Predictive Complexity Priors

Eric Nalisnick

Bayes Theorem

$$p(\boldsymbol{\theta} \mid \mathbf{Y}) = \frac{p(\mathbf{Y} \mid \boldsymbol{\theta})\; p(\boldsymbol{\theta})}{p(\mathbf{Y})}$$

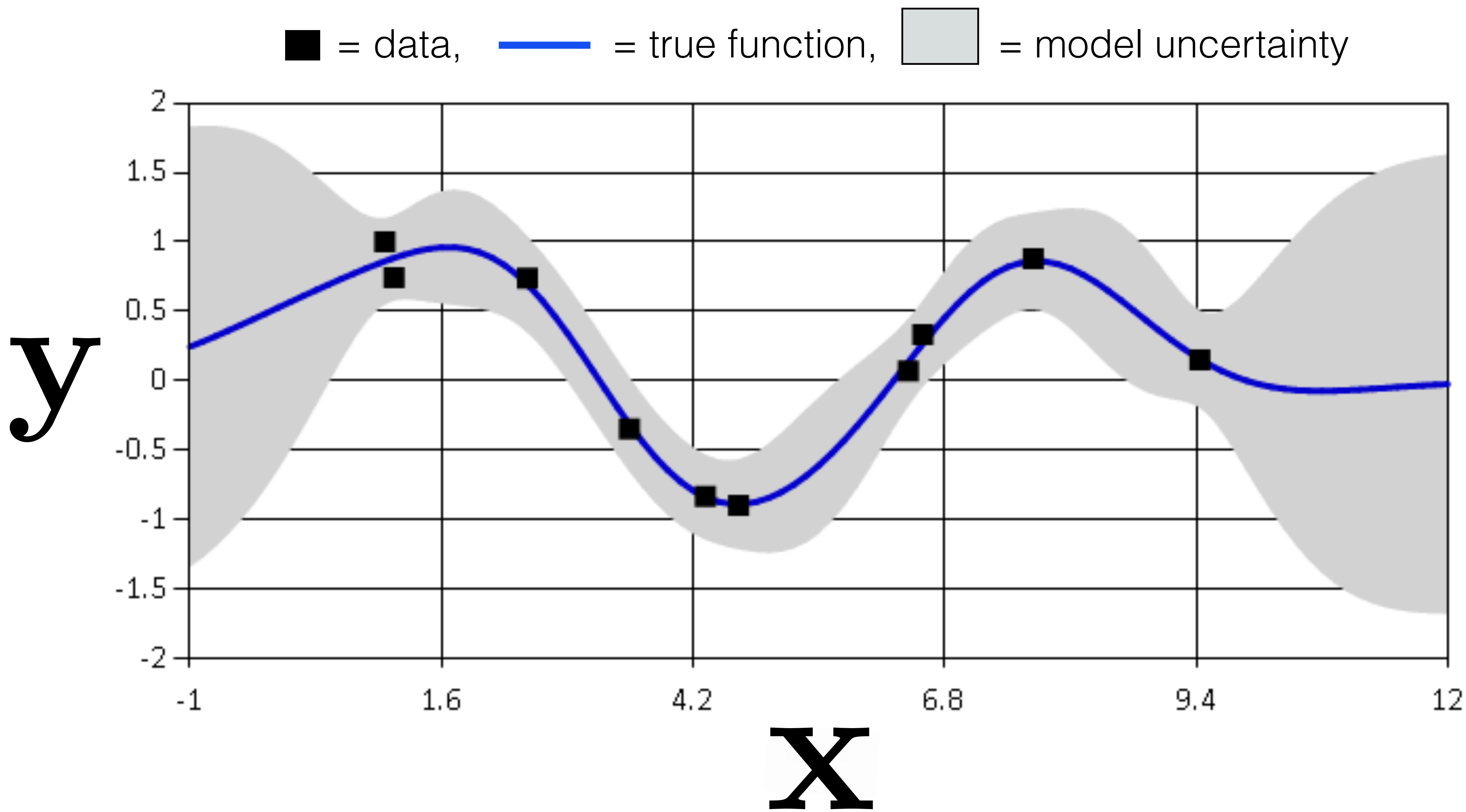
$\mathbf{Y}$ = data $\qquad$ $\boldsymbol{\theta}$ = model parameters

Bayes Theorem

$$p(\boldsymbol{\theta} \,|\, \mathbf{Y}) = \frac{p(\mathbf{Y} \,|\, \boldsymbol{\theta}) \; p(\boldsymbol{\theta})}{p(\mathbf{Y})}$$

Garbage in: *arbitrary priors*
Garbage out: *uncontrollable error bars*

Michael I. Jordan, *MLSS* (2017)


= data,
= true function,
= model uncertainty
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-1
1.6
4.2
6.8
9.4
12
y
x

What are good priors for Bayes neural nets?

Current Priors for Bayes NNs

Current Priors for Bayes NNs

Shrinkage Priors on Weights

$$\theta \sim \mathrm{N}(0,\ \tau)$$

Current Priors for Bayes NNs

Shrinkage Priors on Weights

$$\theta \sim \mathrm{N}(0, \tau)$$

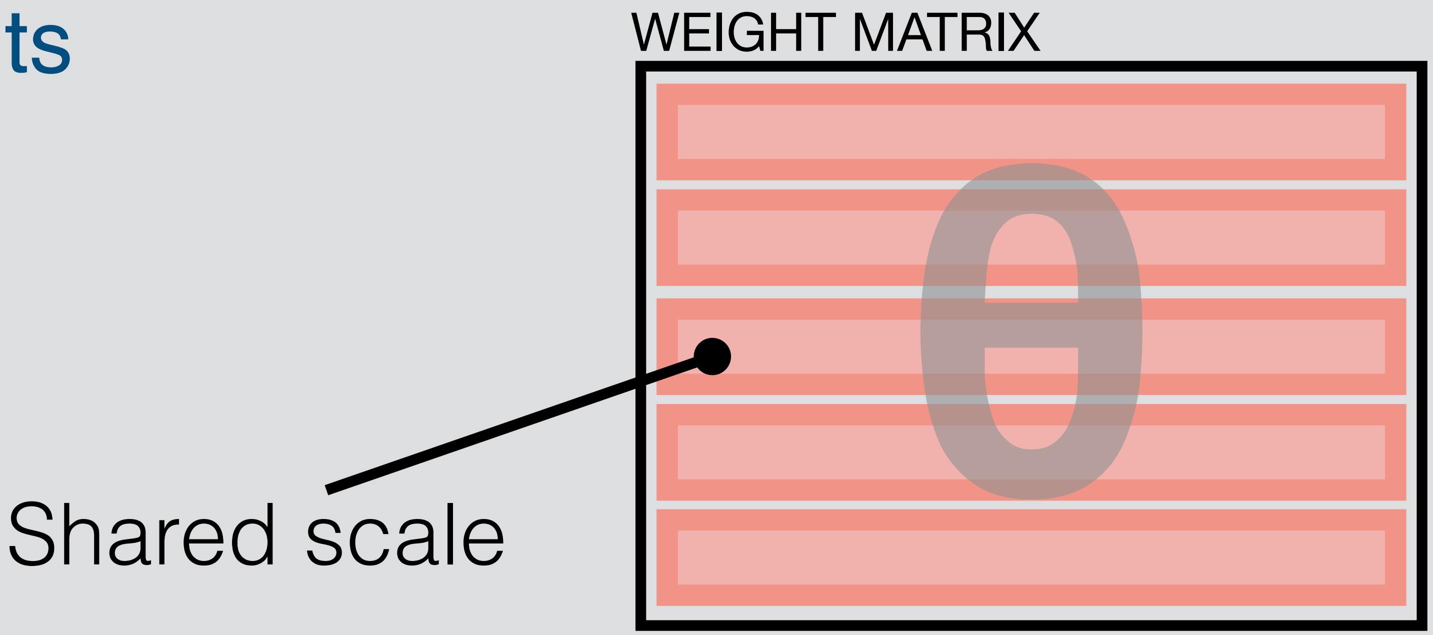
$$\tau \sim p(\tau)$$

Current Priors for Bayes NNs

Shrinkage Priors on Weights

$$\theta \sim \mathrm{N}(0, \tau)$$

$$\tau \sim p(\tau)$$

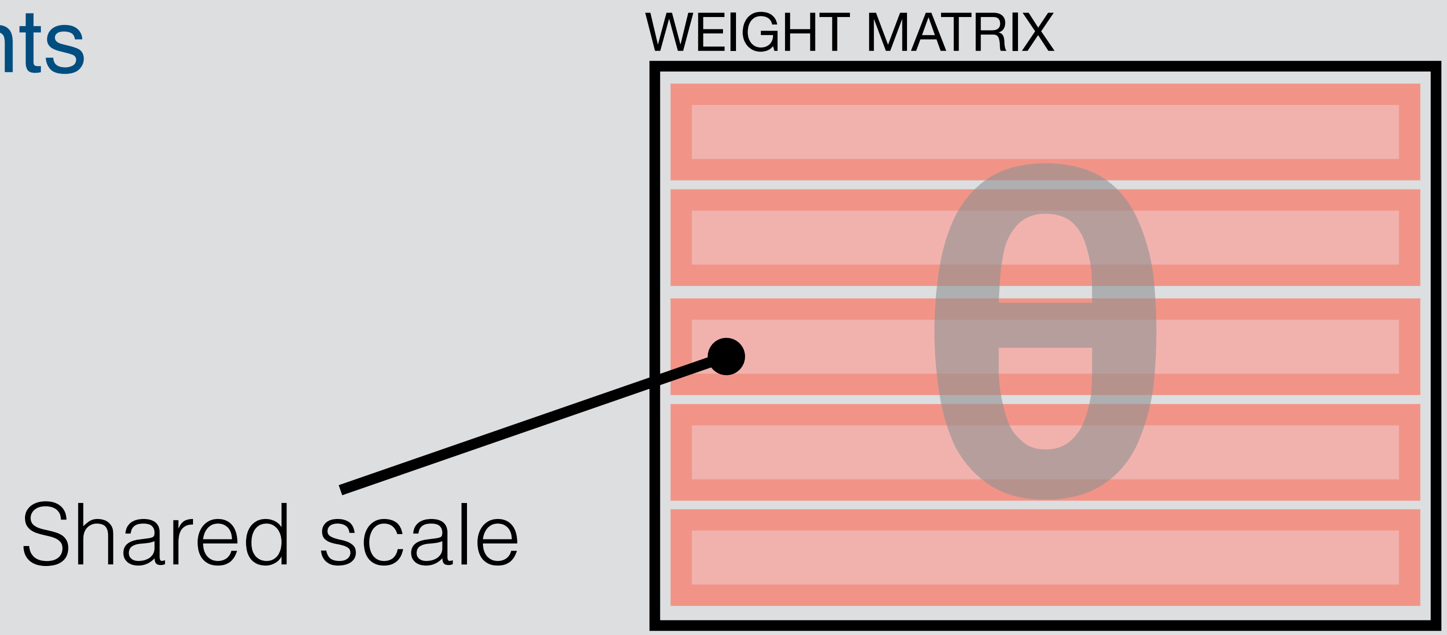


Current Priors for Bayes NNs

Shrinkage Priors on Weights

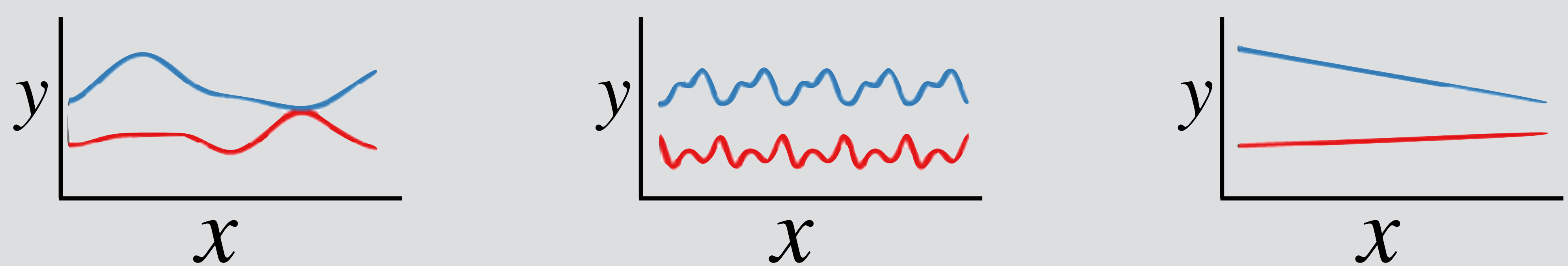
$$\theta \sim \mathrm{N}(0,\ \tau)$$

$$\tau \sim p(\tau)$$



Nonparametric Priors on NN's Function

$$f \sim \mathrm{GP}$$

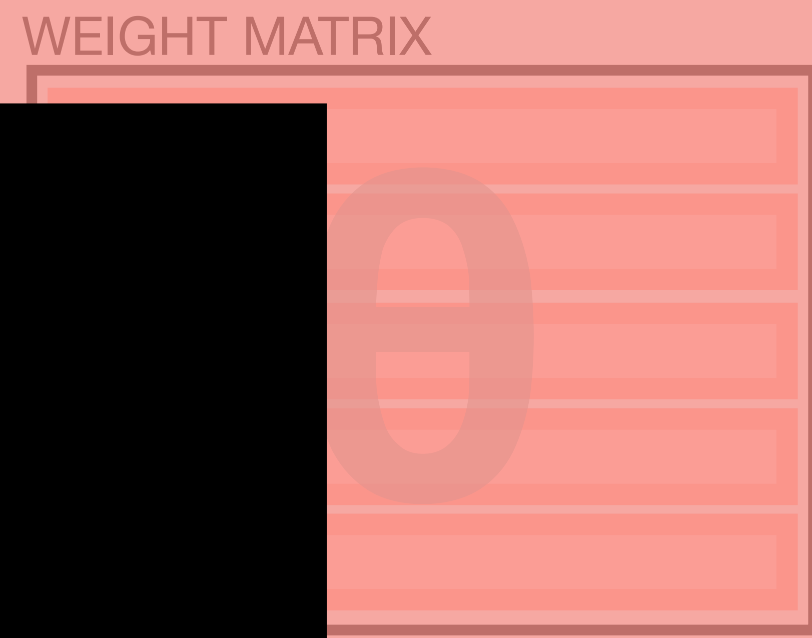


Current Priors for Bayes NNs

Shrinkage Priors on Weights

$\theta \sim \mathrm{N}(0$

$\tau \sim p(\tau$

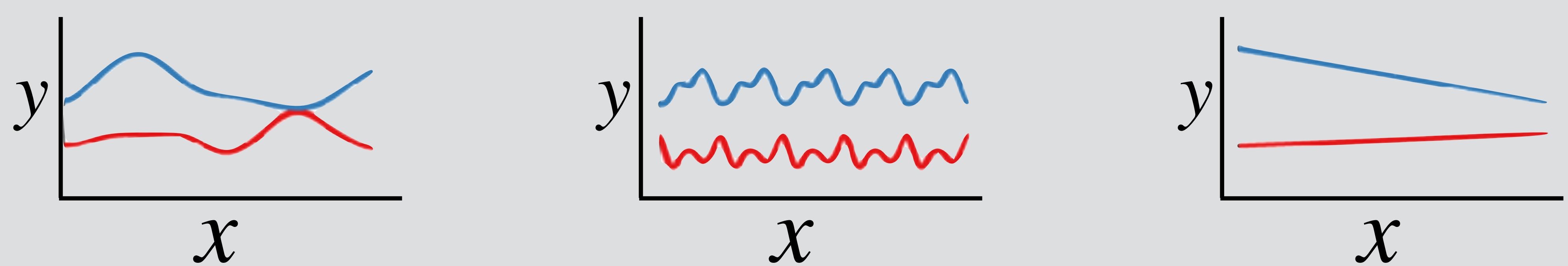


Computation:

Encoding beliefs:

Nonparametric Priors on NN's Function

$$f \sim \mathrm{GP}$$



Current Priors for Bayes NNs

Shrinkage Priors on Weights

$\theta \sim \mathrm{N}(0,$

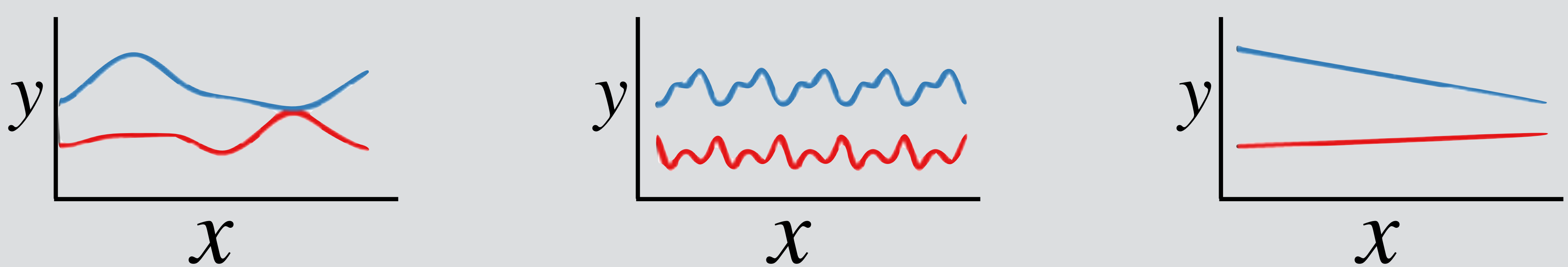
$\tau \sim p(\tau)$

Computation:

Encoding beliefs:

Nonparametric Priors on NN's Function

$f \sim \mathrm{GP}$

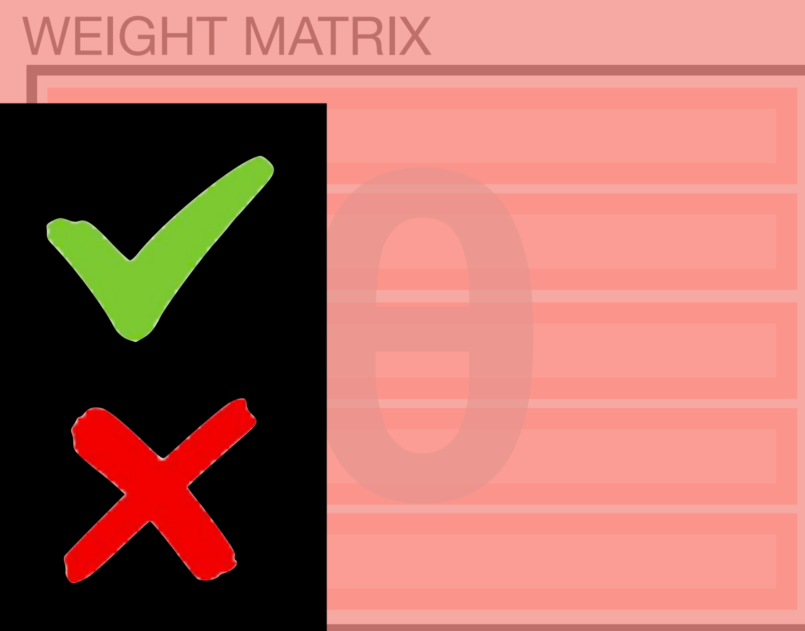


Current Priors for Bayes NNs

Shrinkage Priors on Weights

$\theta \sim \mathrm{N}(0,$

$\tau \sim p(\tau)$

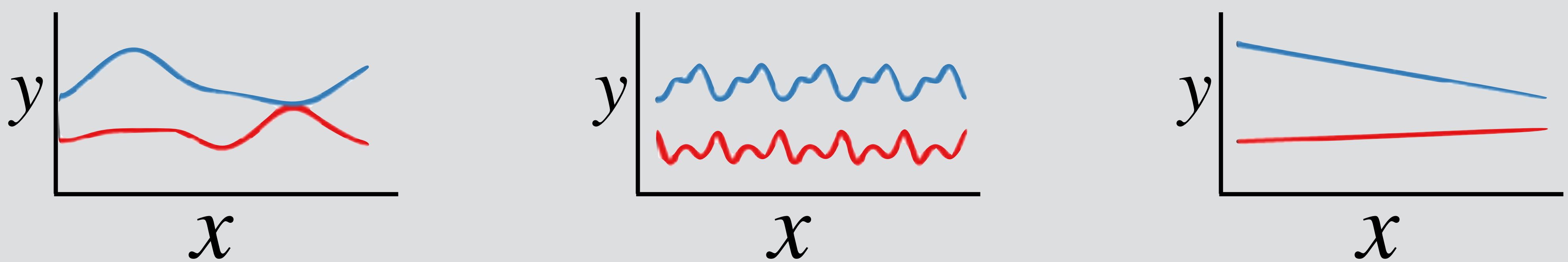


Computation: ✔

Encoding beliefs: ✘

Nonparametric Priors on NN's Function

$$f \sim \mathrm{GP}$$

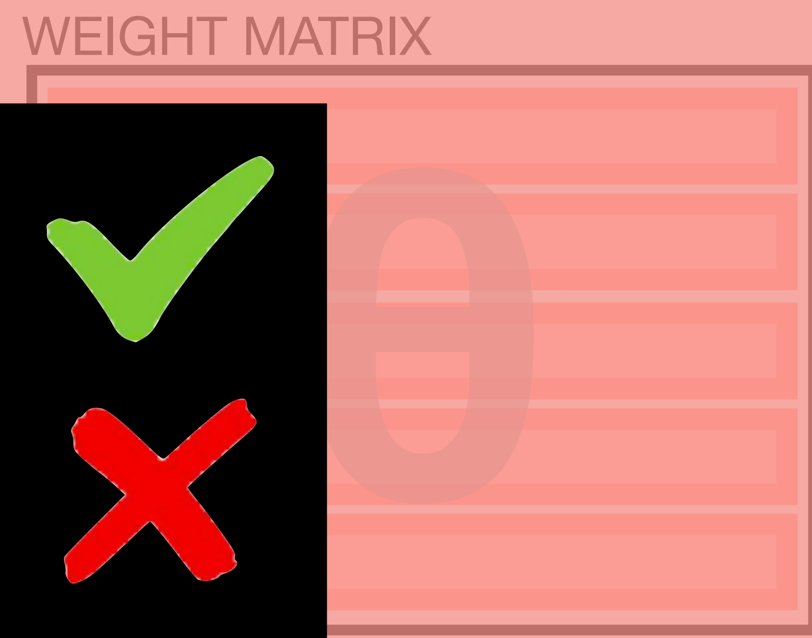


Current Priors for Bayes NNs

Shrinkage Priors on Weights

$\theta \sim \mathrm{N}(0$

$\tau \sim p(\tau)$



Computation:

Encoding beliefs:

Nonparametric Priors on NN's Function

Computation:

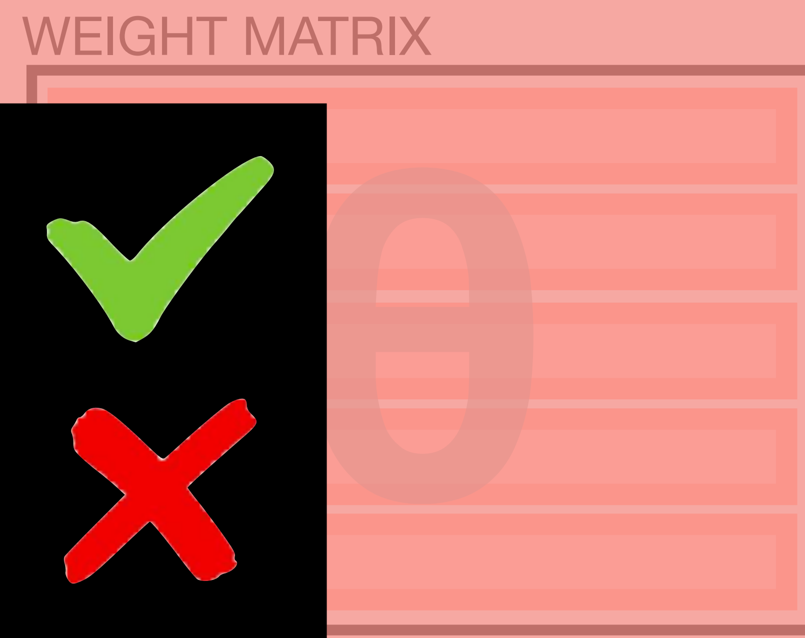
Encoding beliefs:

Current Priors for Bayes NNs

Shrinkage Priors on Weights

$\theta \sim \mathrm{N}(0,$

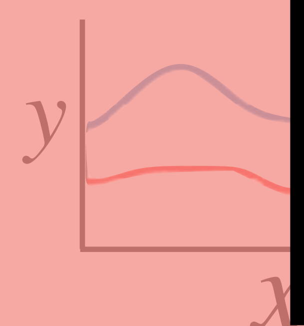
$\tau \sim p(\tau)$

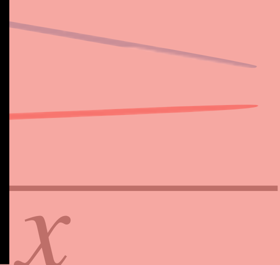


Computation: ✓

Encoding beliefs: ✗

Nonparametric Priors on NN's Function





Computation: ✗

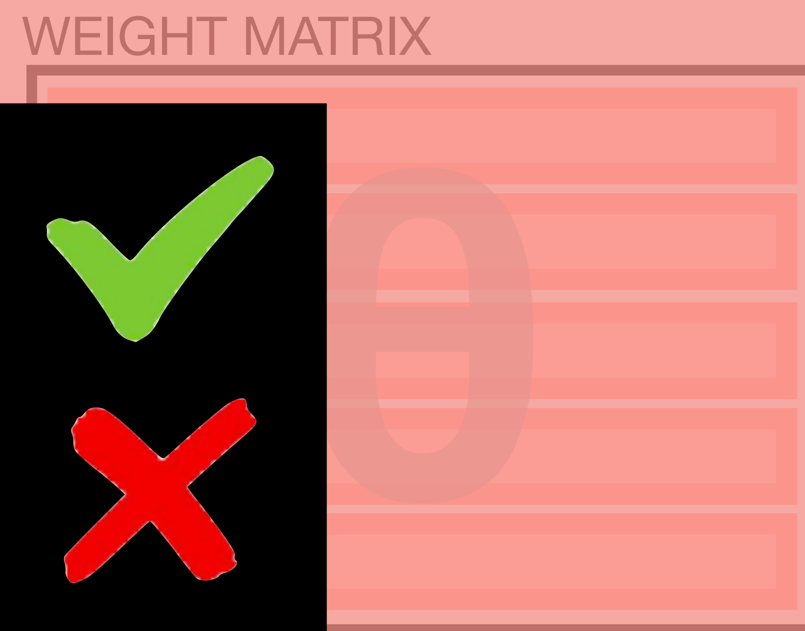
Encoding beliefs:

Current Priors for Bayes NNs

Shrinkage Priors on Weights

$\theta \sim \mathrm{N}(0$

$\tau \sim p(\tau)$



Computation: ✓

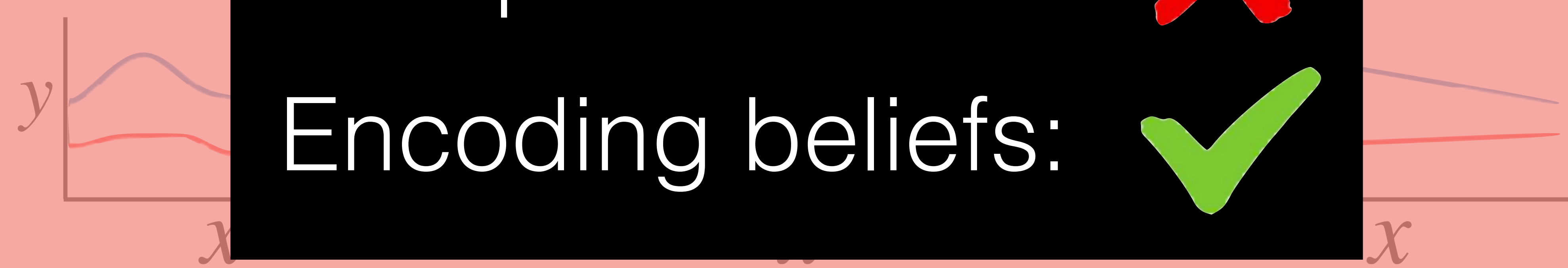
Encoding beliefs: ✗

Nonparametric Priors on NN's Function

Computation: ✗

Encoding beliefs: ✓

Nonparametric Priors on NN's Function



Computation: ✗

Encoding beliefs: ✓

Complications to Bayes Workflow

- ⊗ Infinite width limits
- ⊗ Divergences involving stochastic processes
- ⊗ Pre-training the prior

[Lee, 2004]

Priors for Neural Networks

Herbert K. H. Lee

Department of Applied Mathematics and Statistics

University of California, Santa Cruz

herbie@ams.ucsc.edu

Abstract

Neural networks are commonly used for classification and regression. The Bayesian approach may be employed, but choosing a prior for the parameters presents challenges. This paper reviews several priors in the literature and introduces Jeffreys priors for neural network models. The effect on the posterior is demonstrated through an example.

Key Words: nonparametric classification; nonparametric regression; Bayesian statistics; prior sensitivity

1 Introduction

Neural networks are a popular tool for nonparametric classification and regression. They offer a computationally tractable model that is fully flexible, in the sense of being able to approximate a wide range of functions (such as all continuous functions). Many references on neural networks are available (Bishop, 1995; Fine, 1999; Ripley, 1996). The Bayesian approach is appealing as it allows full accounting for uncertainty in the model and the choice of model (Lee, 2001; Neal, 1996). An important decision in any Bayesian analysis is the choice of prior. The idea is that your prior should reflect your current beliefs (either from previous data

Chapter 3

Survey of Neural Network Priors

We demand rigidly defined areas of doubt and uncertainty!

Douglas Adams
The Hitchhiker's Guide to the Galaxy

Having covered the basics of Bayesian NNs and strategies for inferring their posterior, I now turn to the focal point of the dissertation: prior distributions for both conditional NNs and density networks. Surprisingly, a broad review of Bayesian NN priors has been performed by only Robinson [2001], which is now considerably out of date. Thus, in this chapter I survey the existing work on NN priors, some of which was performed in the early days of Bayesian NNs and therefore also discussed by Robinson [2001]. However, most of the work is recent, some having been conducted concurrently with my own work to be presented in the coming chapters.

NNs have been applied to a myriad of different problems over the past thirty years, and this of course makes it impossible to discuss every prior ever used for a NN. Instead, I attempt to summarize broad themes from the literature that pertain to core NN methodology. For instance,

[Nalisnick, 2018]

[Lee, 2004]

eural Networks

K. H. Lee

Mathematics and Statistics

lifornia, Santa Cruz

rs.ucsc.edu

ostract

ication and regression. The Bayesian approach may be
presents challenges. This paper reviews several priors
neural network models. The effect on the posterior is

rametric regression; Bayesian statistics; prior sensitiv-

Neural networks are a popular tool for nonparametric classification and regression. They offer a computationally tractable model that is fully flexible, in the sense of being able to approximate a wide range of functions (such as all continuous functions). Many references on neural networks are available (Bishop, 1995; Fine, 1999; Ripley, 1996). The Bayesian approach is appealing as it allows full accounting for uncertainty in the model and the choice of model (Lee, 2001; Neal, 1996). An important decision in any Bayesian analysis is the choice of prior. The idea is that your prior should reflect your current beliefs (either from previous data

Chapter 3

Survey of Neural Network Priors

We demand rigidly defined areas of doubt and uncertainty!

Douglas A

The Hitchhiker's Guide to the G

Having covered the basics of Bayesian NNs and strategies for inferring their posterior, I turn to the focal point of the dissertation: prior distributions for both conditional NNs and sity networks. Surprisingly, a broad review of Bayesian NN priors has been performed by Robinson [2001], which is now considerably out of date. Thus, in this chapter I survey th isting work on NN priors, some of which was performed in the early days of Bayesian NNs therefore also discussed by Robinson [2001]. However, most of the work is recent, some ha been conducted concurrently with my own work to be presented in the coming chapters.

NNs have been applied to a myriad of different problems over the past thirty years, and of course makes it impossible to discuss every prior ever used for a NN. Instead, I attem summarize broad themes from the literature that pertain to core NN methodology. For insta

[Nalisnick, 2018]

[Lee, 2004]

eural Networks

K. H. Lee

Neural networks are a popular tool for n tationally tractable model that is fully fl functions (such as all continuous functions Fine, 1999; Ripley, 1996). The Bayesian a the model and the choice of model (Lee, 2 the choice of prior. The idea is that your

Priors in Bayesian Deep Learning: A Review

Vincent Fortuin
Department of Computer Science
ETH Zürich
Zürich, Switzerland
fortuin@inf.ethz.ch

Abstract

While the choice of prior is one of the most critical parts of the Bayesian inference workflow, recent Bayesian deep learning models have often fallen back on vague priors, such as standard Gaussians. In this review, we highlight the importance of prior choices for Bayesian deep learning and present an overview of different priors that have been proposed for (deep) Gaussian processes, variational autoencoders, and Bayesian neural networks. We also outline different methods of learning priors for these models from data. We hope to motivate practitioners in Bayesian deep learning to think more carefully about the prior specification for their models and to provide them with some inspiration in this regard.

1 Introduction

Bayesian models have gained a stable popularity in data analysis [1] and machine learning [2]. Especially in recent years, the interest in combining these models with deep learning has surged[1]. The main idea of Bayesian modeling is to infer a *posterior* distribution over the parameters $\boldsymbol{\theta}$ of the model given some observed data $\mathcal{D}$ using Bayes' theorem [3, 4] as

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\mathcal{D})} = \frac{p(\mathcal{D} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{\int p(\mathcal{D} \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}} \quad (1)$$

[Fortuin, 2021]

CONTRIBUTION:

Predictive Complexity Priors

In collaboration with

Jonathan Gordon

José Miguel Hernández-Lobato

2021

Predictive Complexity Priors

- ⊗ Define a notion of model selection: divergence from a reference model.
- ⊗ Then perform a change of variables to get a proper prior on the weights.

Predictive Complexity Priors

- ⊗ Define a notion of model selection: divergence from a reference model.
- ⊗ Then perform a change of variables to get a proper prior on the weights.

Predictive Complexity Priors

- ⊗ Define a notion of model selection: divergence from a reference model.
- ⊗ Then perform a change of variables to get a proper prior on the weights.

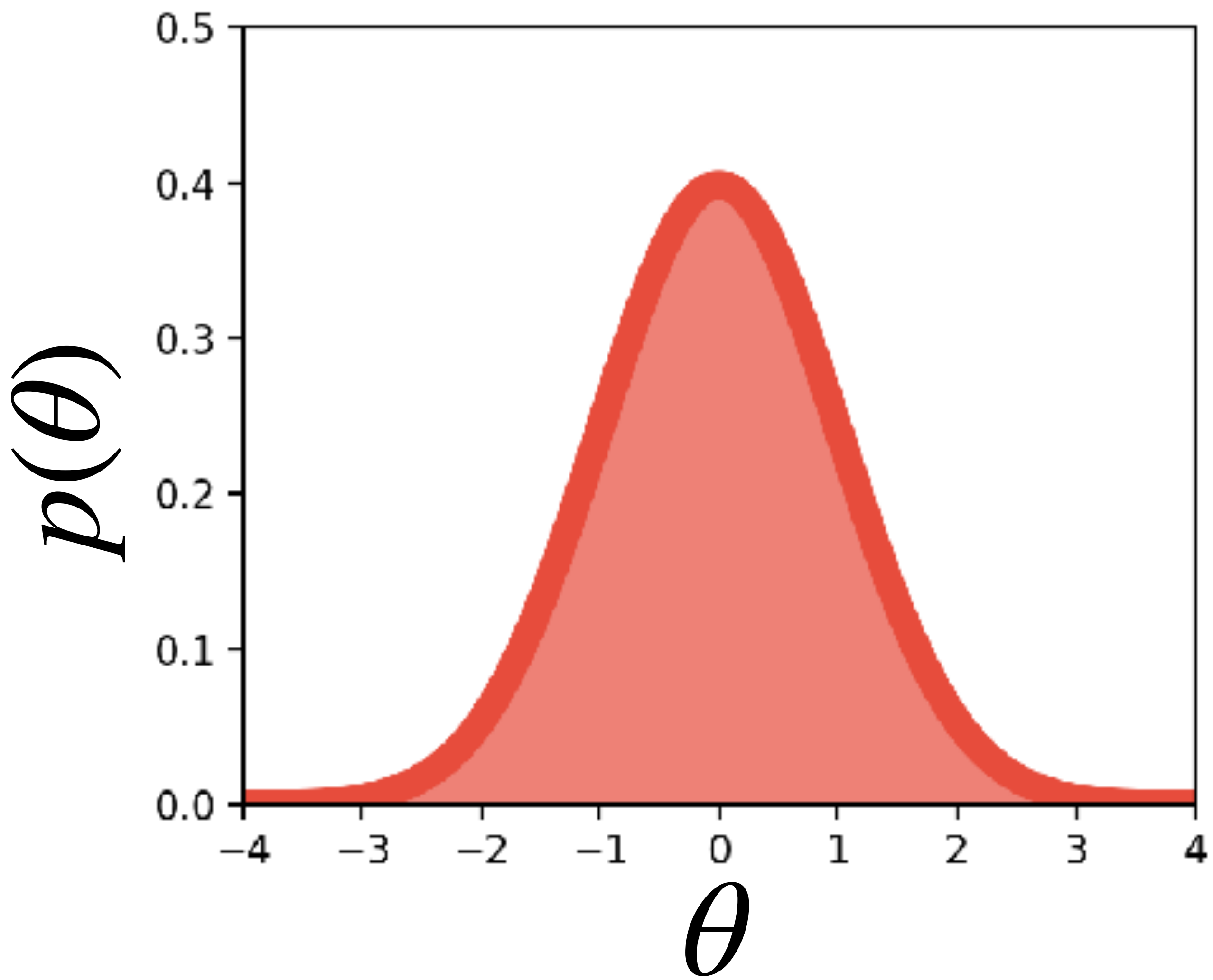
Builds off of

Penalising model component complexity: A principled, practical approach to constructing priors

Daniel Simpson*, Håvard Rue, Thiago G. Martins, Andrea Riebler, and Sigrunn H. Sørbye

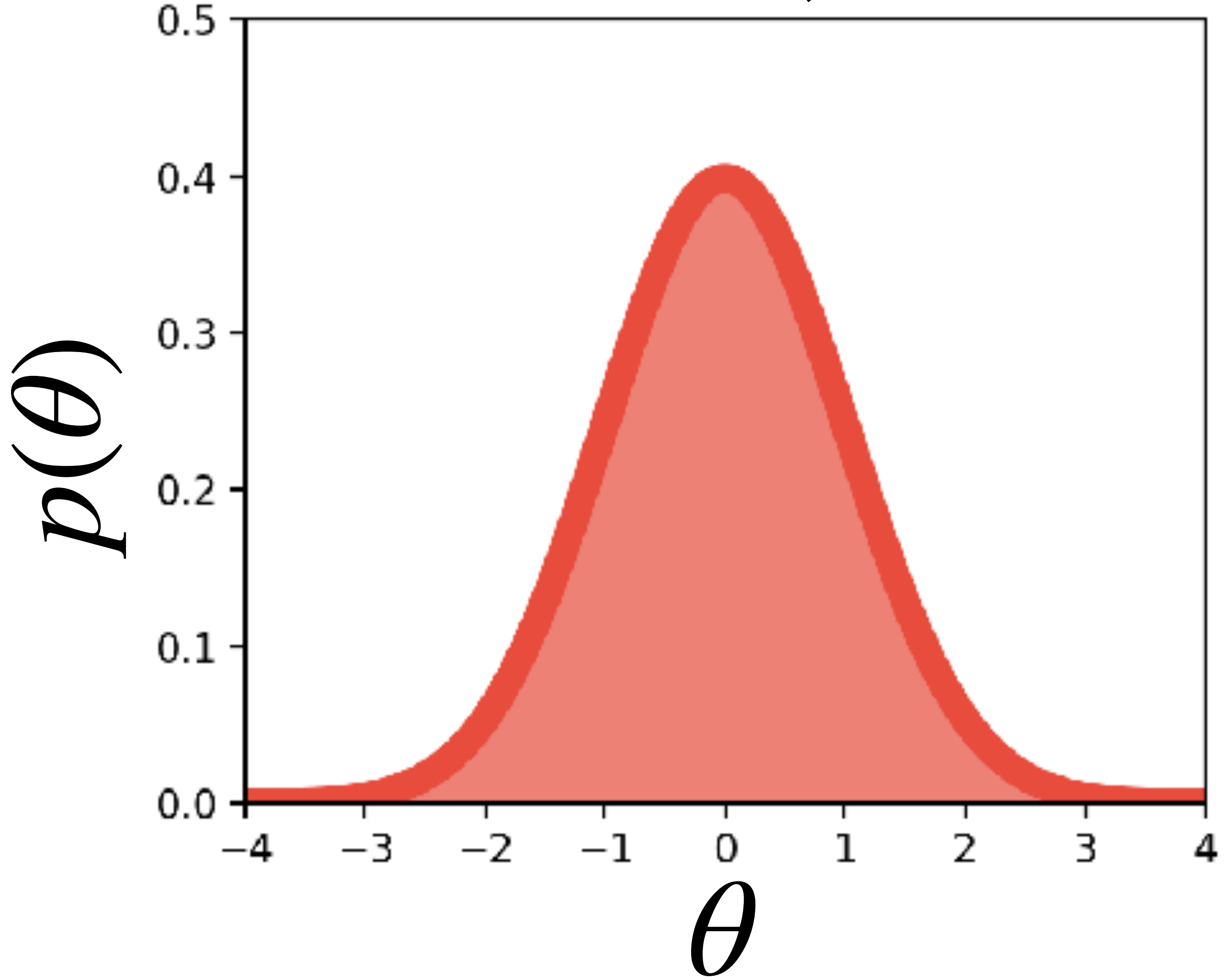
University of Warwick, NTNU, University of Tromsø The Arctic University

Usual Way

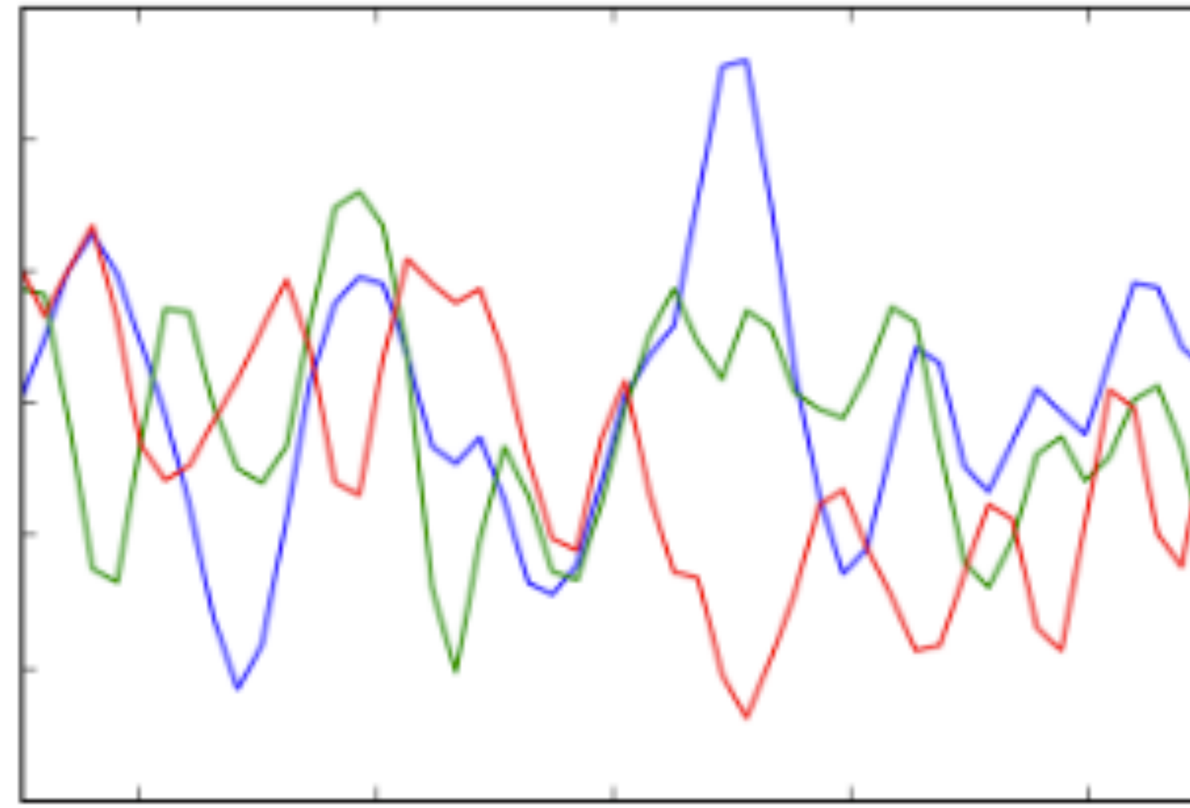


Usual Way

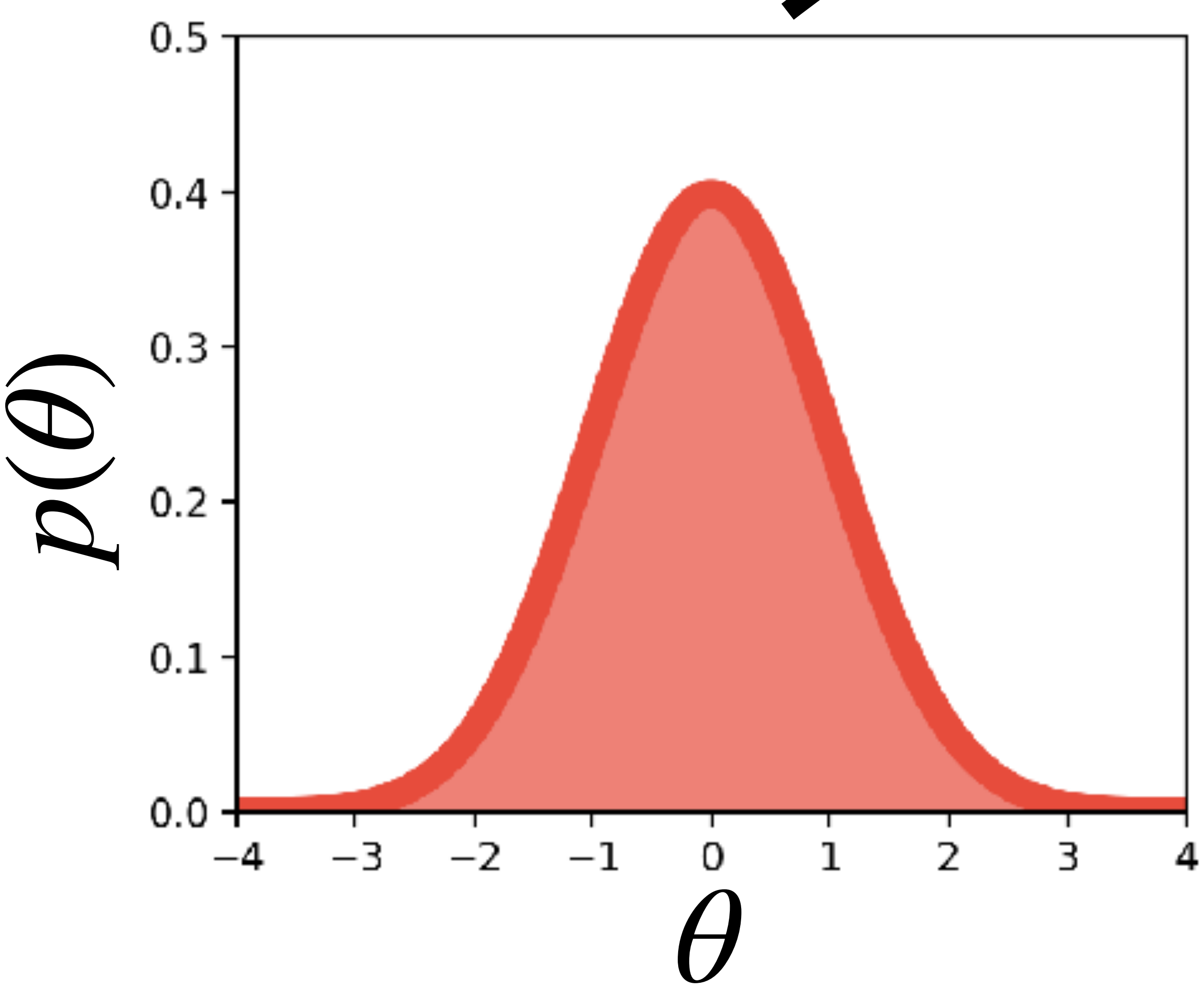
$p(\mathbf{y} \mid \theta)$



Usual Way

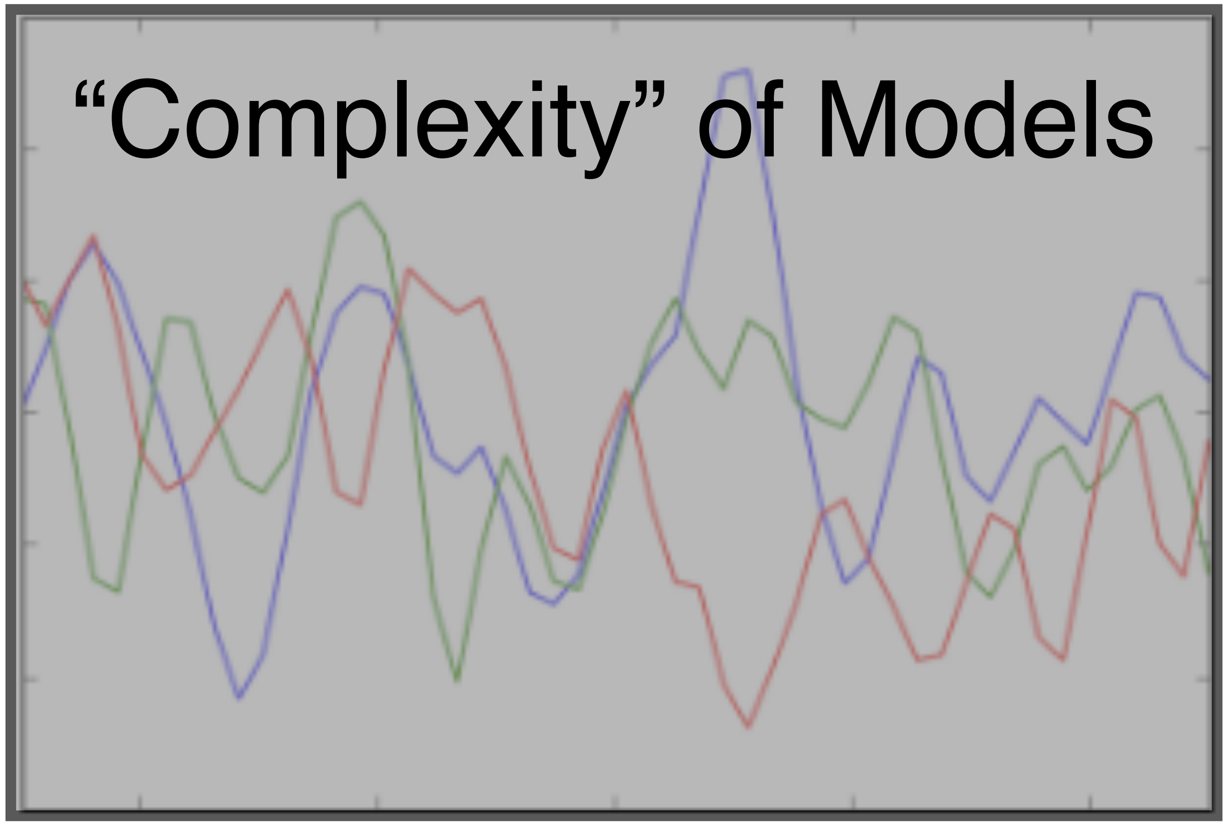


$p(\mathbf{y} \mid \theta)$

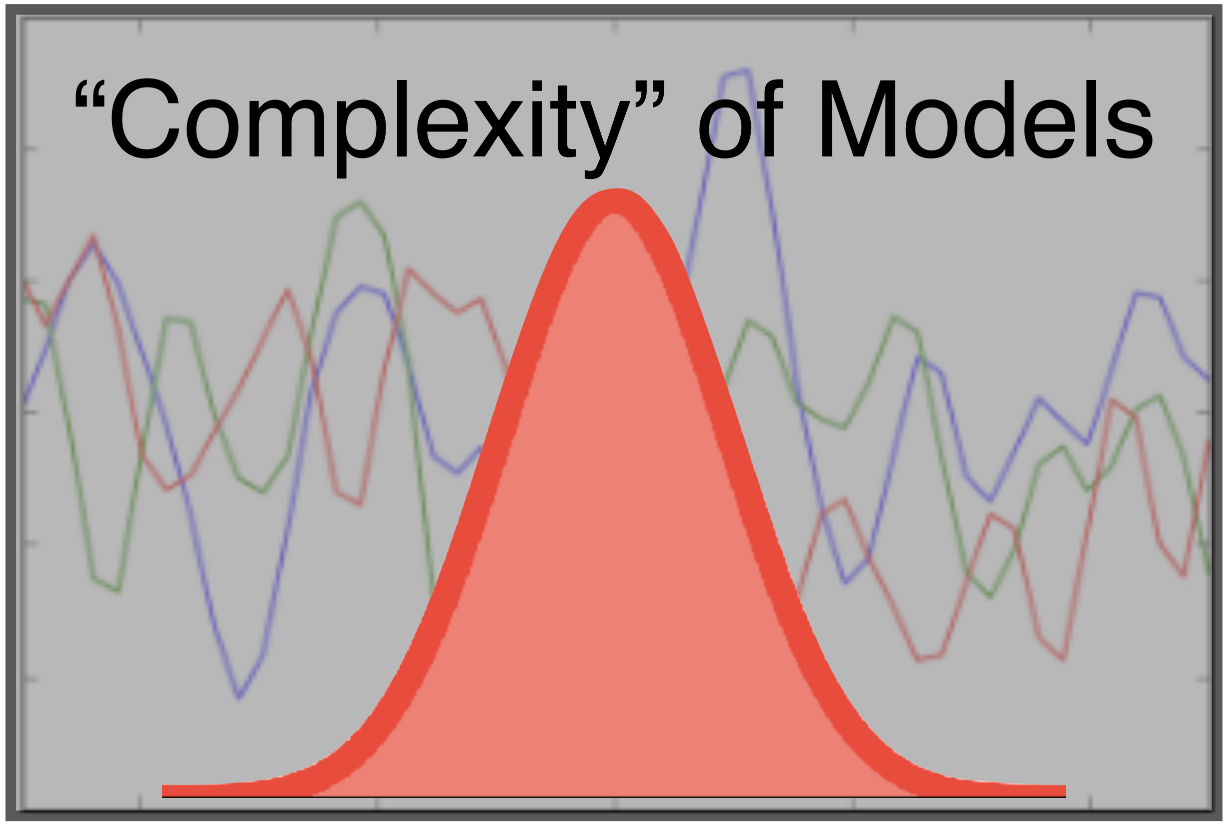


Samples from induced distribution on predictive models / functions

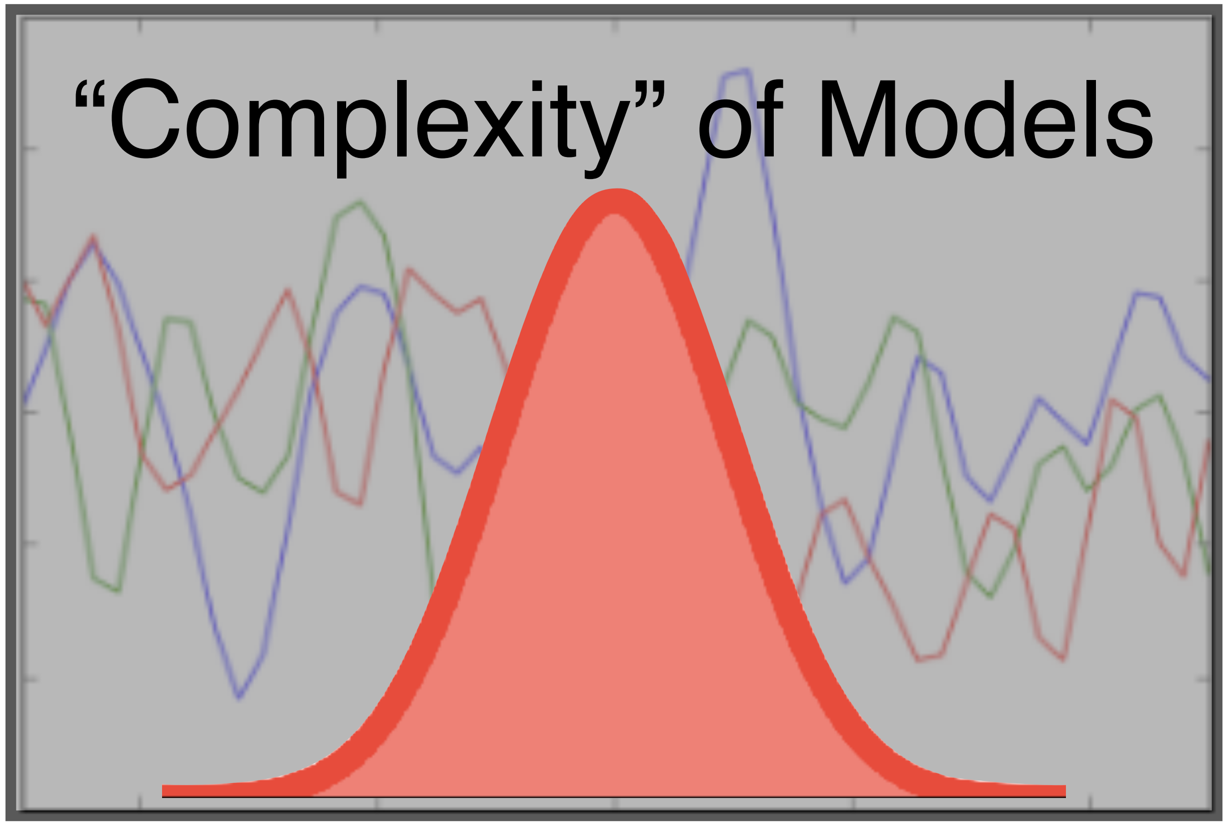
This Work



This Work

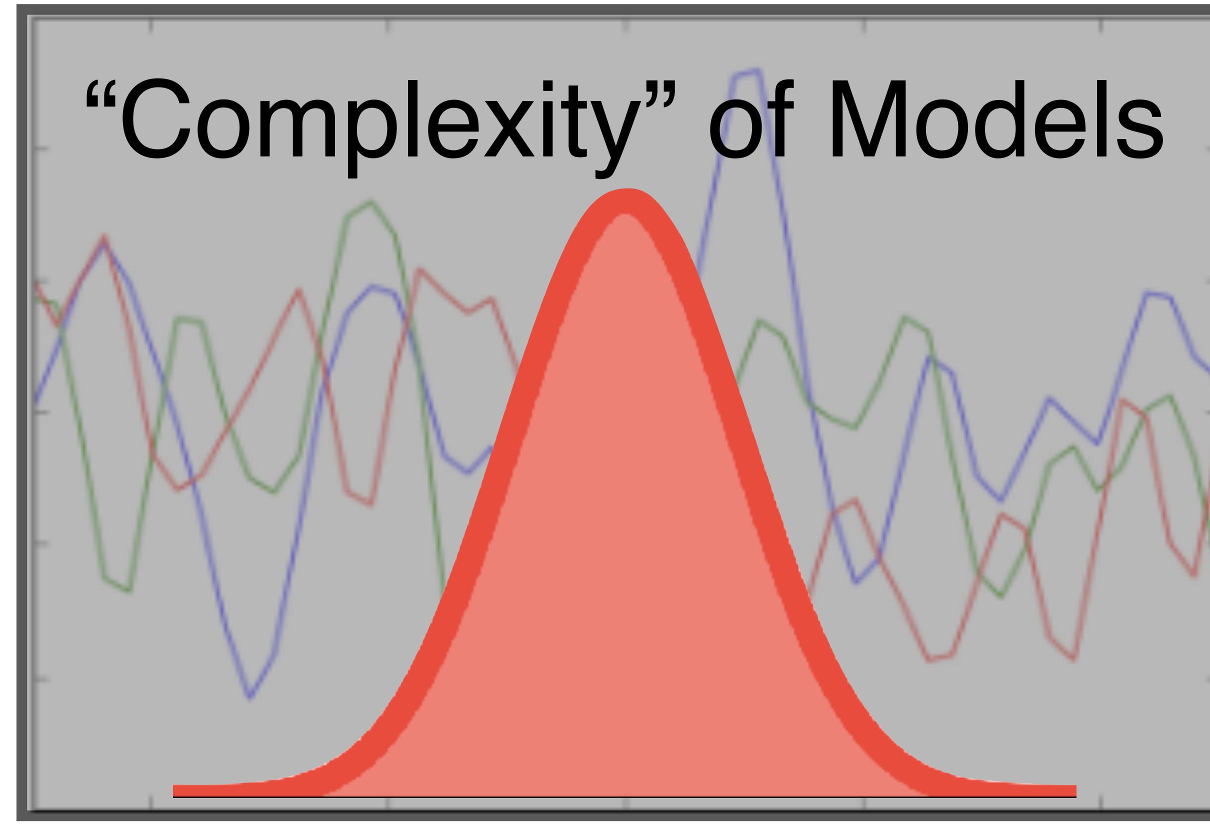


This Work

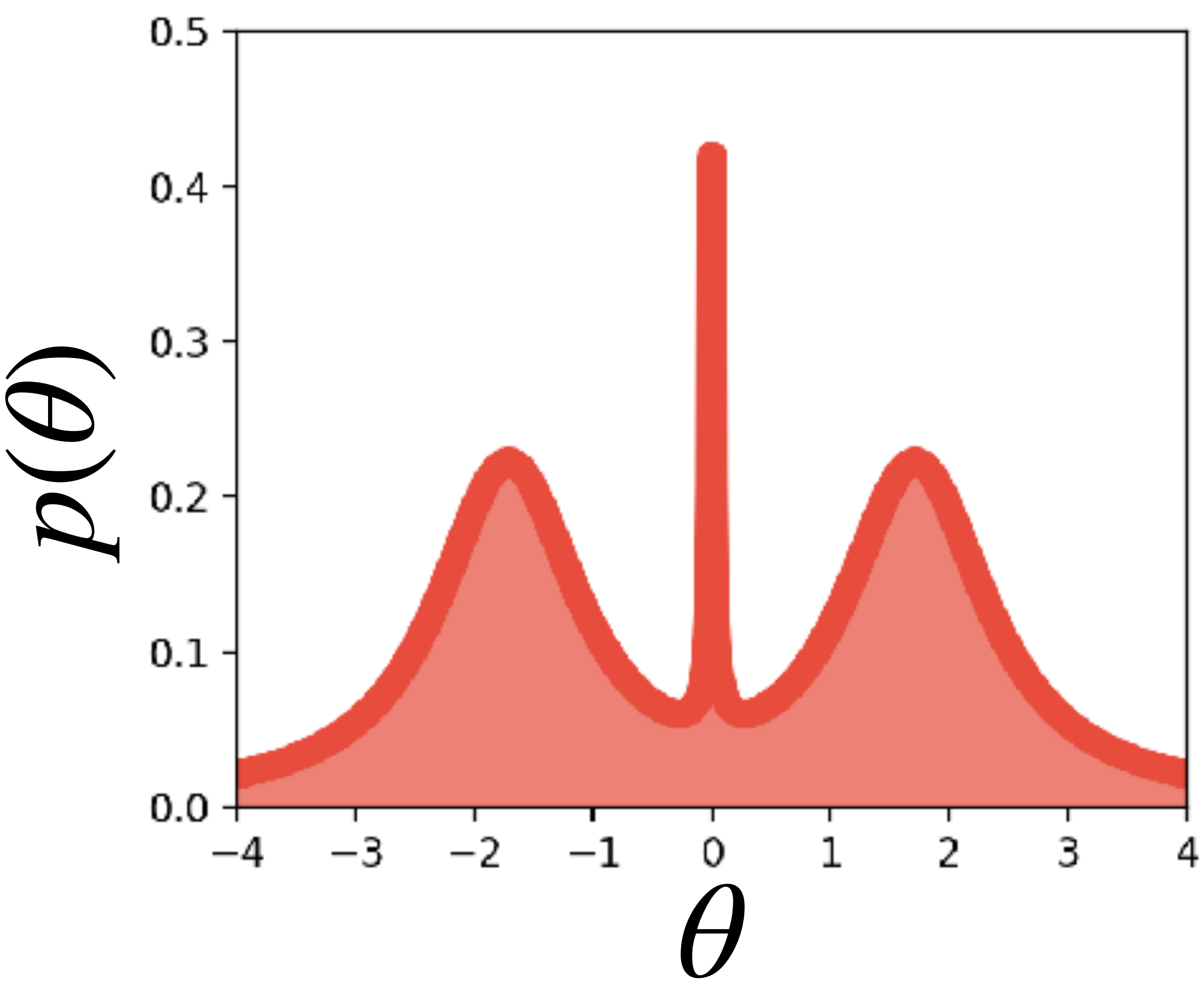


$p^{-1}(\mathbf{y} \mid \theta)$

This Work



$$p^{-1}(\mathbf{y} \mid \theta)$$



Model of Interest

PRIOR: $\boldsymbol{\theta} \sim p(\boldsymbol{\theta} \,|\, \tau)$

DATA MODEL: $\mathbf{y} \sim p(\mathbf{y} \,|\, \boldsymbol{\theta})$

Model of Interest

WEIGHT PRIOR: $$\boldsymbol{\theta} \sim N(\boldsymbol{\phi}, \tau \mathbb{I})$$

NEURAL NET: $$\mathbf{y} \sim p(\mathbf{y} \mid \boldsymbol{\theta})$$

Model of Interest

WEIGHT PRIOR: $\boldsymbol{\theta} \sim N(\boldsymbol{\phi}, \boxed{\boldsymbol{\tau}}\,\square)$

NEURAL NET: $\mathbf{y} \sim p(\mathbf{y} \,|\, \boldsymbol{\theta})$

GOAL	Define Hyper-Prior

$$p(\tau)$$

METHOD:

Recipe for
Prior Specification

Three Steps

Three Steps

STEP #1 Define Reference Model

$$p(\mathbf{y} \mid \boldsymbol{\phi})$$

Same parameters as the
mean of our first-level prior: $\boldsymbol{\theta} \sim N(\boldsymbol{\phi}, \tau \mathbb{I})$

Three Steps

STEP #1 Define Reference Model

$$p(\mathbf{y} \,|\, \boldsymbol{\phi})$$

Same parameters as the mean of our first-level prior: $\boldsymbol{\theta} \sim N(\boldsymbol{\phi}, \tau\mathbb{I})$

These parameters ($\boldsymbol{\phi}$) should encode our inductive bias or prior beliefs.

Three Steps

STEP #2	Specify Divergence

Three Steps

STEP #2	Specify Divergence

$$\kappa = \mathbb{E}_{\theta|\tau}\left[\mathbb{D}[p(\mathbf{y}\,|\,\boldsymbol{\theta}) \;||\; p(\mathbf{y}\,|\,\boldsymbol{\phi})]\right]$$

Three Steps

STEP #2 Specify Divergence

$$\kappa = \mathbb{E}_{\theta|\tau}\left[\underbrace{\mathbb{D}[p(\mathbf{y}\,|\,\boldsymbol{\theta}) \;||\; p(\mathbf{y}\,|\,\boldsymbol{\phi})]}_{\text{Divergence between model of interest and reference model}}\right]$$

Three Steps

STEP #2 | Specify Divergence

$$\kappa = \mathbb{E}_{\theta|\tau}\left[\mathbb{D}[p(\mathbf{y} \,|\, \boldsymbol{\theta}) \;||\; p(\mathbf{y} \,|\, \boldsymbol{\phi})]\right]$$

Expectation taken over first-level prior

Divergence between model of interest and reference model

Three Steps

STEP #2 Specify Divergence

$$\kappa = \mathbb{E}_{\theta|\tau}\left[\mathbb{D}[p(\mathbf{y}\,|\,\boldsymbol{\theta})\ ||\ p(\mathbf{y}\,|\,\boldsymbol{\phi})]\right]$$

Divergence between model of interest and reference model

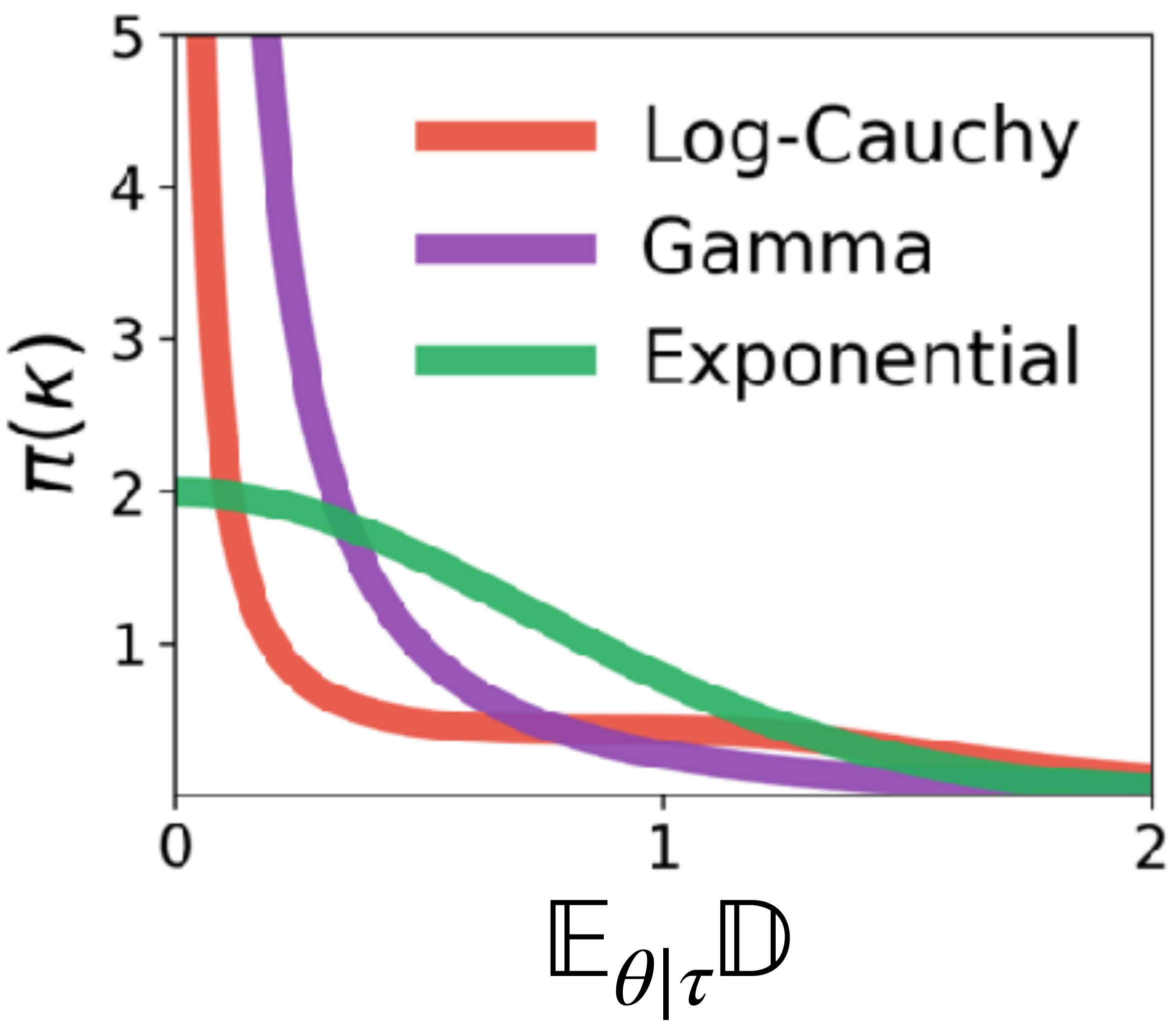
Expectation taken over first-level prior

If divergence is KL, then this is the **expected bits lost** when approximating the full model with the reference model.

Three Steps

STEP #3 Define Prior & Reparametrize

$$\pi(\kappa)$$



Three Steps

STEP #3 Define Prior & Reparametrize

$$p(\tau) = \pi(\kappa) \left| \frac{\partial \kappa}{\partial \tau} \right|$$

Three Steps

STEP #3 Define Prior & Reparametrize

$$p(\tau) = \pi(\kappa) \left| \frac{\partial \kappa}{\partial \tau} \right|$$

$$= \pi\left(\mathbb{E}_{\theta|\tau}\left[\mathbb{D}[p_\theta \;||\; p_\phi]\right]\right) \left| \frac{\partial \mathbb{E}_{\theta|\tau}\mathbb{D}}{\partial \tau} \right|$$

Three Steps

STEP #3 Define Prior & Reparametrize

$$p(\tau) = \pi(\kappa) \left| \frac{\partial \kappa}{\partial \tau} \right|$$

$$\boxed{= \pi\left(\mathbb{E}_{\theta|\tau}\left[\mathbb{D}[p_\theta \,||\, p_\phi] \right] \right) \left| \frac{\partial \mathbb{E}_{\theta|\tau}\mathbb{D}}{\partial \tau} \right|}$$

FINAL FORM

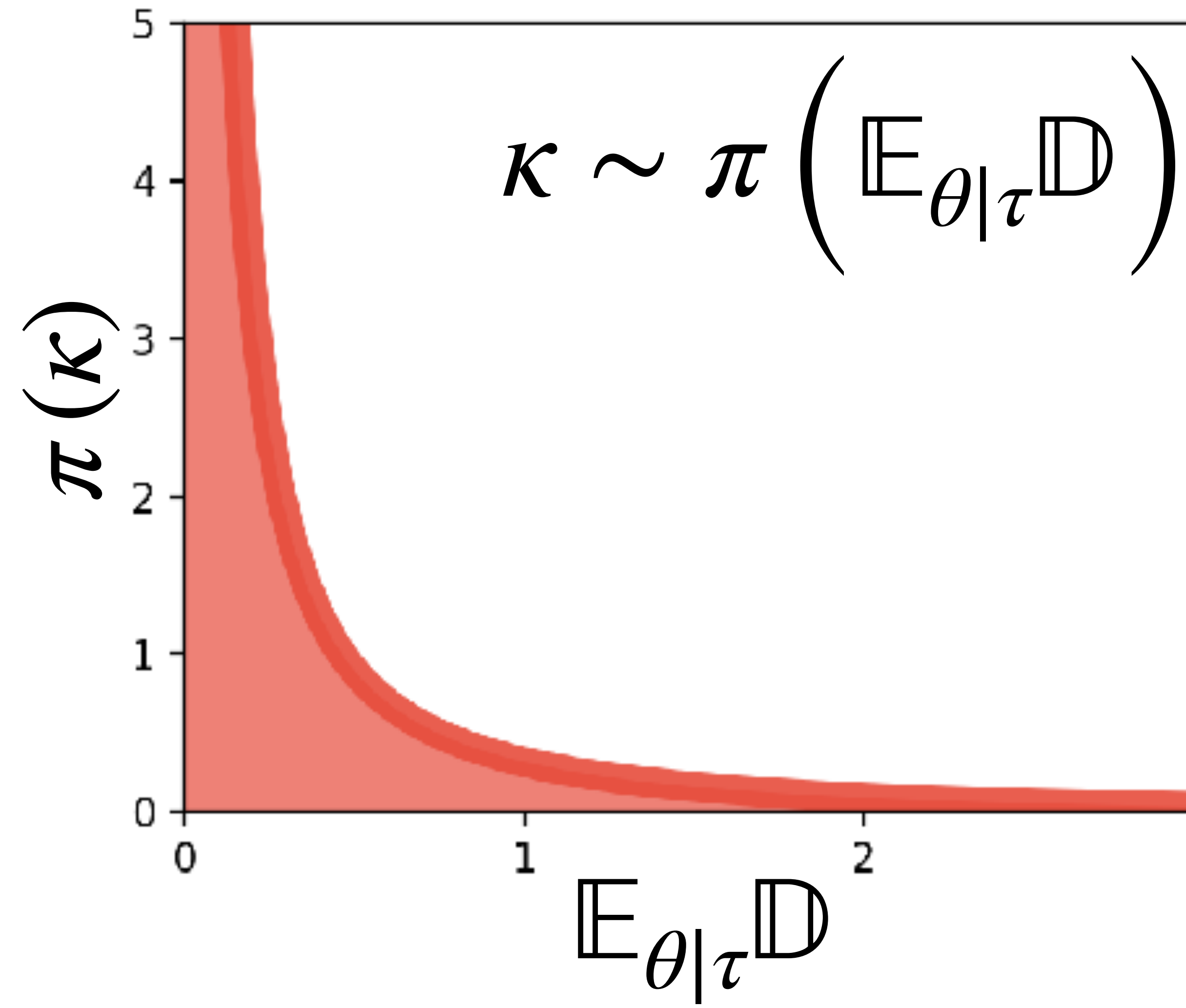
Three Steps

STEP #1	Define Reference Model
STEP #2	Specify Divergence
STEP #3	Define Prior & Reparametrize

$$= \pi\left(\mathbb{E}_{\theta|\tau}\left[\mathbb{D}[p_\theta \,||\, p_\phi]\right]\right)\left|\frac{\partial \mathbb{E}_{\theta|\tau}\mathbb{D}}{\partial \tau}\right|$$

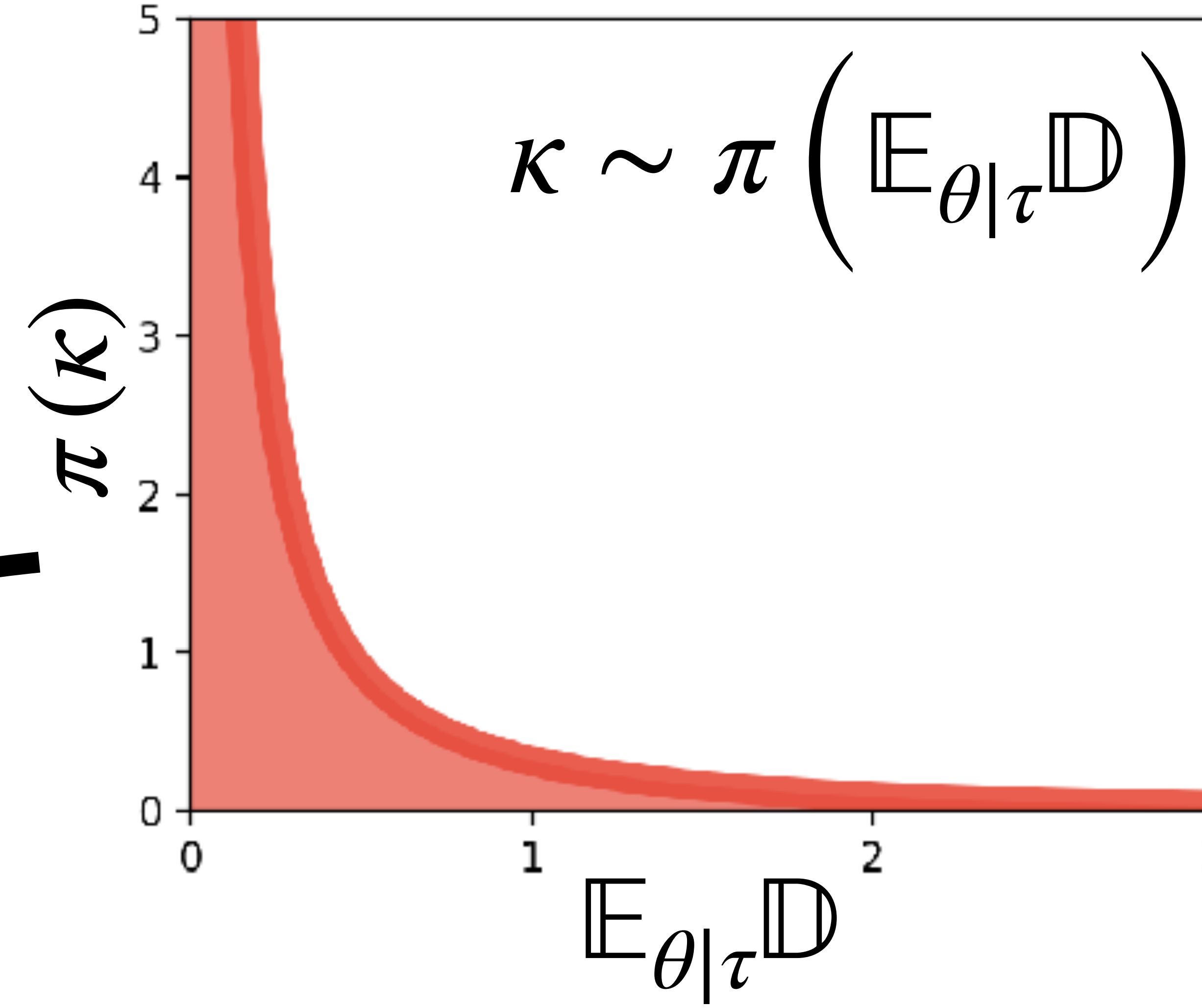
FINAL FORM

Generative Process



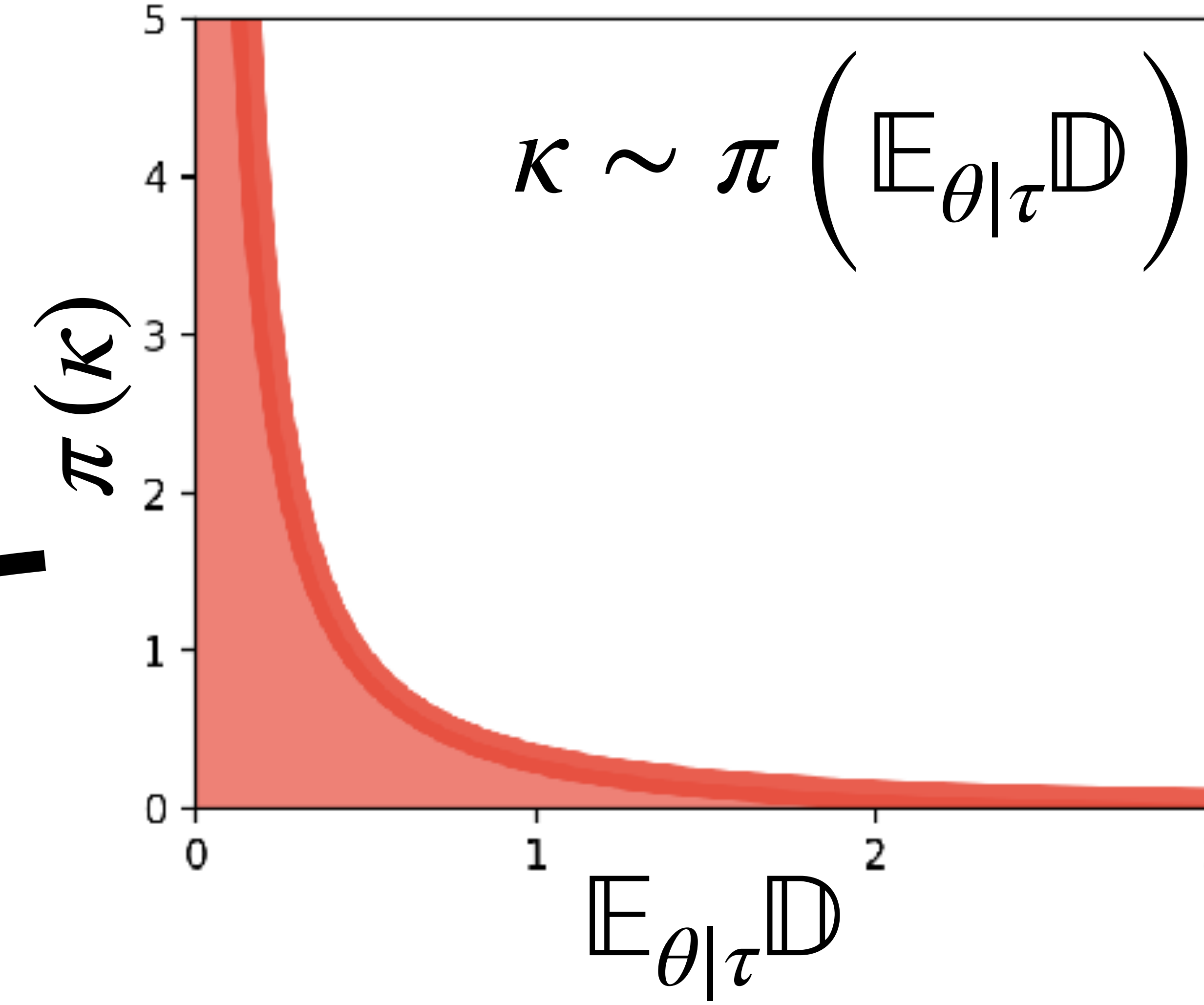
Generative Process

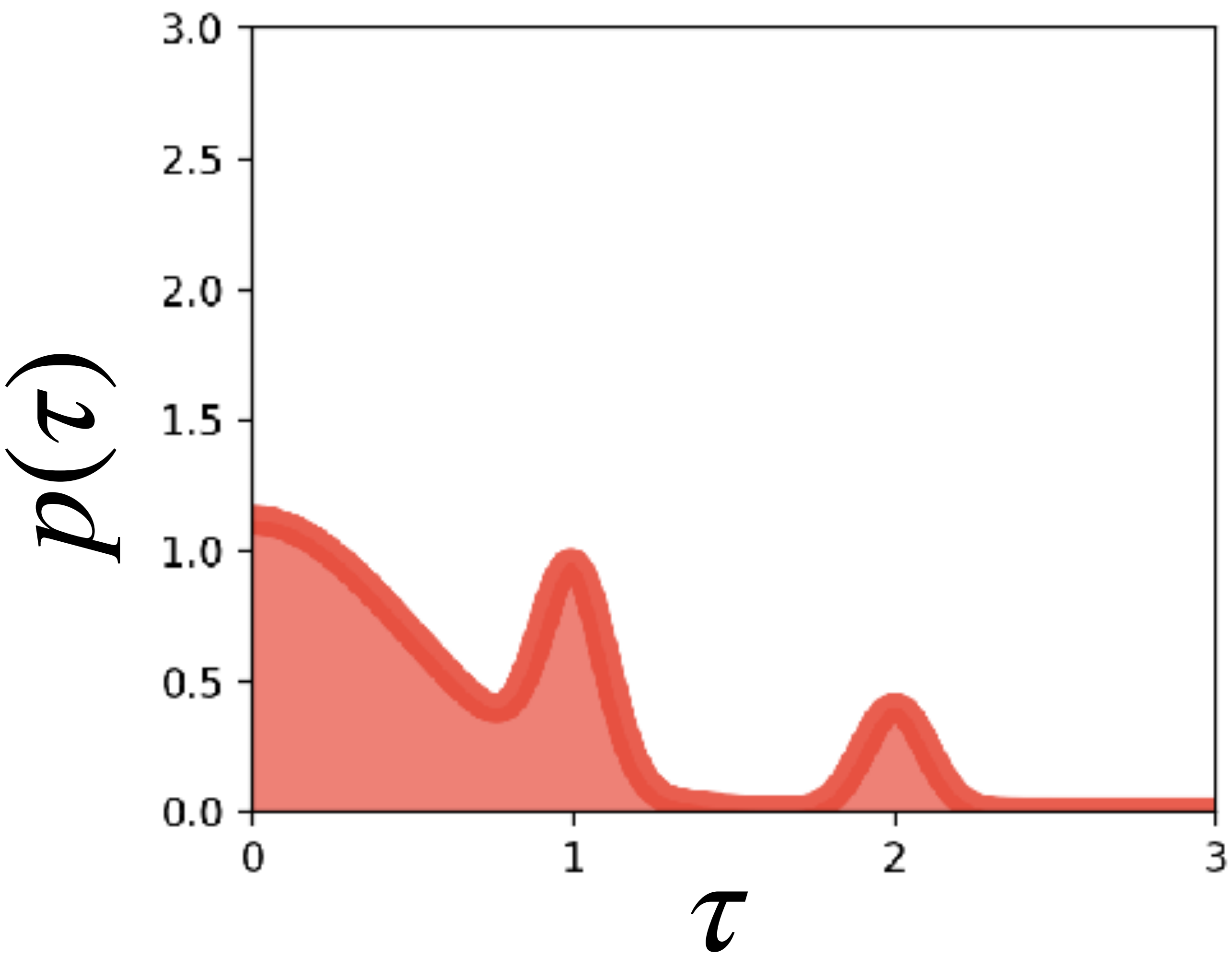
$$\tau = (\mathbb{E}_{\theta|\tau}\mathbb{D})^{-1}(\kappa; p_\theta, p_\phi)$$



Generative Process

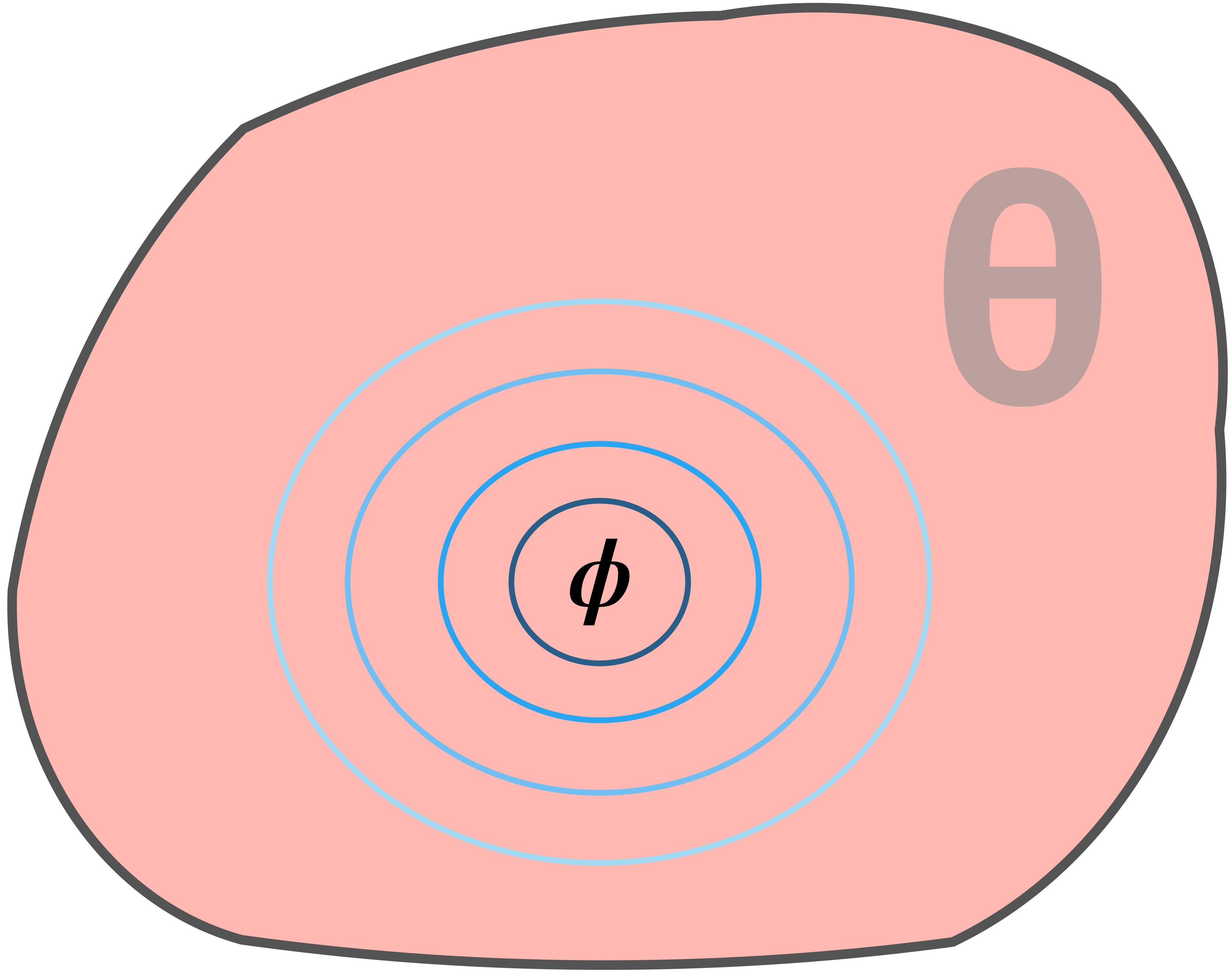
$$\tau = (\mathbb{E}_{\theta|\tau}\mathbb{D})^{-1}(\kappa; p_\theta, p_\phi)$$





Meaningful Notion of Scale

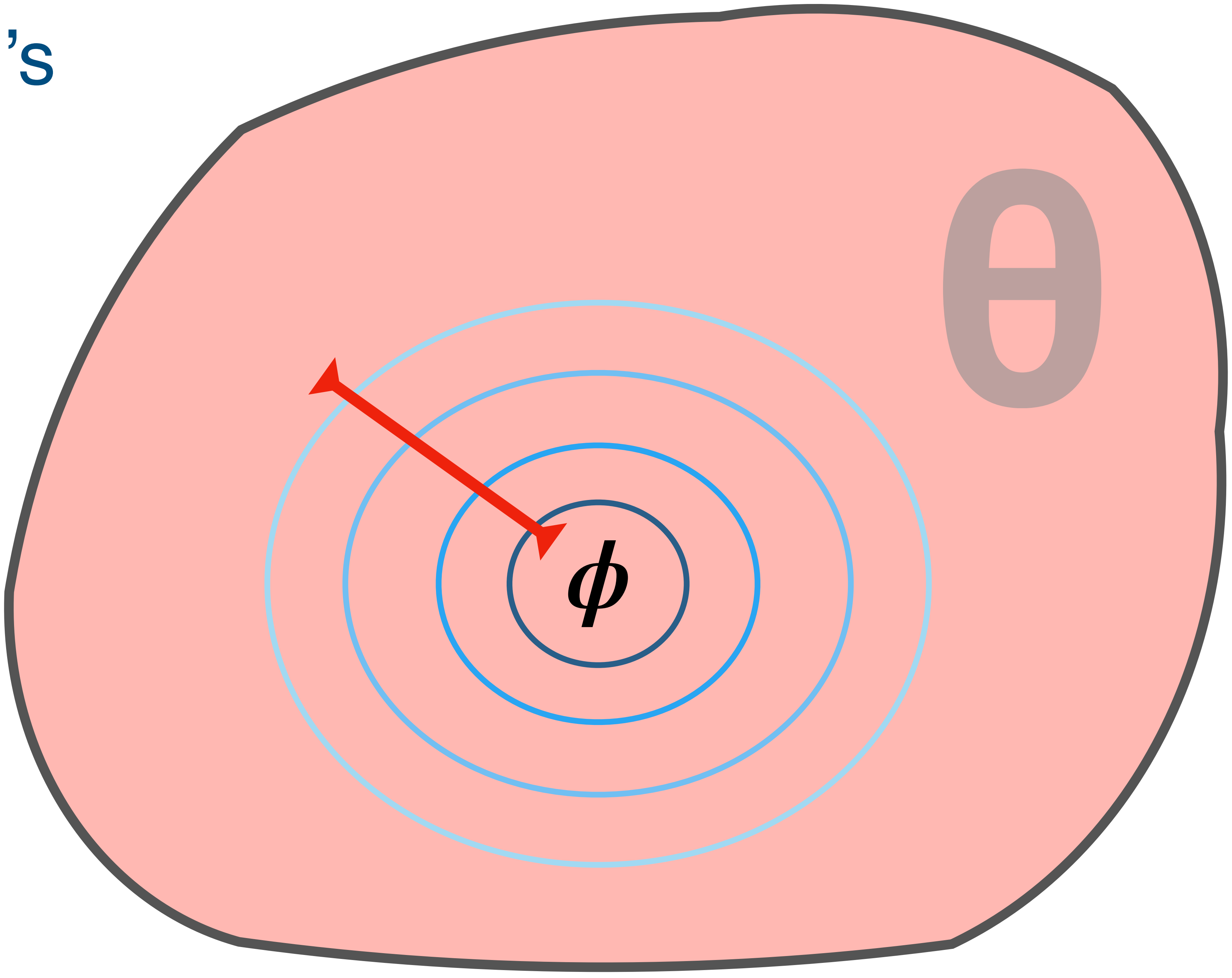
$$\boldsymbol{\theta} \sim N(\boldsymbol{\phi}, \tau \mathbb{I})$$



Meaningful Notion of Scale

Scale ($\boldsymbol{\tau}$) is determined by **how quickly our model's predictions (on training data) deviate from the reference model's**

$$\boldsymbol{\theta} \sim N(\boldsymbol{\phi}, \tau \mathbb{I})$$



Limiting Prior

$$\lim_{\theta \to \phi} \mathbb{KLD}\left[p_\theta \;||\; p_\phi \right]$$

Limiting Prior

$$\lim_{\theta \to \phi} \mathbb{KLD}\left[p_\theta \,||\, p_\phi\right] = \frac{1}{2}(\theta - \phi)^2 I[\phi] + \text{higher order terms}$$

Limiting Prior

$$\lim_{\theta \to \phi} \mathbb{KLD}\left[p_\theta \mid\mid p_\phi\right] = \frac{1}{2}(\theta - \phi)^2 I[\phi] + \text{higher order terms}$$

$$p(\tau) = \pi\left(\frac{1}{2}\,\tau\, I[\phi]\right)\frac{1}{2}\, I[\phi]$$

Limiting Prior

$$\lim_{\theta \to \phi} \mathbb{KLD}\left[p_\theta \mid\mid p_\phi\right] = \frac{1}{2}(\theta - \phi)^2 I[\phi] + \text{higher order terms}$$

$$p(\tau) = \pi\left(\frac{1}{2}\,\tau\, I[\phi]\right)\frac{1}{2}\, I[\phi]$$

Fisher information for the <u>reference</u> parameter

SANITY CHECK: SHRINKAGE FOR LOGISTIC REGRESSION

Shrinkage Prior for Logistic Regression

$$\mathbf{y} \sim \textbf{Bernoulli}\left(f(\boldsymbol{\beta}^T\mathbf{x})\right)$$

$$\beta_d \sim N(0,\ \tau\lambda_d)$$

$$\lambda_d \sim C^+(0,1)$$

Shrinkage Prior for Logistic Regression

$$\mathbf{y} \sim \textbf{Bernoulli}\left(f(\boldsymbol{\beta}^T \mathbf{x})\right)$$

$$\beta_d \sim N(0,\ \tau\lambda_d)$$

$$\lambda_d \sim C^+(0,1)$$

$$\tau \sim p(\tau)$$

Shrinkage Prior for Logistic Regression

$$\mathbf{y} \sim \textbf{Bernoulli}\left(f(\boldsymbol{\beta}^T\mathbf{x})\right)$$

$$\beta_d \sim N(0,\ \tau\lambda_d)$$

$$\lambda_d \sim C^+(0,1)$$

$$\tau \sim p(\tau)$$

Reference model:

$$\mathbf{y} \sim \textbf{Bernoulli}\left(f(\mathbf{0}^T\mathbf{x})\right) = \textbf{Bernoulli}\left(0.5\right)$$

Shrinkage Prior for Logistic Regression

Table 1: *Logistic Regression*. Below we report test set predictive log-likelihoods for the half-Cauchy prior, ECP, and PredCP under both VI and MCMC. Results are averaged across 20 splits.

			Markov Chain Monte Carlo		
Data Set	N_{train}	D	Half-Cauchy	**ECP**	**PredCP**
allaml	51	7129			
colon	44	2000			
breast	82	9			

Shrinkage Prior for Logistic Regression

Table 1: *Logistic Regression*. Below we report test set predictive log-likelihoods for the half-Cauchy prior, ECP, and PredCP under both VI and MCMC. Results are averaged across 20 splits.

			Markov Chain Monte Carlo		
Data Set	N_{train}	D	Half-Cauchy	**ECP**	**PredCP**
`allaml`	51	7129	$-0.19_{\pm.02}$	$\mathbf{-0.17}_{\pm.02}$	$\mathbf{-0.17}_{\pm.02}$
`colon`	44	2000	$-0.54_{\pm.05}$	$\mathbf{-0.52}_{\pm.05}$	$-0.54_{\pm.04}$
`breast`	82	9	$-0.55_{\pm.02}$	$-0.55_{\pm.01}$	$-0.55_{\pm.02}$

Shrinkage Prior for Logistic Regression

Table 1: *Logistic Regression*. Below we report test set predictive log-likelihoods for the half-Cauchy prior, ECP, and PredCP under both VI and MCMC. Results are averaged across 20 splits.

			Markov Chain Monte Carlo		
Data Set	N_{train}	D	Half-Cauchy	ECP	PredCP
allaml	51	7129	$-0.19 \pm .02$	$\mathbf{-0.17} \pm .02$	$\mathbf{-0.17} \pm .02$
colon	44	2000	$-0.54 \pm .05$	$\mathbf{-0.52} \pm .05$	$-0.54 \pm .04$
breast	82	9	$-0.55 \pm .02$	$-0.55 \pm .01$	$-0.55 \pm .02$

			Variational Inference		
Data Set	N_{train}	D	Half-Cauchy	ECP	PredCP
allaml	51	7129			
colon	44	2000			
breast	82	9			

Shrinkage Prior for Logistic Regression

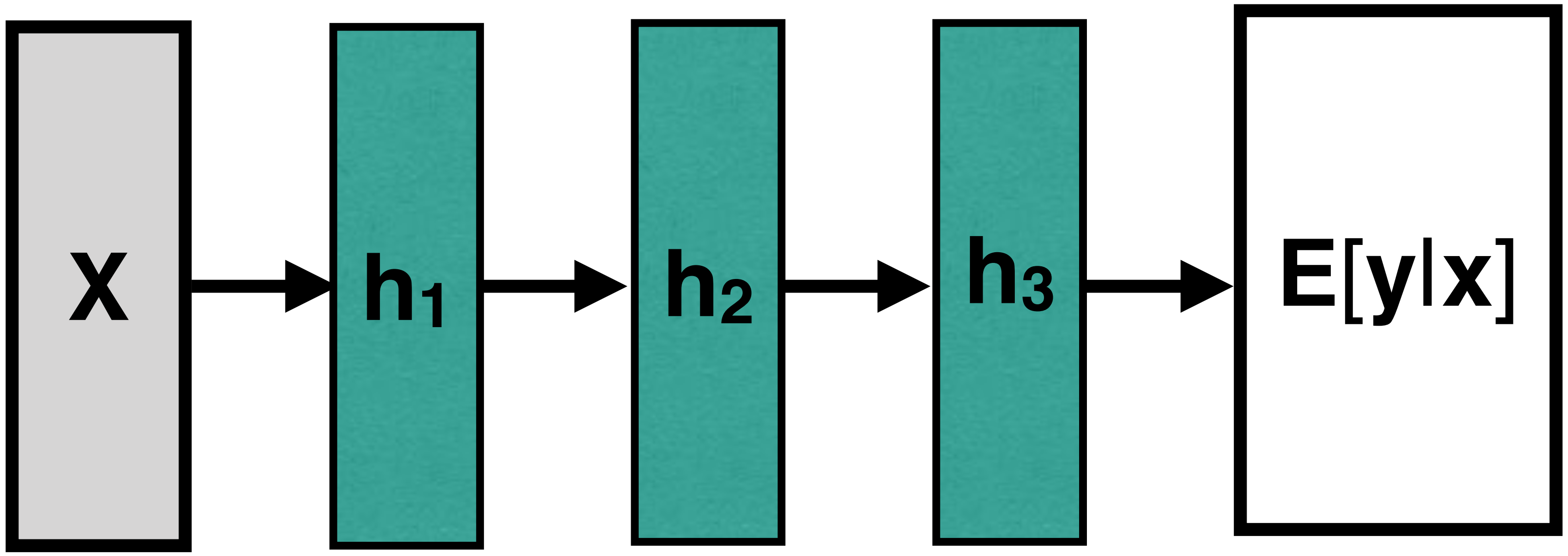
Table 1: *Logistic Regression*. Below we report test set predictive log-likelihoods for the half-Cauchy prior, ECP, and PredCP under both VI and MCMC. Results are averaged across 20 splits.

			Markov Chain Monte Carlo		
Data Set	N_{train}	D	Half-Cauchy	**ECP**	**PredCP**
allaml	51	7129	$-0.19 \pm .02$	$\mathbf{-0.17} \pm .02$	$\mathbf{-0.17} \pm .02$
colon	44	2000	$-0.54 \pm .05$	$\mathbf{-0.52} \pm .05$	$-0.54 \pm .04$
breast	82	9	$-0.55 \pm .02$	$-0.55 \pm .01$	$-0.55 \pm .02$

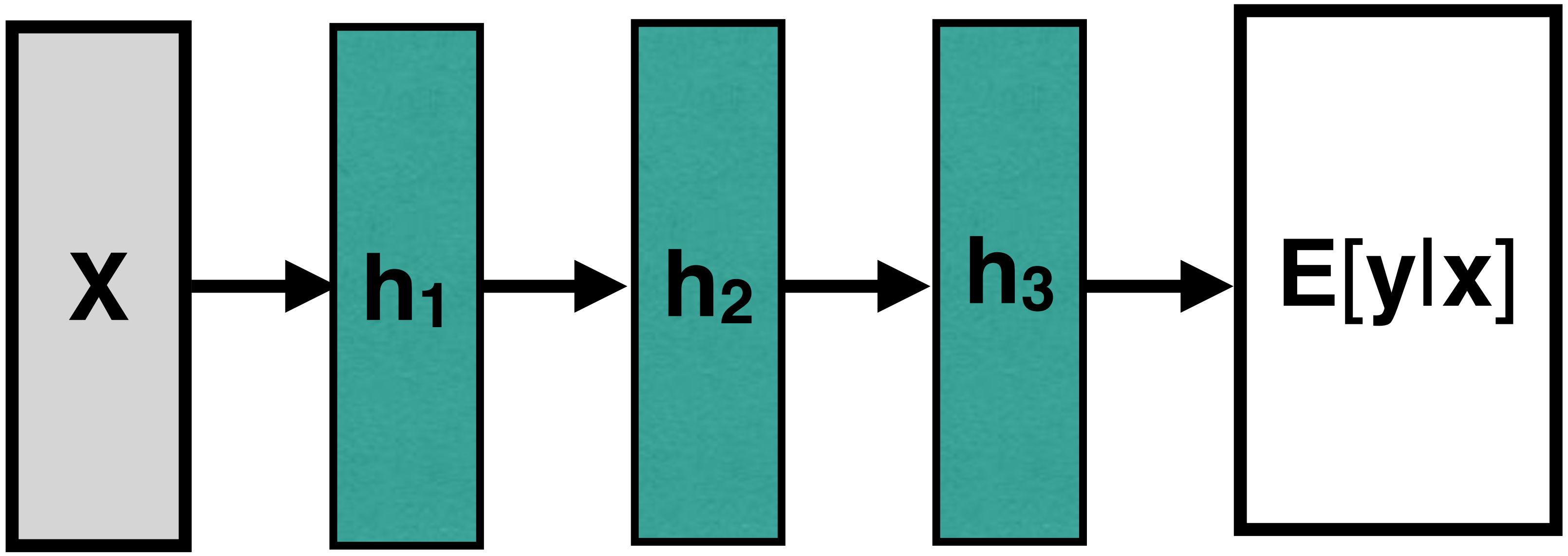
			Variational Inference		
Data Set	N_{train}	D	Half-Cauchy	**ECP**	**PredCP**
allaml	51	7129	$-0.43 \pm .01$	$\mathbf{-0.32} \pm .01$	$\mathbf{-0.32} \pm .01$
colon	44	2000	$\mathbf{-0.61} \pm .02$	$-0.63 \pm .03$	$-0.66 \pm .02$
breast	82	9	$-0.60 \pm .01$	$\mathbf{-0.58} \pm .01$	$\mathbf{-0.58} \pm .01$

Application to Depth Selection in Neural Networks

Layer-Wise Prior for NNs



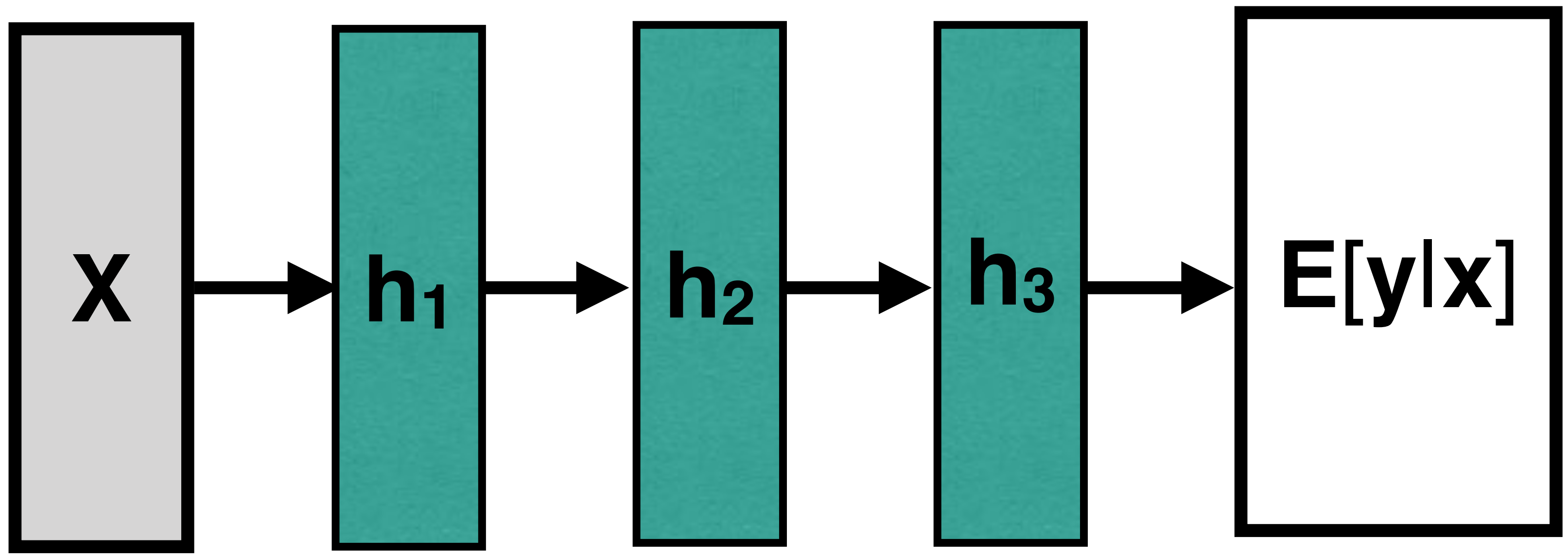
Layer-Wise Prior for NNs



$$\boldsymbol{\theta}_l \sim N(\mathbf{0},\ \tau_l \boldsymbol{\Sigma})$$

Layer index

Layer-Wise Prior for NNs

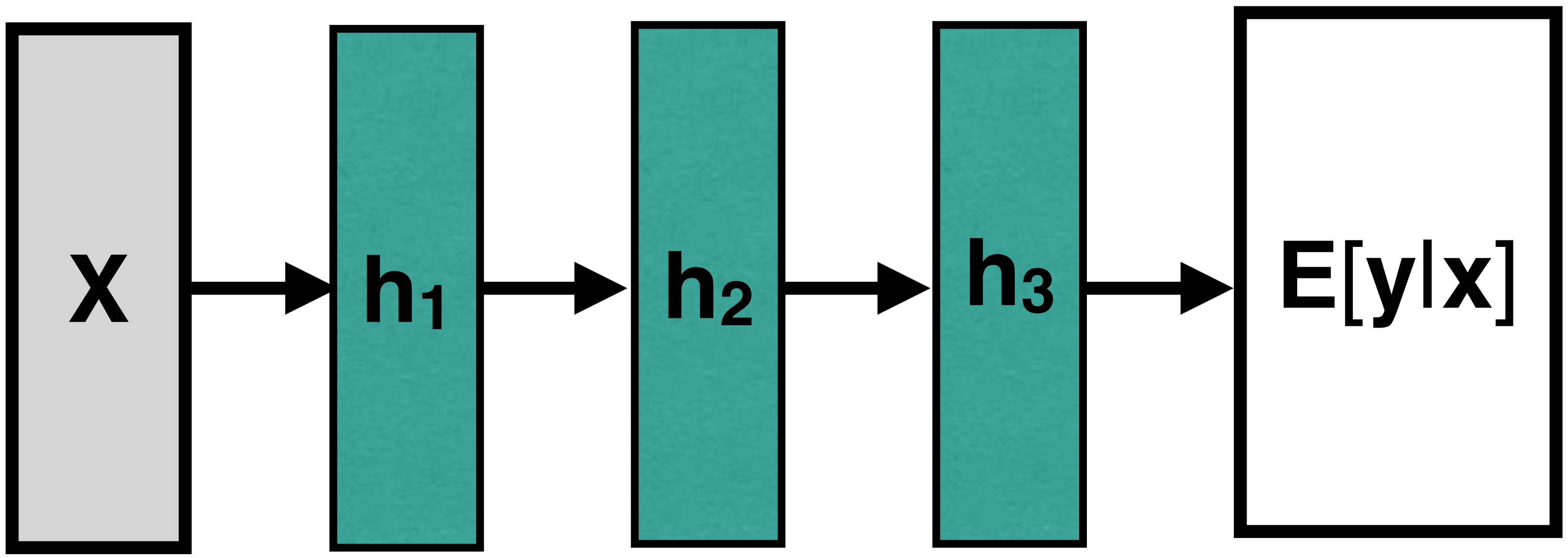


$$\boldsymbol{\theta}_l \sim N(\mathbf{0},\ \tau_l \boldsymbol{\Sigma})$$

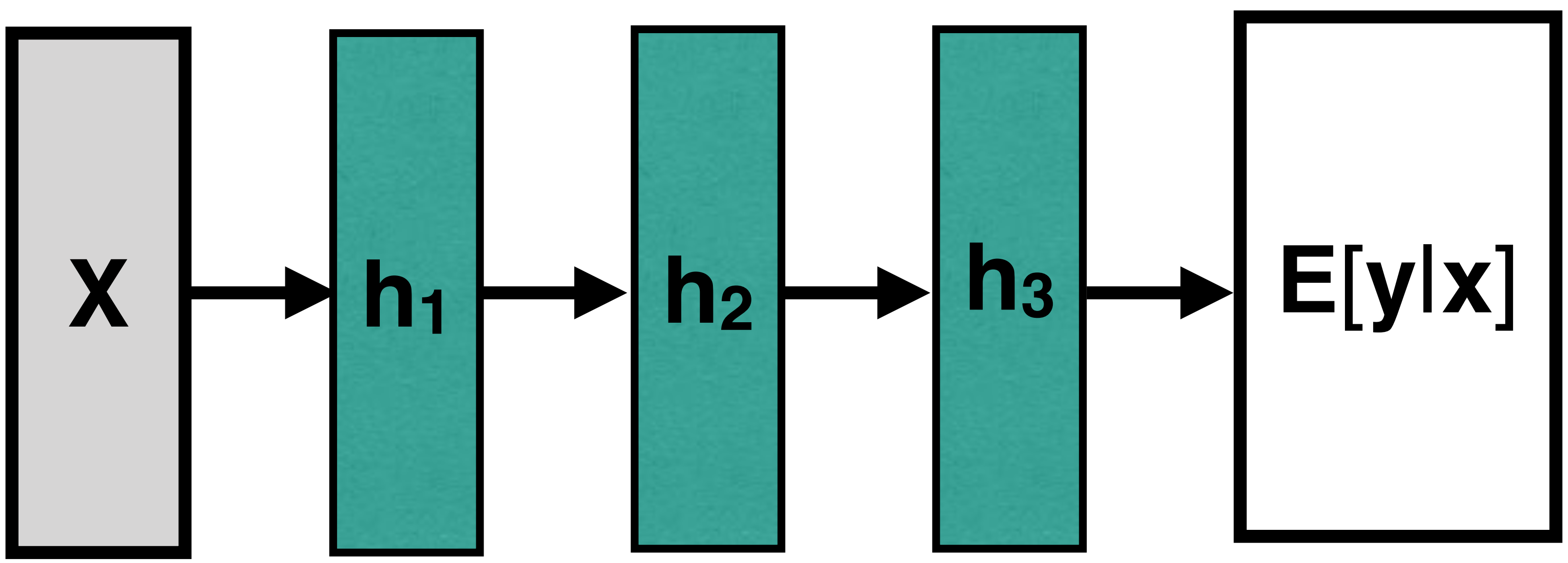
Layer index

$$\mathbf{y} \sim p(\mathbf{y} \,|\, \mathbf{X}, \{\boldsymbol{\theta}_l\}_{l=1}^{L})$$

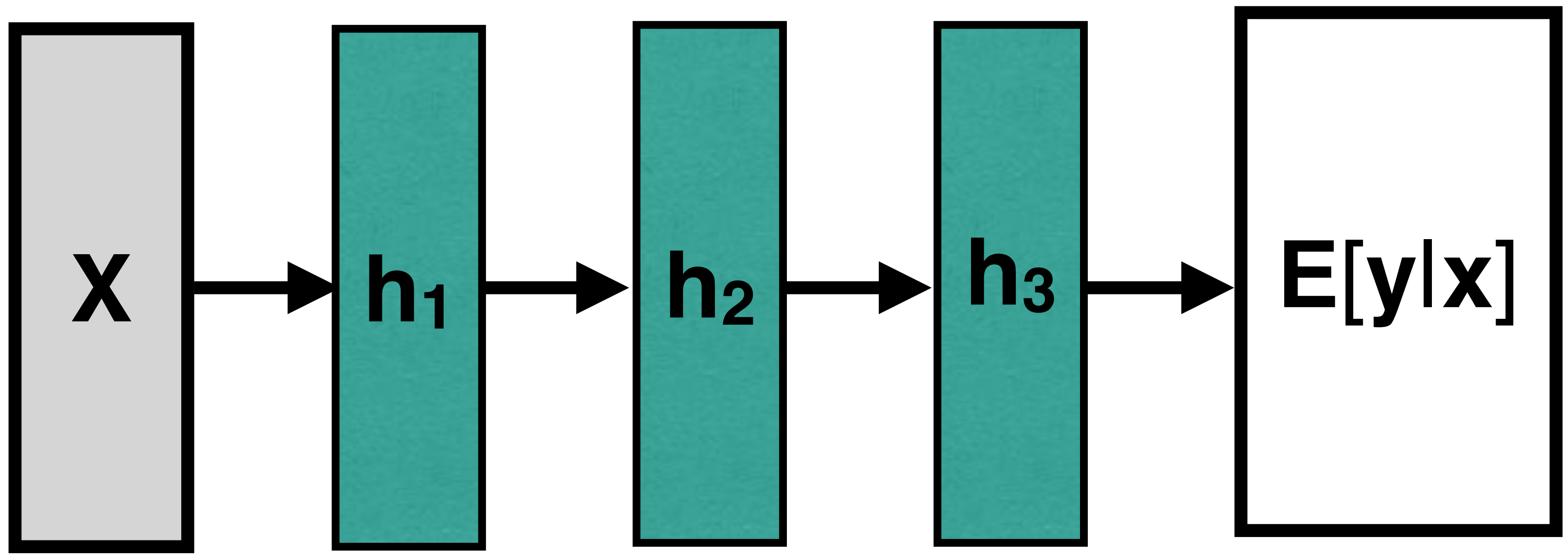
Layer-Wise Prior for NNs



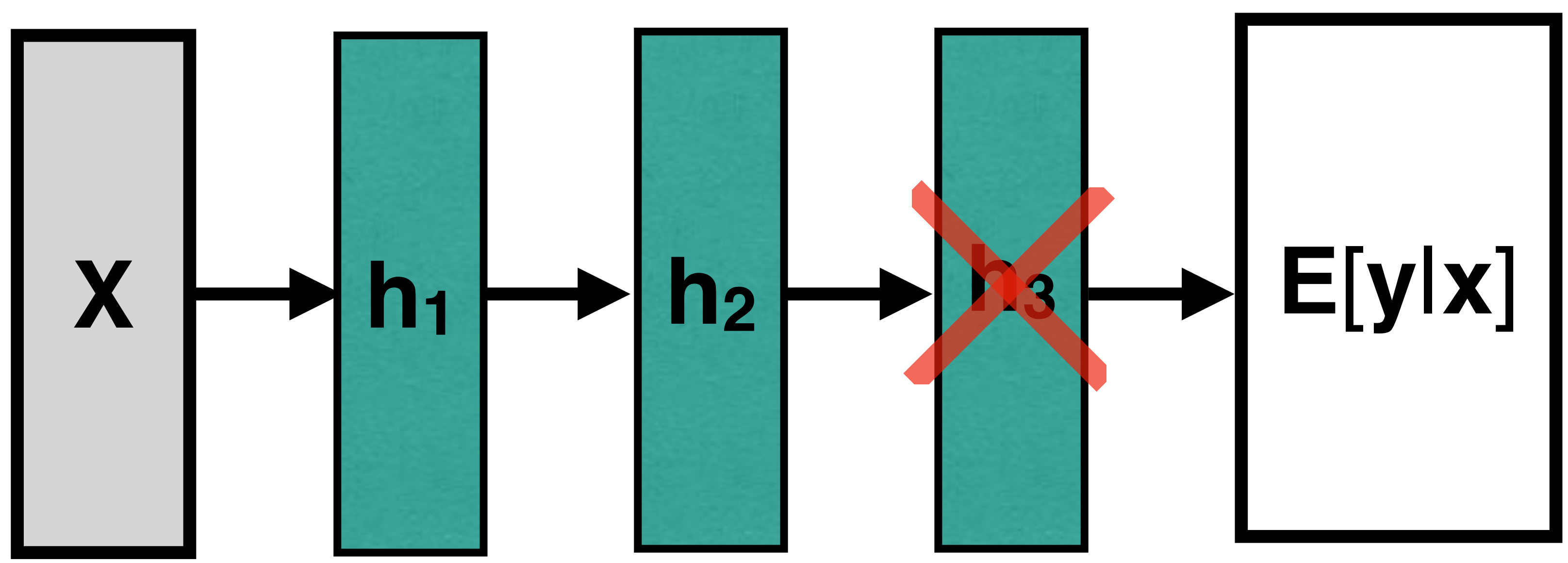
Self-referential reference model:



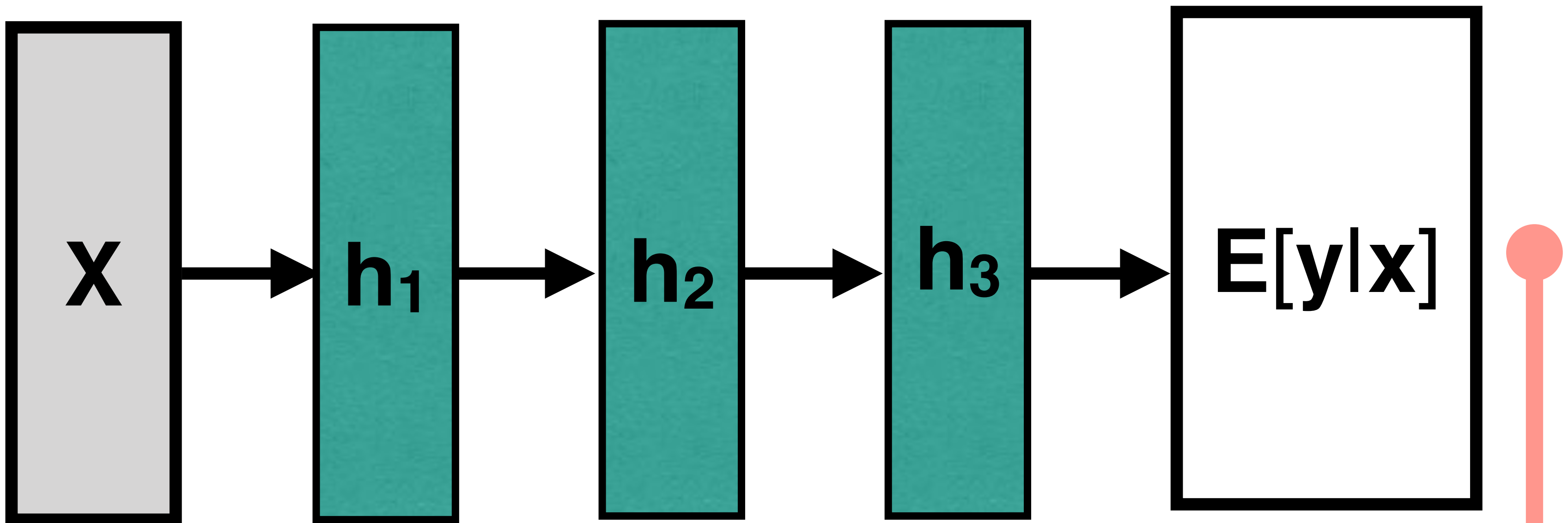
Layer-Wise Prior for NNs



Self-referential reference model:



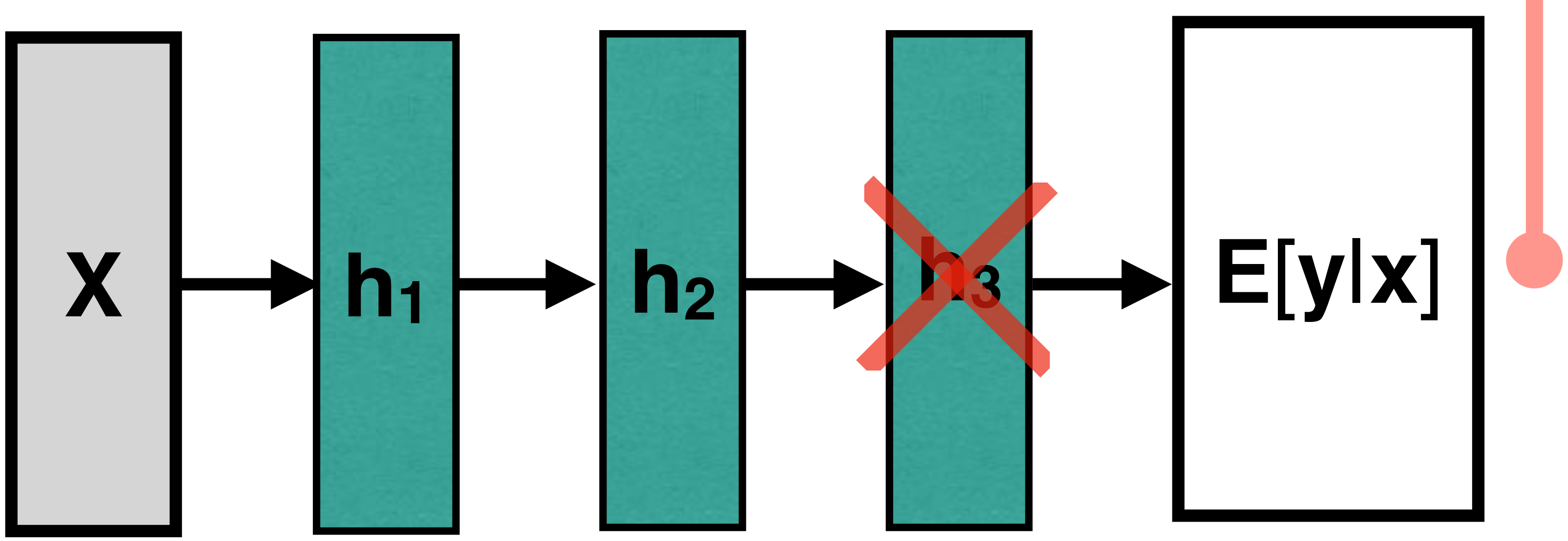
Layer-Wise Prior for NNs



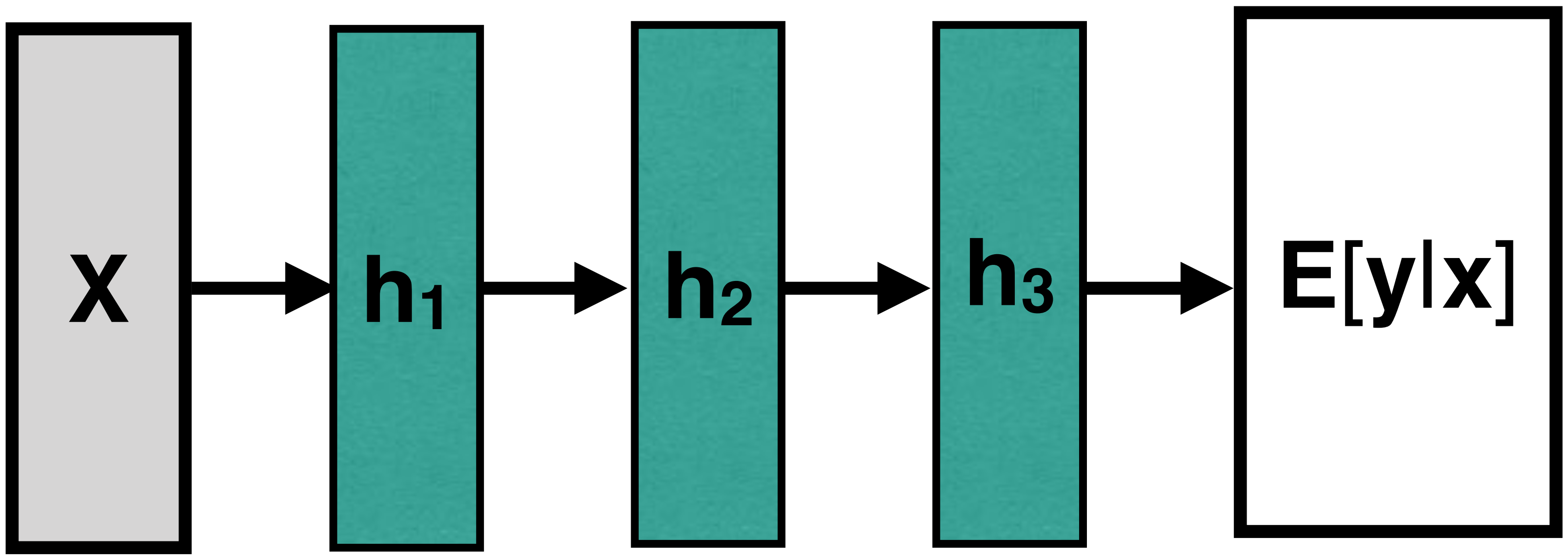
$$\mathbb{D}[p_l \mid\mid p_{l-1}]$$

Divergence measures the additional capacity afforded by the extra layer

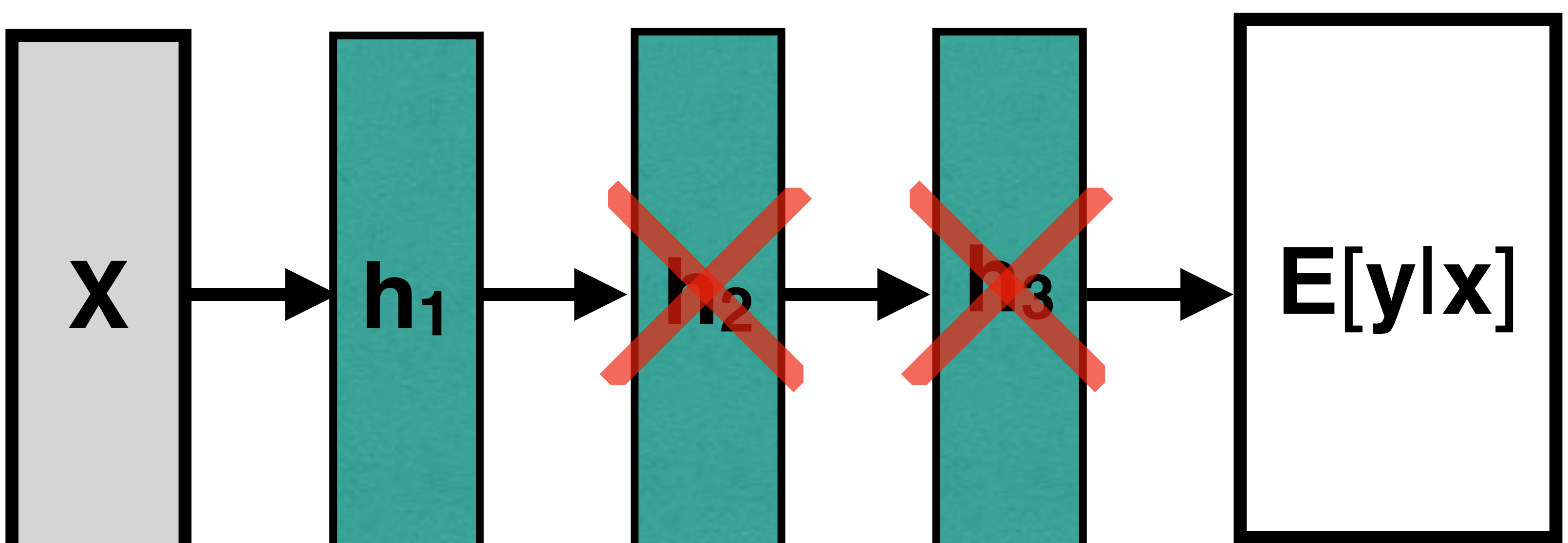
Self-referential reference model:



Layer-Wise Prior for NNs



Self-referential reference model:



Layer-Wise Prior for NNs

Joint Prior:

$$\pi(\tau_1, \ldots, \tau_L) = \pi(\tau_1) \prod_{l=2}^{L} \pi(\tau_l | \tau_1, \ldots, \tau_{l-1})$$

Factorization nicely follows the NN's layer structure.

Layer-Wise Prior for NNs

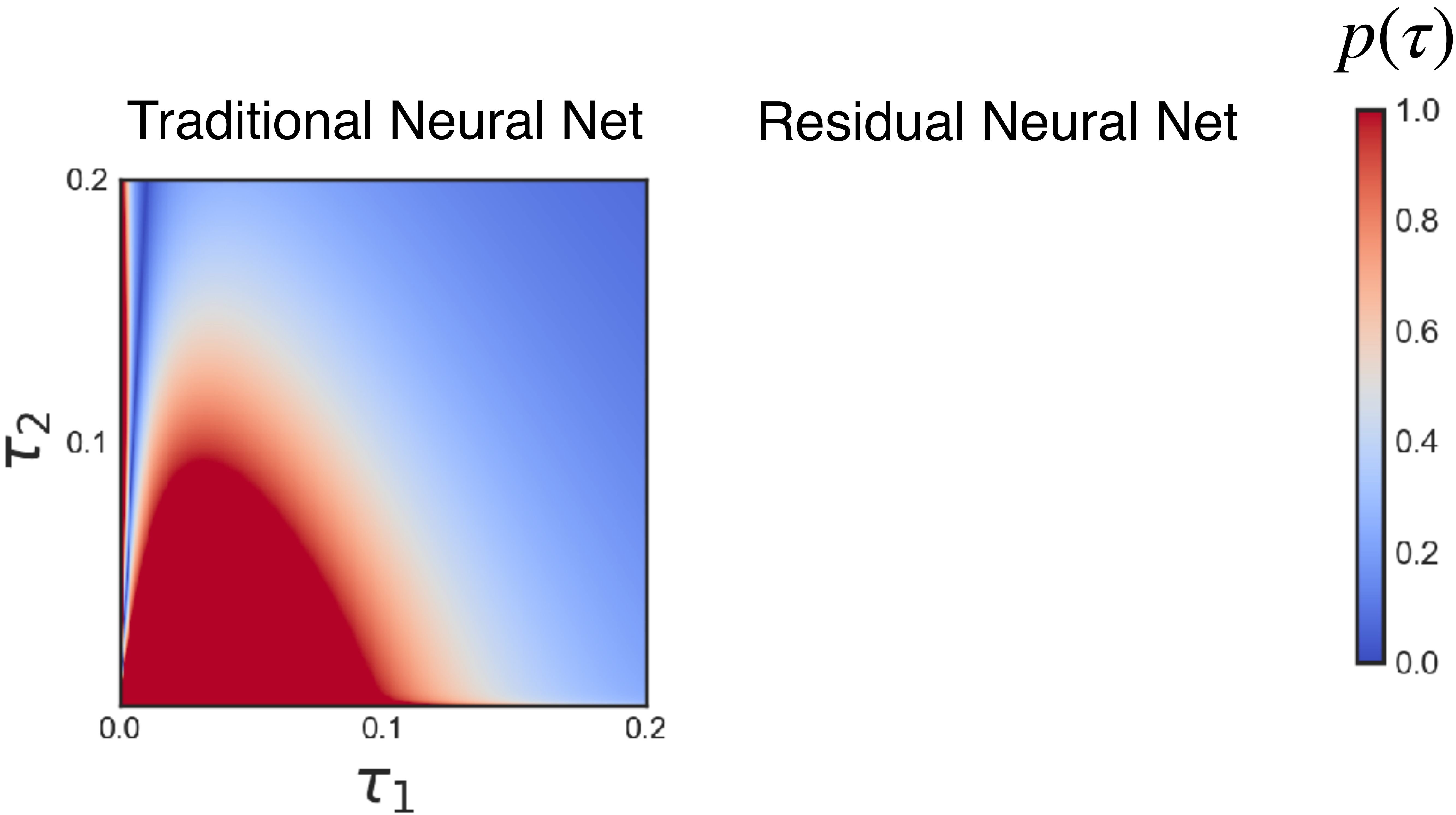
Traditional Neural Net

$$\mathbf{h}_{l+1} = F(\mathbf{h}_l)$$

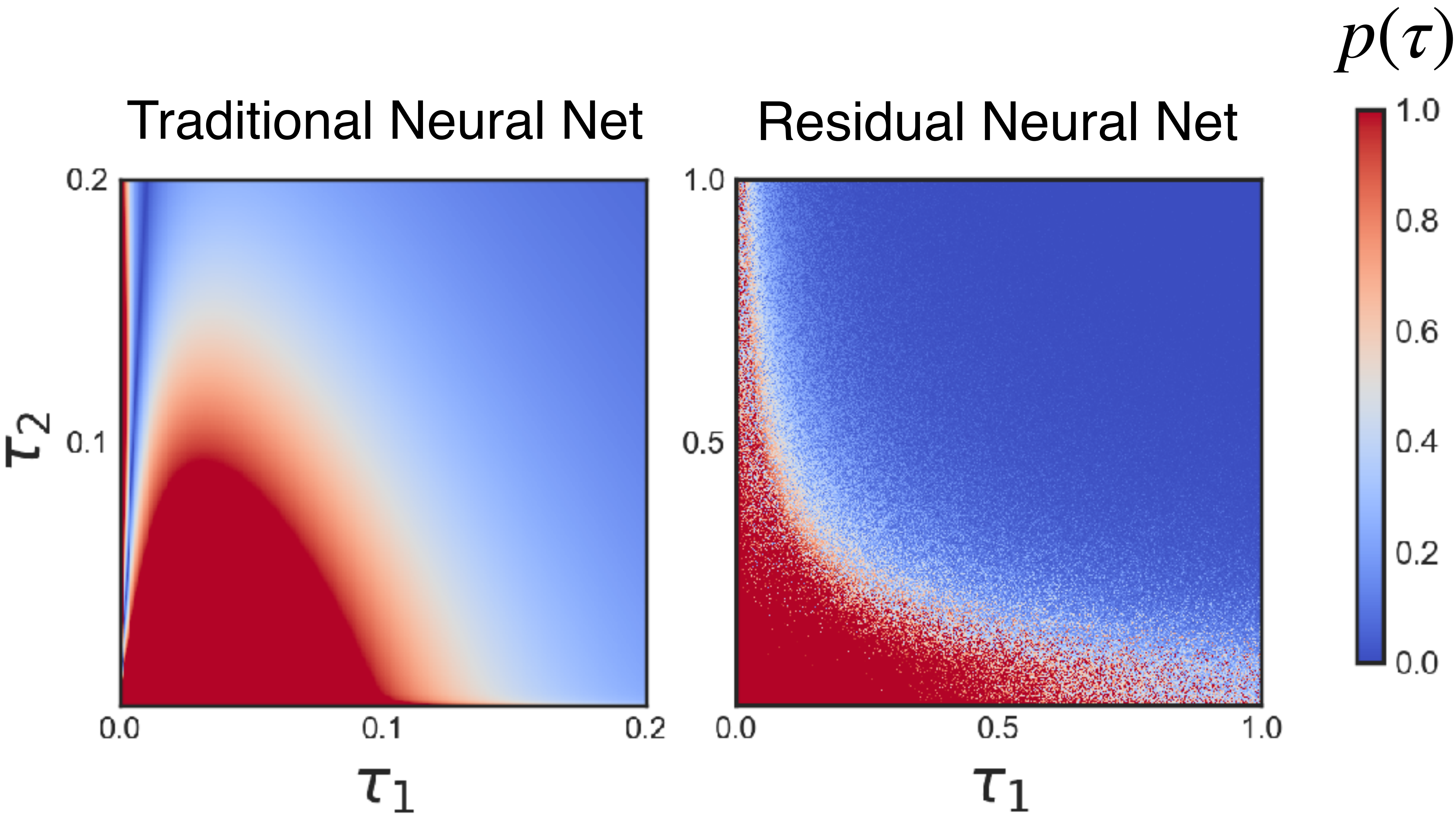
Residual Neural Net

$$\mathbf{h}_{l+1} = F(\mathbf{h}_l) + \mathbf{h}_l$$

Layer-Wise Prior for NNs



Layer-Wise Prior for NNs



Layer-Wise Prior for NNs

Table 2: *ARD-ADD Resnet*. Below we report test set RMSE for UCI benchmarks, comparing the PredCP against a shrinkage prior [37] and a fixed scale. Results are averaged across 20 splits.

Prior Type	boston	concrete	energy	kin8nm	power	wine	yacht
FIXED							
SHRINKAGE [37]							
PREDCP							

Layer-Wise Prior for NNs

Table 2: *ARD-ADD Resnet*. Below we report test set RMSE for UCI benchmarks, comparing the PredCP against a shrinkage prior [37] and a fixed scale. Results are averaged across 20 splits.

Prior Type	boston	concrete	energy	kin8nm	power	wine	yacht
Fixed	2.29 ±.33	**3.51** ±.41	0.83 ±.14	0.06 ±.00	3.32 ±.09	0.58 ±.04	0.66 ±.12
Shrinkage [37]	2.37 ±.18	3.76 ±.23	0.85 ±.08	0.06 ±.00	**3.24** ±.07	**0.54** ±.03	0.60 ±.16
PredCP	**2.26** ±.06	3.70 ±.46	**0.82** ±.07	0.06 ±.00	3.27 ±.09	0.56 ±.03	**0.57** ±.03

Application to Meta-Learning

Few-Shot Learning

Few-Shot Learning

$$\left\{\mathbf{y}_t, \mathbf{X}_t\right\}_{t=1}^{T}$$

Few-Shot Learning

[Chen et al., 2019] propose the model:

$$\mathbf{y}_t \sim p(\mathbf{y}_t \mid \mathbf{X}_t, \boldsymbol{\theta}_t),$$

Task-specific parameters

Few-Shot Learning

[Chen et al., 2019] propose the model:

$$\mathbf{y}_t \sim p(\mathbf{y}_t \mid \mathbf{X}_t, \boldsymbol{\theta}_t), \quad \boldsymbol{\theta}_t \sim N(\boldsymbol{\phi}, \tau \mathbb{I})$$

Task-specific parameters

Global, task-agnostic parameters

Few-Shot Learning

Full Model:

$$\mathbf{y}_t \sim p(\mathbf{y}_t \,|\, \mathbf{X}_t, \boldsymbol{\theta}_t)$$

Reference Model:

$$\mathbf{y}_t \sim p(\mathbf{y}_t \,|\, \mathbf{X}_t, \boldsymbol{\phi})$$

Few-Shot Learning

Full Model:

$$\mathbf{y}_t \sim p(\mathbf{y}_t \,|\, \mathbf{X}_t, \boldsymbol{\theta}_t)$$

Divergence represents how much information is lost when we use the task-agnostic parameters

Reference Model:

$$\mathbf{y}_t \sim p(\mathbf{y}_t \,|\, \mathbf{X}_t, \boldsymbol{\phi})$$

Few-Shot Learning

	FEWSHOT-CIFAR100	
	1-SHOT	5-SHOT
MAML		
σ-MAML + uniform prior [11]		
σ-MAML + shrinkage prior		
σ-MAML + **PredCP**		

Few-Shot Learning

	FEWSHOT-CIFAR100	
	1-SHOT	5-SHOT
MAML	35.6 ± 1.8	50.3 ± 0.9
σ-MAML + uniform prior [11]	39.3 ± 1.8	51.0 ± 1.0
σ-MAML + shrinkage prior	40.9 ± 1.9	52.7 ± 0.9
σ-MAML + **PredCP**	$\mathbf{41.2} \pm 1.8$	$\mathbf{52.9} \pm 0.9$

Few-Shot Learning

	FEWSHOT-CIFAR100	
	1-SHOT	5-SHOT
MAML	35.6 ± 1.8	50.3 ± 0.9
σ-MAML + uniform prior [11]	39.3 ± 1.8	51.0 ± 1.0
σ-MAML + shrinkage prior	40.9 ± 1.9	52.7 ± 0.9
σ-MAML + **PredCP**	$\mathbf{41.2} \pm 1.8$	$\mathbf{52.9} \pm 0.9$

	MINI-IMAGENET	
	1-SHOT	5-SHOT
MAML		
σ-MAML + uniform prior [11]		
σ-MAML + shrinkage prior		
σ-MAML + **PredCP**		

Few-Shot Learning

	FEWSHOT-CIFAR100 1-SHOT	FEWSHOT-CIFAR100 5-SHOT
MAML	35.6 ± 1.8	50.3 ± 0.9
σ-MAML + uniform prior [11]	39.3 ± 1.8	51.0 ± 1.0
σ-MAML + shrinkage prior	40.9 ± 1.9	52.7 ± 0.9
σ-MAML + **PredCP**	$\mathbf{41.2} \pm 1.8$	$\mathbf{52.9} \pm 0.9$

	MINI-IMAGENET 1-SHOT	MINI-IMAGENET 5-SHOT
MAML	46.8 ± 1.9	58.4 ± 0.9
σ-MAML + uniform prior [11]	47.7 ± 0.7	60.1 ± 0.8
σ-MAML + shrinkage prior	48.5 ± 1.9	60.9 ± 0.7
σ-MAML + **PredCP**	$\mathbf{49.3} \pm 1.8$	$\mathbf{61.9} \pm 0.9$

ONGOING WORK:

Bayesian Updating

Bayesian Updating

Consider Bayesian updating under the PredCP.

Bayesian Updating

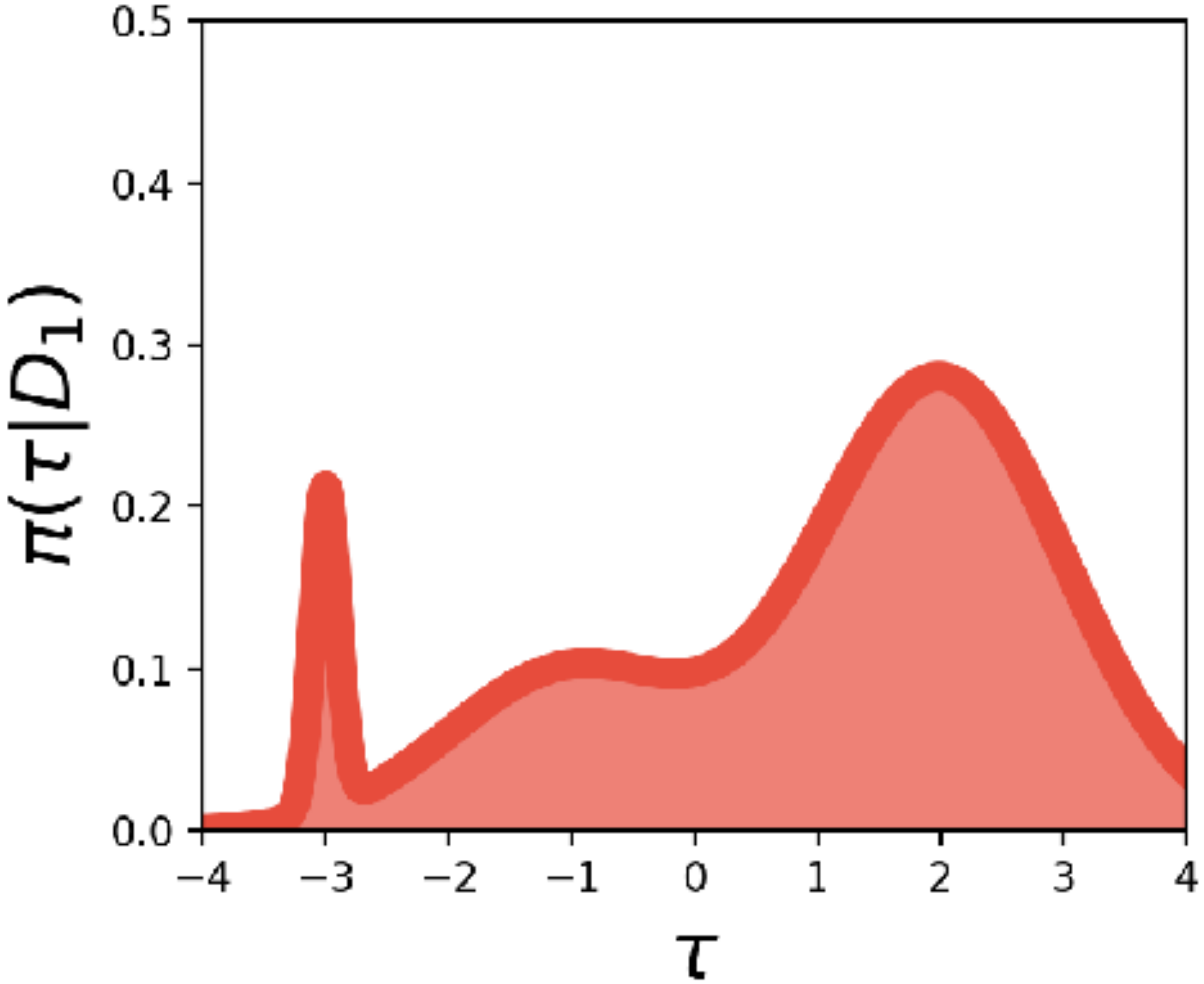
Consider Bayesian updating under the PredCP.

It's a bit weird because the PredCP conditions the model on the first set of features.

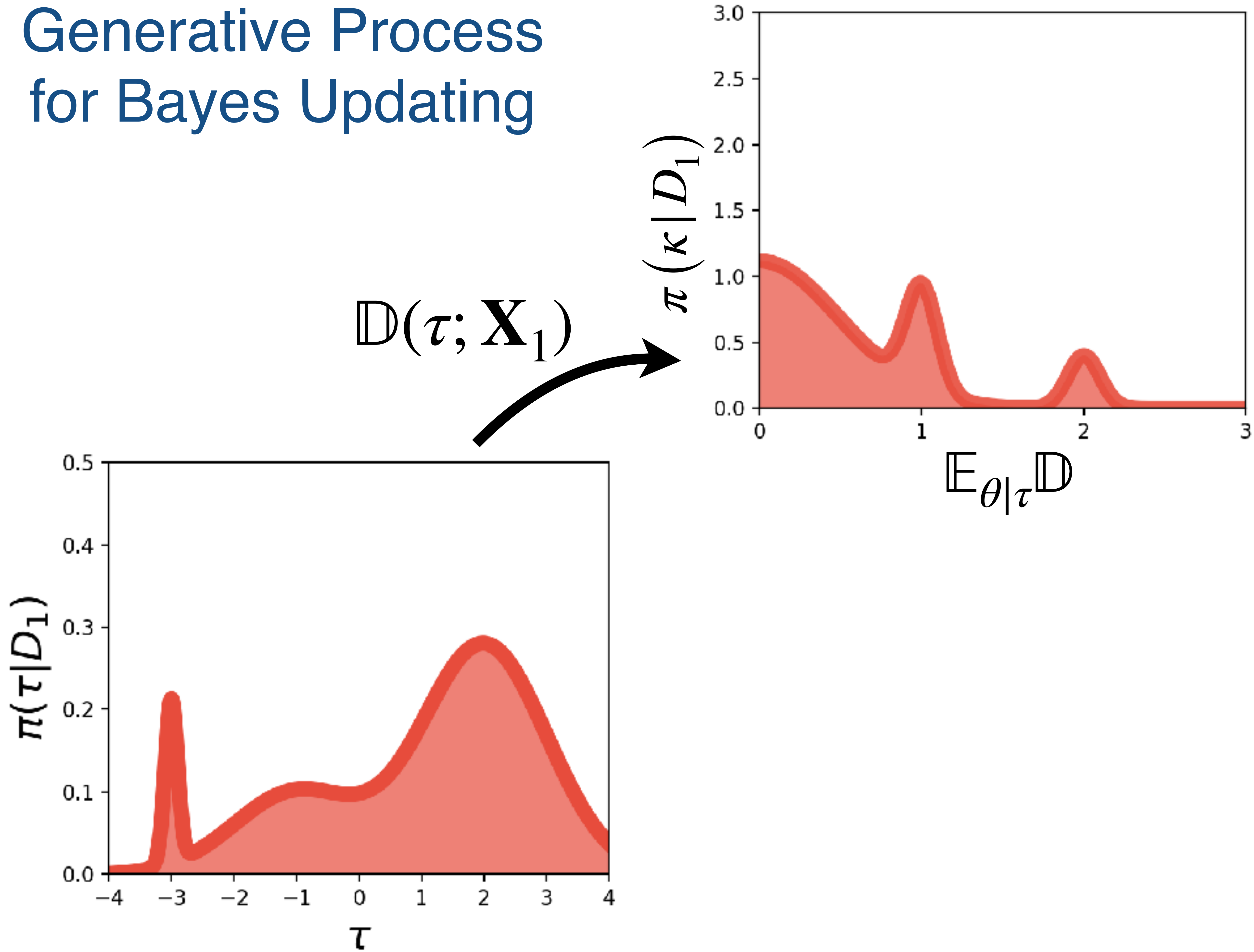
$$\tau \sim p(\tau; \mathbf{X}_1)$$

But shouldn't we also account for subsequent feature observations?

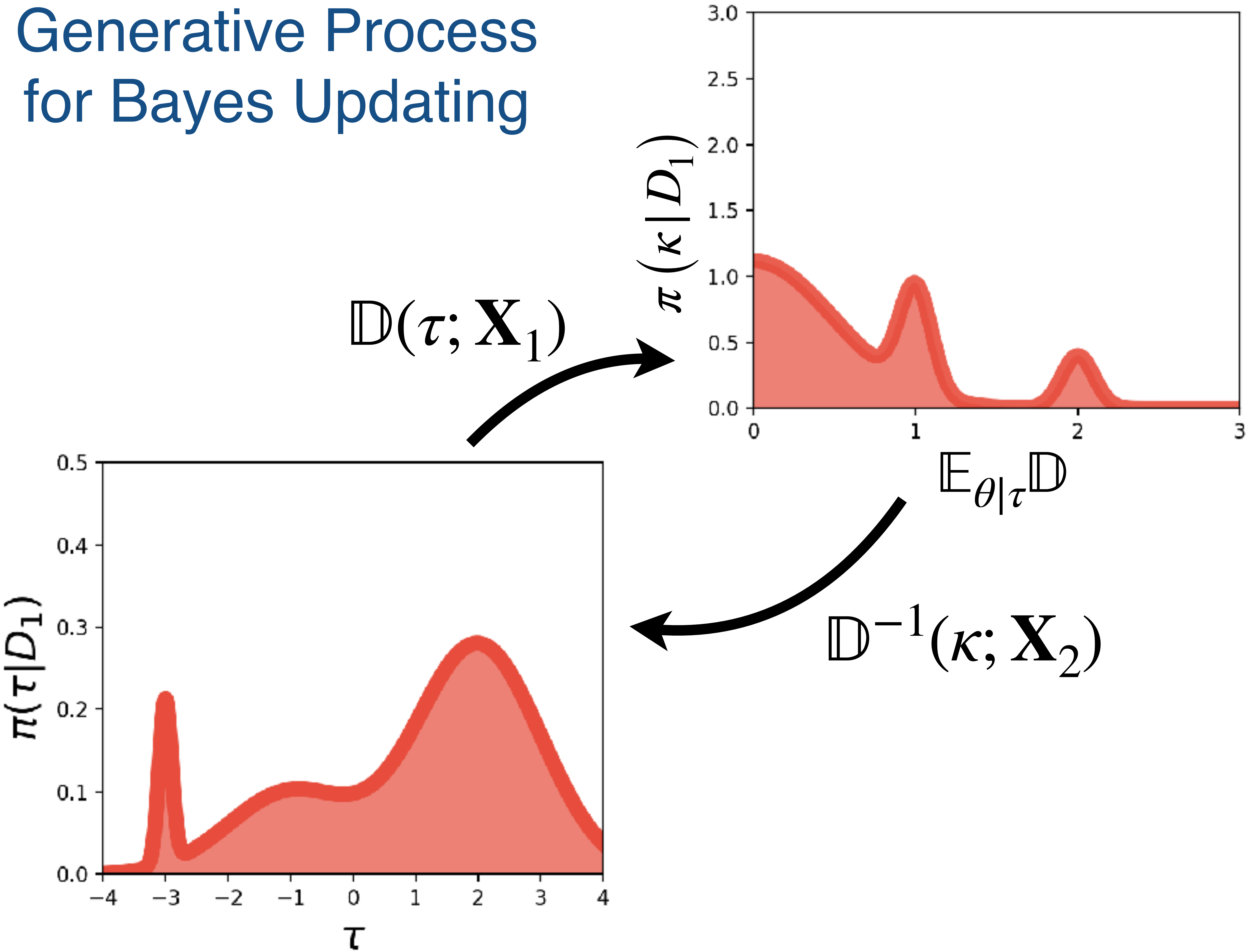
Generative Process for Bayes Updating



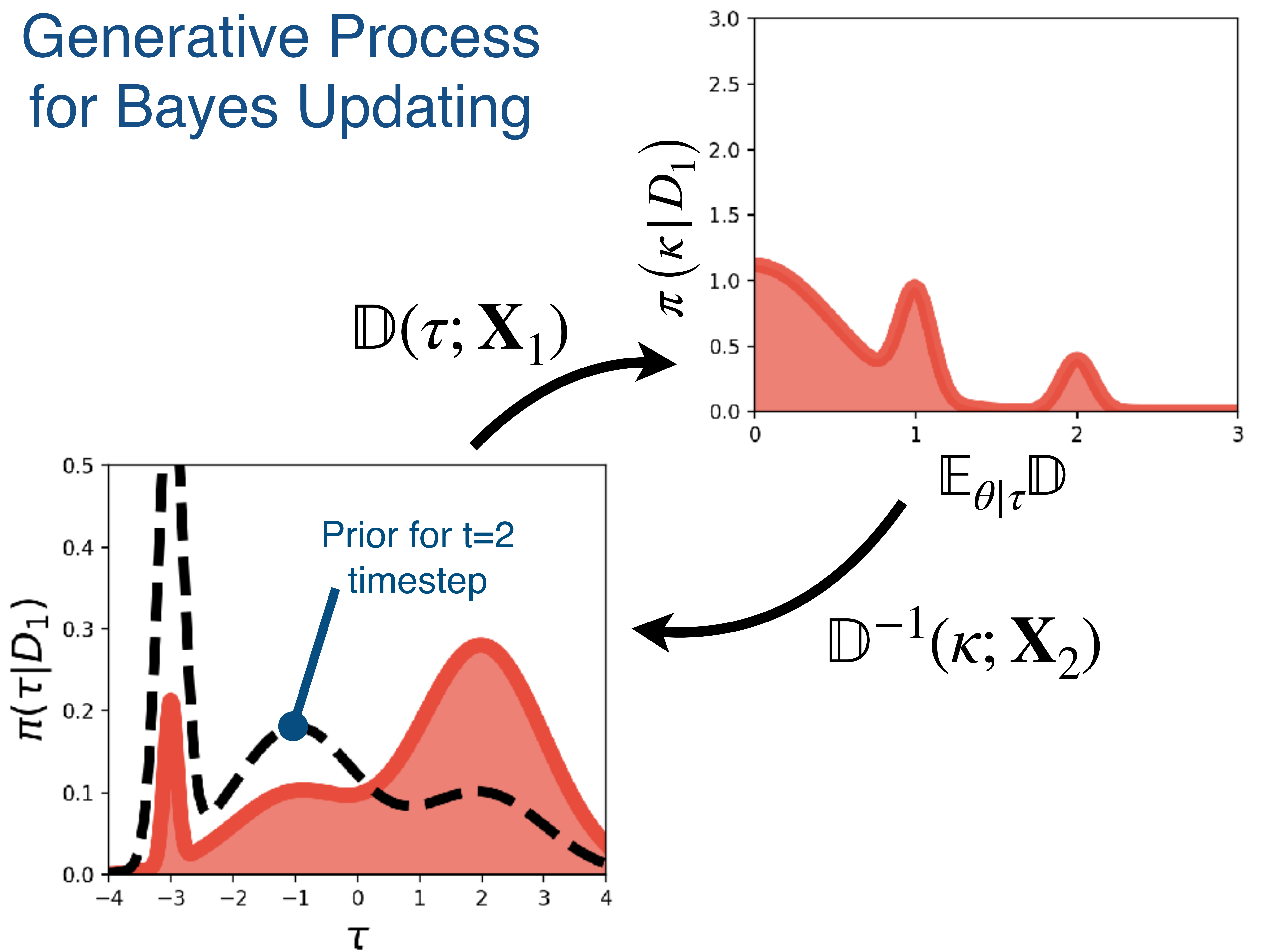
Generative Process for Bayes Updating

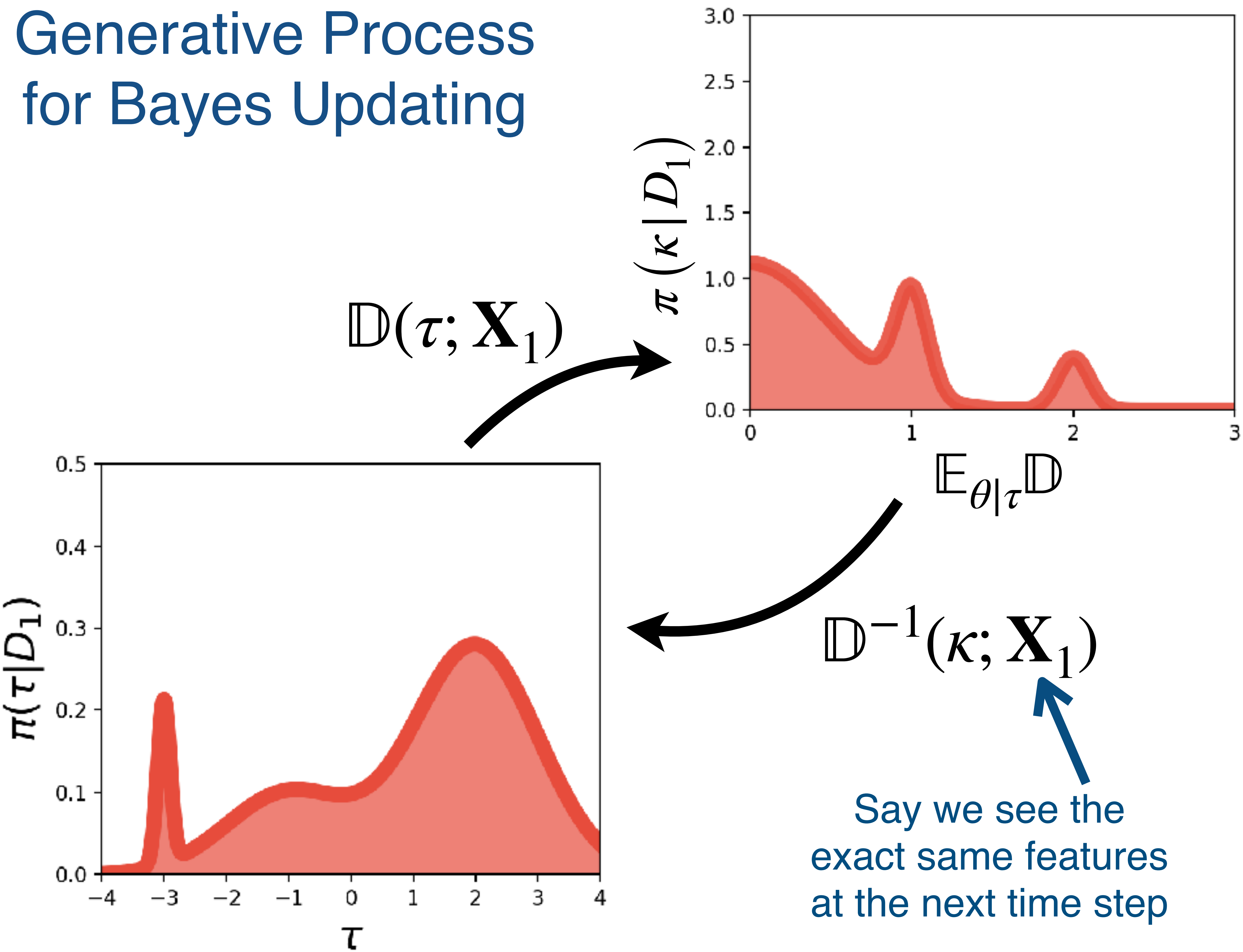


Generative Process for Bayes Updating



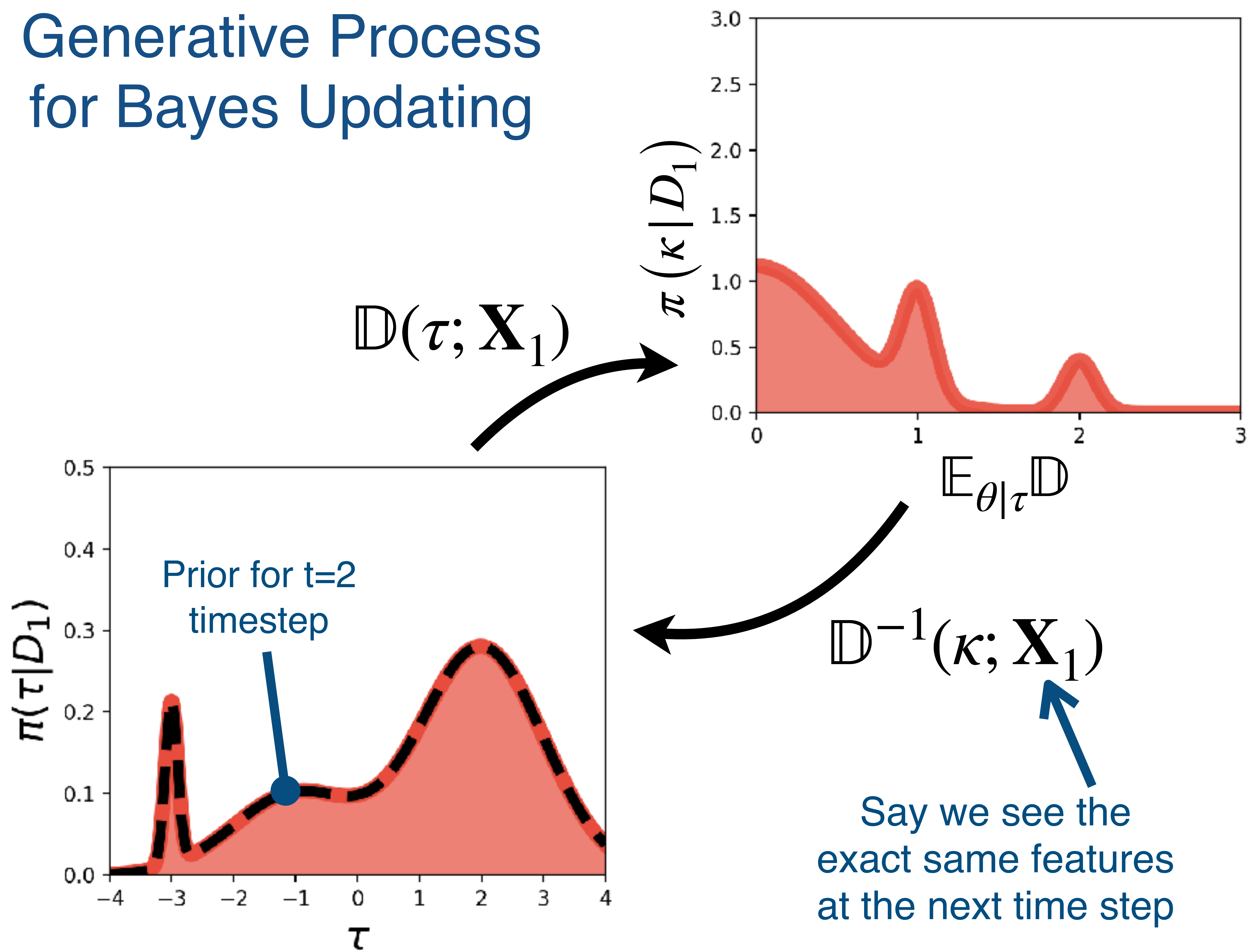
Generative Process for Bayes Updating





Generative Process
for Bayes Updating
$\mathbb{D}(\tau; \mathbf{X}_1)$
$\pi(\kappa \mid D_1)$
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
$\mathbb{E}_{\theta|\tau}\mathbb{D}$
$\mathbb{D}^{-1}(\kappa; \mathbf{X}_1)$
$\pi(\tau|D_1)$
0.5
0.4
0.3
0.2
0.1
0.0
−4
−3
−2
−1
0
1
2
3
4
τ
Say we see the
exact same features
at the next time step

Generative Process for Bayes Updating



Simple Example: Linear Regression

Simple Example: Linear Regression

For the shrinkage regression model, we get the t=2 prior (t=1 posterior):

$$\pi(\boldsymbol{\tau} \mid \mathcal{D}_1; \boldsymbol{x}_2) =$$

Simple Example: Linear Regression

For the shrinkage regression model, we get the t=2 prior (t=1 posterior):

$$\pi(\boldsymbol{\tau} \mid \mathcal{D}_1; \boldsymbol{x}_2) = p\left(\mathbb{D}_{X_2}^{-1} \circ \mathbb{D}_{X_1}(\tau) \;\middle|\; \mathcal{D}_1 \right) \left| \frac{\partial \; \mathbb{D}_{X_2}^{-1} \circ \mathbb{D}_{X_1}(\tau)}{\partial \tau} \right|$$

Simple Example: Linear Regression

For the shrinkage regression model, we get the t=2 prior (t=1 posterior):

$$\pi(\boldsymbol{\tau} \mid \mathcal{D}_1; \boldsymbol{x}_2) = p\left(\mathbb{D}_{X_2}^{-1} \circ \mathbb{D}_{X_1}(\tau) \;\middle|\; \mathcal{D}_1 \right) \left| \frac{\partial\, \mathbb{D}_{X_2}^{-1} \circ \mathbb{D}_{X_1}(\tau)}{\partial \tau} \right|$$

$$= p\left(\frac{N_2 \sum_{n=1}^{N_1} x_{1,n}^2}{N_1 \sum_{n=1}^{N_2} x_{2,n}^2} \cdot \boldsymbol{\tau} \;\middle|\; \mathcal{D}_1 \right) \left| \frac{N_2 \sum_{n=1}^{N_1} x_{1,n}^2}{N_1 \sum_{n=1}^{N_2} x_{2,n}^2} \right|$$

Simple Example: Linear Regression

For the shrinkage regression model, we get the t=2 prior (t=1 posterior):

$$\pi(\boldsymbol{\tau} \mid \mathcal{D}_1; \boldsymbol{x}_2) = p\left(\mathbb{D}_{X_2}^{-1} \circ \mathbb{D}_{X_1}(\tau) \,\middle|\, \mathcal{D}_1 \right) \left| \frac{\partial\, \mathbb{D}_{X_2}^{-1} \circ \mathbb{D}_{X_1}(\tau)}{\partial \tau} \right|$$

$$= p\left(\frac{N_2 \sum_{n=1}^{N_1} x_{1,n}^2}{N_1 \sum_{n=1}^{N_2} x_{2,n}^2} \cdot \boldsymbol{\tau} \,\middle|\, \mathcal{D}_1 \right) \left| \frac{N_2 \sum_{n=1}^{N_1} x_{1,n}^2}{N_1 \sum_{n=1}^{N_2} x_{2,n}^2} \right|$$

Ratio will be ~1 when features have similar second moments

Simple Example: Linear Regression

For the shrinkage regression model, we get the t=2 prior (t=1 posterior):

$$\pi(\boldsymbol{\tau} \mid \mathcal{D}_1; \boldsymbol{x}_2) = p\left(\mathbb{D}_{X_2}^{-1} \circ \mathbb{D}_{X_1}(\tau) \;\middle|\; \mathcal{D}_1 \right) \left| \frac{\partial\, \mathbb{D}_{X_2}^{-1} \circ \mathbb{D}_{X_1}(\tau)}{\partial \tau} \right|$$

$$= p\left(\frac{N_2 \sum_{n=1}^{N_1} x_{1,n}^2}{N_1 \sum_{n=1}^{N_2} x_{2,n}^2} \cdot \boldsymbol{\tau} \;\middle|\; \mathcal{D}_1 \right) \left| \frac{N_2 \sum_{n=1}^{N_1} x_{1,n}^2}{N_1 \sum_{n=1}^{N_2} x_{2,n}^2} \right|$$

Ratio will be ~1 when features have similar second moments

Of course, we usually standardize the first two moments anyway (z-scoring).

Layer-Wise Prior for ResNets

$$\mathbb{D}_t^{-1} \circ \mathbb{D}_{t-1}(\tau_l) = \frac{N_t}{N_{t-1}} \frac{\sum_{n=1}^{N_{t-1}} \mathrm{Var}_{\tilde{\mathbf{W}}, \mathbf{W}_o | \mathcal{D}} \left[f_l(\mathbf{h}_{t-1,n,l-1} \tilde{\mathbf{W}}_l) \mathbf{W}_o \right]}{\sum_{n'=1}^{N_t} \mathrm{Var}_{\tilde{\mathbf{W}}, \mathbf{W}_o | \mathcal{D}} \left[f_l(\mathbf{h}_{t,n,l-1} \tilde{\mathbf{W}}_l) \mathbf{W}_o \right]} \boldsymbol{\tau}_l$$

Ratio of (prior)
predictive variances

Future Work

One downside of the current formulation is that dependence across data points is not accounted for.

$$\mathbb{E}_{\theta|\tau}\mathbb{KLD}\left[p_\theta \,||\, p_\phi\right] = \sum_{n=1}^{N} \mathbb{E}_{\theta|\tau}\mathbb{KLD}\left[p(\mathsf{y}\,|\,\mathsf{x}_n, \theta) \,||\, p(\mathsf{y}\,|\,\mathsf{x}_n, \phi)\right]$$

Future Work

One downside of the current formulation is that dependence across data points is not accounted for.

$$\mathbb{E}_{\theta|\tau}\mathbb{KLD}\left[p_\theta \;||\; p_\phi\right] = \sum_{n=1}^{N} \mathbb{E}_{\theta|\tau}\mathbb{KLD}\left[p(\mathsf{y} \,|\, \mathsf{x}_n, \theta) \;||\; p(\mathsf{y} \,|\, \mathsf{x}_n, \phi)\right]$$

Ideally, we want to compute:

$$\mathbb{KLD}\left[\mathbb{E}_{\theta|\tau}[p(\mathsf{Y} \,|\, \mathsf{X}, \theta)] \;||\; p(\mathsf{Y} \,|\, \mathsf{X}, \phi)\right]$$

Summary

- ⊗ **Framework** for specifying priors using a reference model
- ⊗ **Reparametrization** allows us to think about whole models but then transfer beliefs to parameters.
- ⊗ Reference models can be constructed by **exploiting the compositional nature of NNs** (eg layers)

Thank you. Questions?

arxiv.org/abs/2006.10801

2021

AISTATS

Predictive Complexity Priors

Eric Nalisnick
University of Amsterdam

Jonathan Gordon
University of Cambridge

José Miguel Hernández-Lobato
University of Cambridge

Abstract

Specifying a Bayesian prior is notoriously difficult for complex models such as neural networks. Reasoning about parameters is made challenging by the high-dimensionality and over-parameterization of the space. Pri-

has assumed the negative and resorted to priors of convenience. For instance, the standard normal distribution is by far the most popular prior for Bayesian NNs (Zhang et al., 2020; Heek and Kalchbrenner, 2019; Wenzel et al., 2020).

In this paper, we present a novel framework to specify priors for black-box models. Rather than working

enalisnick.github.io/

eric_nalisnick