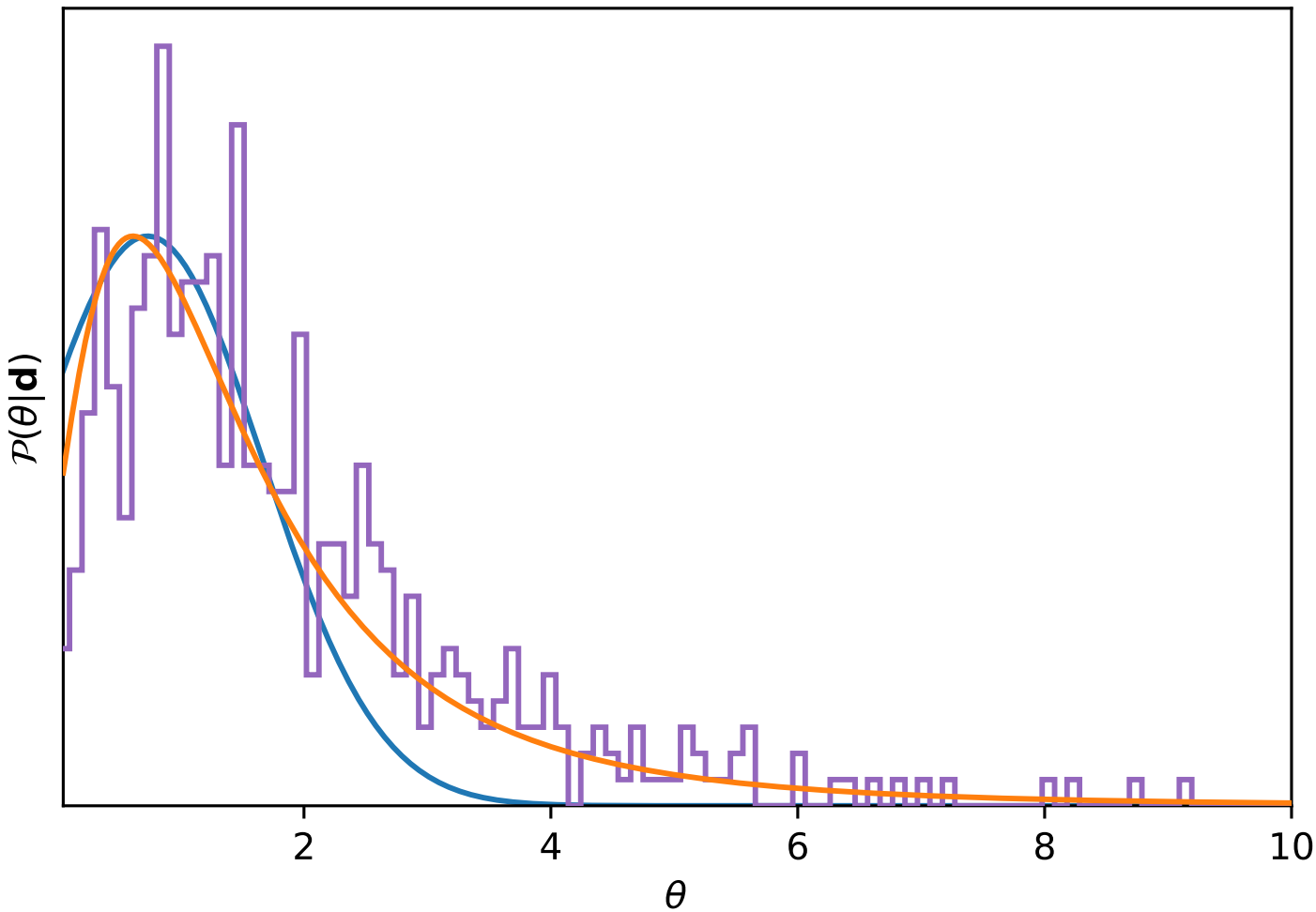


Beyond the square

Posterior distribution for the variance, $P(\theta|\mathbf{d})$

$$\mathbf{d} = \left\{ d_i \curvearrowleft \mathcal{N}(0, \theta + \sigma^2_{\rm noise}) \middle| \; i = 1 \rightarrow n_{\mathbf{d}} \right\}$$

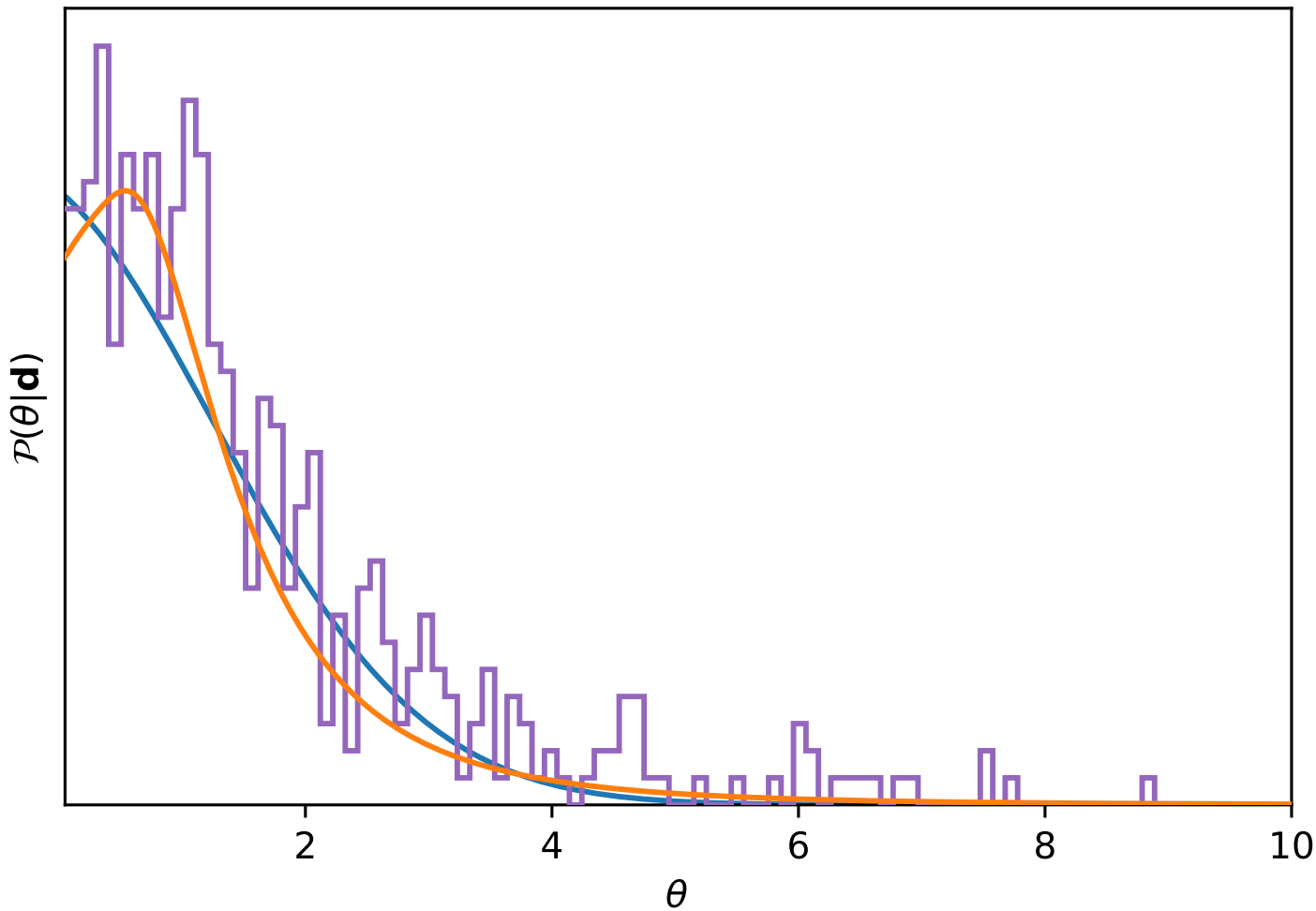
$$\theta_{\rm fid} = 1, \; \sigma^2_{\rm noise} = 1$$



Posterior distribution for the variance, $P(\theta|\mathbf{d})$

$$\mathbf{d} = \left\{ d_i \curvearrowleft \mathcal{N}(0, \theta + \sigma^2_{\rm noise}) \middle| \; i = 1 \rightarrow n_{\mathbf{d}} \right\}$$

$$\theta_{\rm fid} = 1, \; \sigma^2_{\rm noise} = [0, 2]$$



Information maximising non-linear data compression

- Linear data compression can find summaries of Gaussian likelihood (and approximations thereof).
- Non-linear data compression can find summaries of any data where a function of the data can be written like a Gaussian.
- The function of the data which maximises the Fisher information is the function which best summarises the data - although this function is unknown.
- A neural network can be trained (using relatively few training samples) to find the function by maximising the Fisher information as the loss function.
- This function can now be used for ABC to find $P(\theta|\mathbf{d})$.

Further beyond the square

Lyman-α

Generate quasar spectrum with Lyman-α absorption

Power spectrum of flux absorbing neutral hydrogen

$$P_{\rm F}^{\rm 1D}(k,z) = b_\delta^2 (1+\beta\mu)^2 D_{\rm F}(k,\mu) \left(\frac{1+z}{2.25}\right)^\alpha P^{\rm 1D}(k)$$

Random Gaussian field in 1D Fourier space

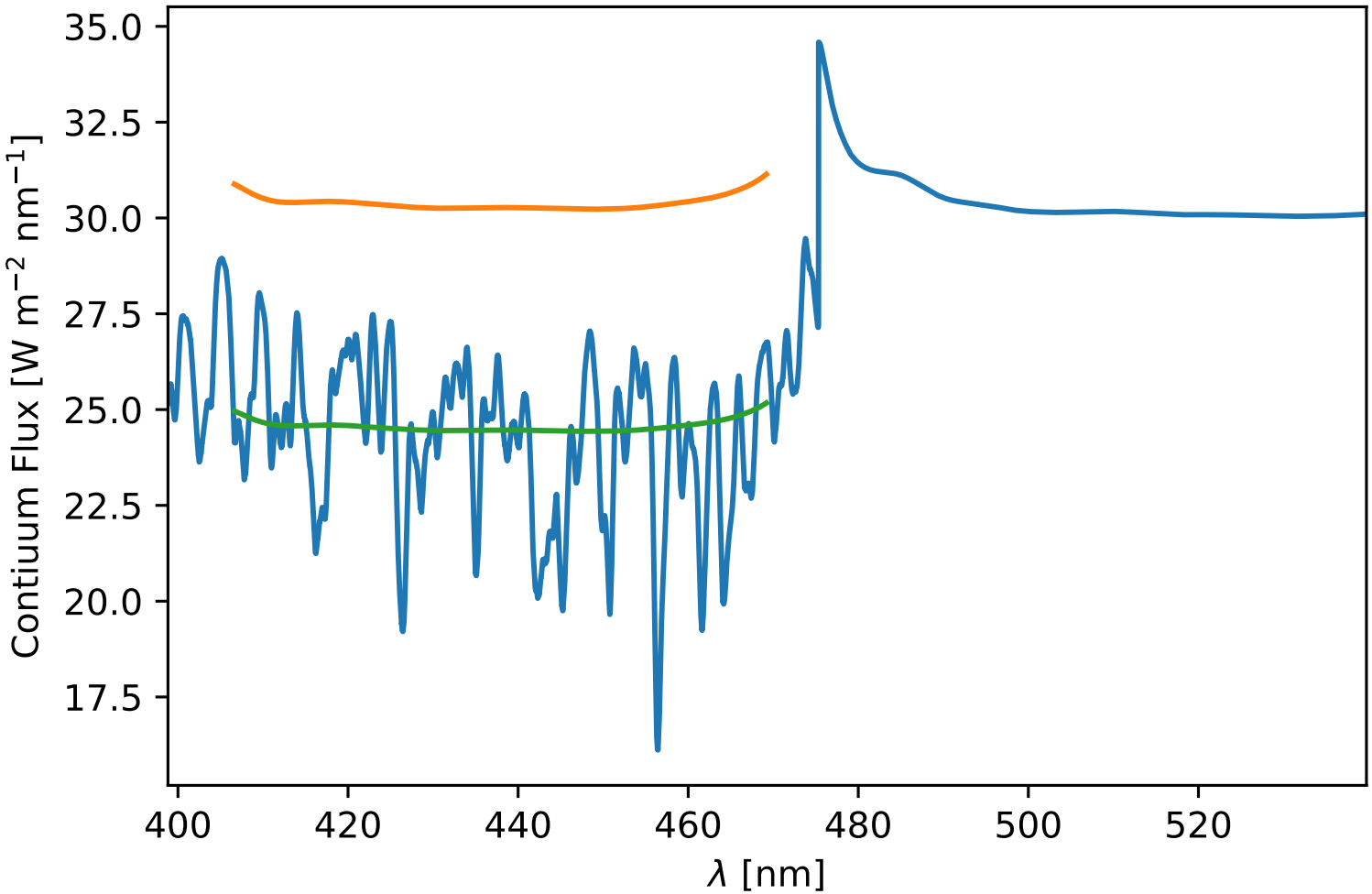
Calculate optical depth

$$\tau = 1.54 \left(\frac{T_0}{10^4{\rm K}}\right)^{-0.7} \frac{10^{-12}{\rm s}^{-1}}{\Gamma_{\rm UV}} \left(\frac{1+z}{1+3}\right)^6 \times \frac{0.7}{h} \left(\frac{\Omega_{\rm b}h^2}{0.02156}\right)^2 \frac{4.0927}{H(z)/H_0} \rho^{2-0.7\gamma}$$

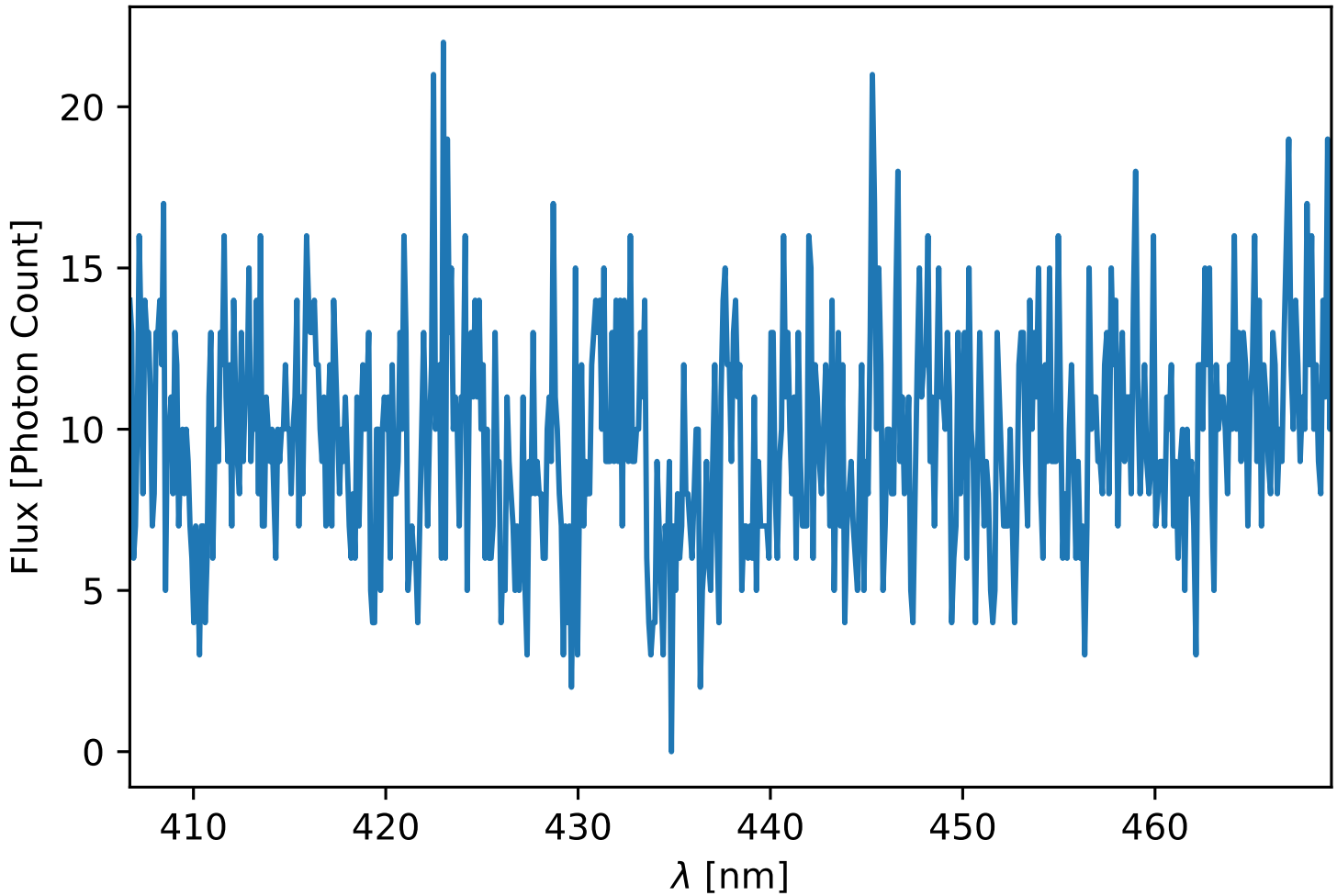
Calculate continuum flux from PCA

$$r(\lambda) = \mu(\lambda) + \sum_i c_i(\lambda)\xi_i(\lambda)$$

Continuum modulated quasar spectrum



Input spectrum



Constrain $A_{\rm s}$

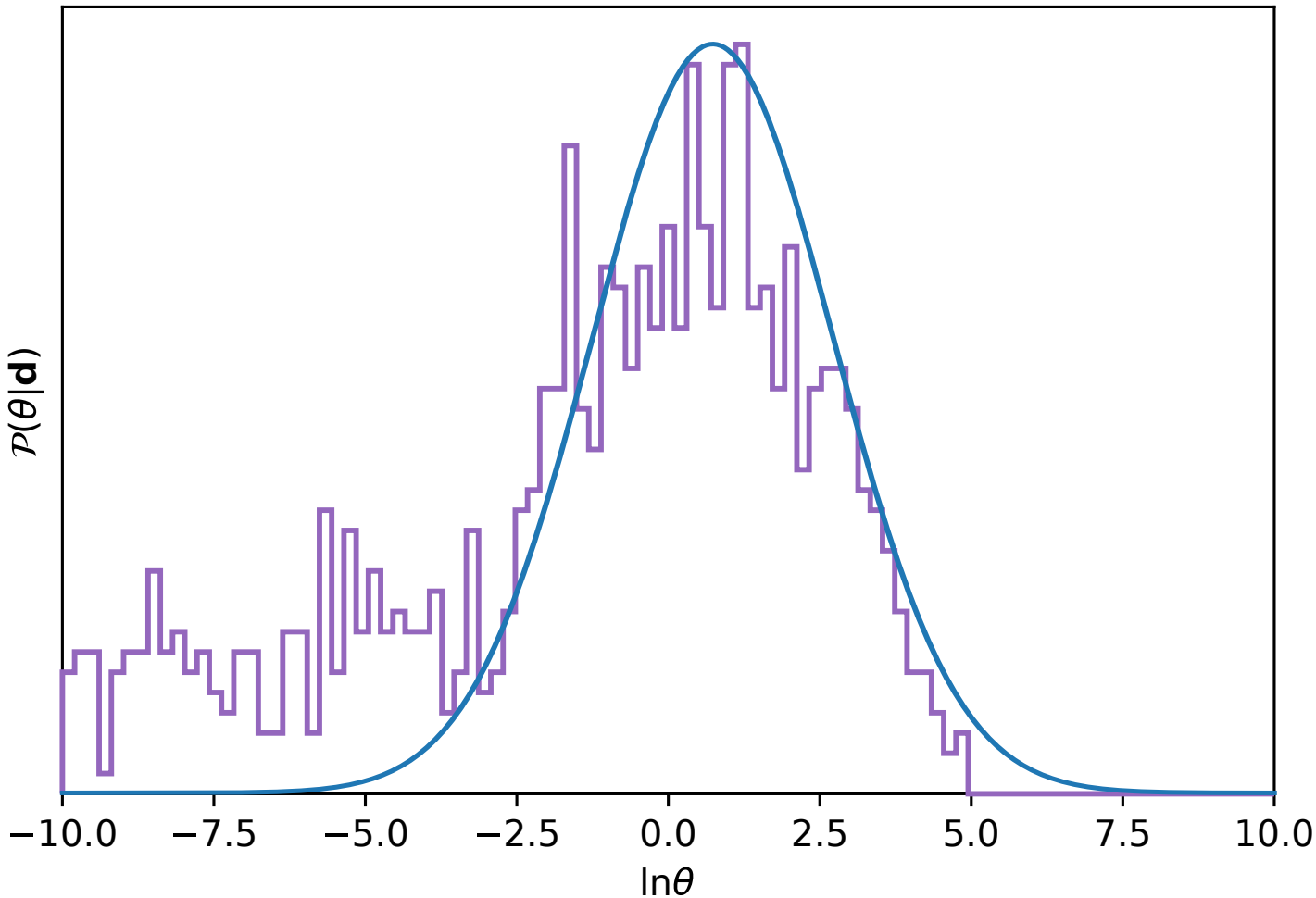
Amplitude of scalar perturbations, $A_{\rm s}$

$$A_{\rm s} = \theta A_{\rm init}$$

Scale the power spectrum

$$P_{\rm F}^{\rm 1D}(k, z) = \theta P_{\rm F,init}^{\rm 1D}(k, z)$$

Posterior distribution for $A_{\rm s} = \theta A_{\rm init}$, $P(\theta|\mathbf{d})$



Posterior distribution for $A_{\rm s} = \theta A_{\rm init}$, $P(\theta|\mathbf{d})$ when there are 10 quasars