Predictive Complexity Priors

Nalisnick, E.; Gordon, J.; Hernández-Lobato, J.M.

Publication date
2021
Document Version
Final published version
Published in
Proceedings of Machine Learning Research
License
Other

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. Priors that seem benign and uninformative can have unintuitive and detrimental effects on a model's predictions. For this reason, we propose predictive complexity priors: a functional prior that is defined by comparing the model's predictions to those of a reference model. Although originally defined on the model outputs, we transfer the prior to the model parameters via a change of variables. The traditional Bayesian workflow can then proceed as usual. We apply our predictive complexity prior to high-dimensional regression, reasoning over neural network depth, and sharing of statistical strength for few-shot learning.

1 INTRODUCTION

Choosing the prior for a Bayesian model is the most important—and often, the most difficult—step in model specification (Robert, 2001). Unfortunately, prior specification within machine learning is additionally fraught and challenging. Popular models such as neural networks (NNs) are high dimensional and unidentifiable, making it extremely hard to reason about what makes a good prior. Moreover, since the true posterior can almost never be recovered, it is difficult to isolate a prior's influence (even empirically). We are left asking: do the specifics of the prior even matter if they are blunted by our posterior approximations and large data sets? Until recently, most work in machine learning 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 with the uninterpretable parameter space, we place the Bayesian prior on the model's *functional complexity*. Our prior, termed the *predictive complexity prior* (PredCP), compares the model's predictions to those of a reference model. For example, we define the reference model for a NN to be a NN with one fewer layer. The PredCP can then assess and control the effect of depth on the model's capacity. Unlike previous work on functional priors (Sun et al., 2019), we use a change of variables to *exactly* translate the prior into a proper prior on the model parameters. Bayesian inference can then proceed as usual and without involving extra machinery. We claim the following contributions:

- **Methodology:** We propose *predictive complexity priors* (PredCPs). These extend Simpson et al. (2017)'s framework to the model predictions, thereby allowing our data-space intuitions to guide prior specification. Moreover, we introduce crucial modifications that allow the PredCP to scale to large, black-box models such as NNs.

- **Applications:** We demonstrate the wide applicability of the PredCP by using it for three disparate tasks: high-dimensional regression, reasoning over depth in Bayesian NNs, and sharing information across tasks for few-shot learning. For Bayesian NNs, we investigate the PredCP's behavior in detail, revealing its mechanism of action: regularizing predictive variance.

- **Experiments:** We report results across a variety of tasks (classification, regression, few-shot learning), models (logistic regression, NNs), and posterior inference strategies (Markov chain Monte Carlo, variational inference, MAP estimation). The PredCP provides consistent improvements

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

in predictive generalization over alternative priors (uninformative, shrinkage).

2 SETTING OF INTEREST

Notation Matrices are denoted with upper-case and bold letters (e.g. $\boldsymbol{Y}$), vectors with lower-case and bold (e.g. $\boldsymbol{y}$), and scalars with no bolding (e.g. y or Y). We use italics to differentiate observations and constants (e.g. $\boldsymbol{y}$, $\boldsymbol{\theta}$) from random variables (e.g. $\mathbf{y}$, $\boldsymbol{\theta}$).

Model We consider hierarchical models of the form:

$$\mathbf{y} \sim p(\mathbf{y}|\boldsymbol{\theta}), \quad \boldsymbol{\theta} \sim p(\boldsymbol{\theta}|\tau), \quad \tau \sim p(\tau) \tag{1}$$

where $p(\mathbf{y}|\boldsymbol{\theta})$ is the data (sampling) model, $p(\boldsymbol{\theta}|\tau)$ is a first-level prior, and $p(\tau)$ is a second-level hyper-prior. We are primarily concerned with models for which $p(\mathbf{y}|\boldsymbol{\theta})$ is parameterized by a complicated function and τ plays a significant role in controlling the complexity of that function. One such example is parameterizing $\mathbf{y}|\boldsymbol{\theta}$ with a NN whose weights are given a normal prior with variance τ: $\boldsymbol{\theta} \sim \mathrm{N}(\mathbf{0}, \tau\mathbb{I})$. As τ grows, the weights become less constrained and the model becomes more flexible. A common strategy for controlling τ is to give it a shrinkage prior such as a zero-favoring inverse gamma (Neal, 1994) or half-Cauchy (Carvalho et al., 2009). While this is a sensible approach, it can be hard to understand how $p(\tau)$ regularizes the downstream predictive function (Piironen and Vehtari, 2017).

A Sketch of Our Approach We propose a novel prior for τ that goes beyond simply encouraging its value to be small, as a shrinkage prior does. Instead, we control τ via model-based (Gelman et al., 2017) functional regularization. Inspired by Simpson et al. (2017),[1] we define a divergence function between the model of interest—denote it $p(\mathbf{y}|\tau)$ for now—and a reference model denoted $p_0(\mathbf{y})$, which does not depend on τ. Denote the divergence as $\kappa = \mathbb{D}[p(\mathbf{y}|\tau)||p_0(\mathbf{y})]$. We derive $p(\tau)$ by placing a prior on κ and *reparameterizing* w.r.t. τ: $\kappa \sim \pi(\kappa)$, $\tau = \mathbb{D}^{-1}(\kappa)$ where $\mathbb{D}^{-1}$ denotes the inverse of the aforementioned divergence function. The τ-prior's density function can then be written using the change of variables formula:

$$\begin{aligned} p(\tau) &= \pi(\kappa) \left|\frac{\partial \kappa}{\partial \tau}\right| \\ &= \pi\left(\mathbb{D}[p(\mathbf{y}|\tau)||p_0(\mathbf{y})]\right) \left|\frac{\partial\, \mathbb{D}[p(\mathbf{y}|\tau)||p_0(\mathbf{y})]}{\partial \tau}\right| \end{aligned} \tag{2}$$

where $|\partial\kappa/\partial\tau|$ is the absolute value of the divergence function's derivative w.r.t. τ. Note that $\mathbb{D} : \tau \mapsto \kappa$ must be differentiable and bijective for $p(\tau)$ to be proper (i.e. integrate to one).[2] For these conditions to be satisfied, τ must be a scalar since κ is a scalar.

These technical conditions aside, the crucial property of our framework is that the divergence is computed by integrating over $\mathbf{y}$, the random variable that corresponds to data. In turn, the prior can represent data-space intuitions (via $\pi(\kappa)$) and automatically translate them (via $\mathbb{D}$) into a prior on the model parameters. This direction runs counter to how priors are usually defined: by specifying $p(\tau)$ directly and having little-to-no information about the induced distribution on $\mathbf{y}$. Despite its dependence on the observation model, we emphasize that our prior is not an empirical Bayesian prior (Casella, 1985) since $\mathbf{y}$ is integrated out.

3 IDEALIZED SETTING

In order to implement the prior sketched in Equation 2, we need to make both theoretical choices (e.g. the divergence) and practical choices. We separate the description of our prior into two sections (3 and 4) so as to delineate theory from practice. In this section, we provide a full theoretical description. Section 4 then introduces some modifications that increase the scalability and broaden the applicability of our prior.

We call the idealized version the *evidence complexity prior* (ECP). It implements Equation 2 via the following three steps:

Step #1: Define Reference Model Given the model of interest in Equation 1, the first step is defining the reference model $p_0(\mathbf{y})$. Crucially, $\mathbf{y}$ denotes a *random variable* that corresponds to data, not an observation. Thus, we call $p_0(\mathbf{y})$ and $p(\mathbf{y}|\tau)$ 'evidence functions' (Bishop, 2006), not marginal likelihoods. In general, we define $p_0(\mathbf{y})$ by replacing the first-level prior $p(\boldsymbol{\theta}|\tau)$ with a less expressive prior $p_0(\boldsymbol{\theta})$. We obtain the final version of the reference model by marginalizing over $p_0(\boldsymbol{\theta})$:

$$p_0(\mathbf{y}) = \int_{\boldsymbol{\theta}} p(\mathbf{y}|\boldsymbol{\theta})\; p_0(\boldsymbol{\theta})\; d\boldsymbol{\theta}. \tag{3}$$

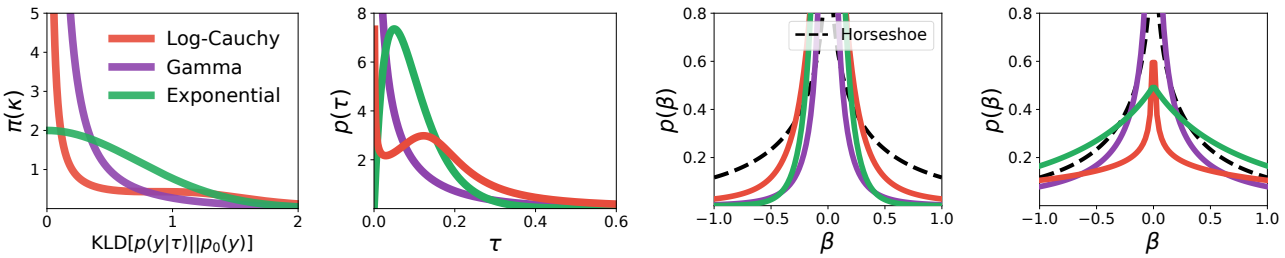
We will exclusively use a point-mass prior $p_0(\boldsymbol{\theta}) = \delta(|\boldsymbol{\theta} - \boldsymbol{\theta}_0|)$ so that this integration is made trivial: $p_0(\mathbf{y}) = p(\mathbf{y}|\boldsymbol{\theta}_0)$. To form a corresponding representation of our model of interest, we marginalize over $p(\boldsymbol{\theta}|\tau)$:

$$p(\mathbf{y}|\tau) = \int_{\boldsymbol{\theta}} p(\mathbf{y}|\boldsymbol{\theta})\; p(\boldsymbol{\theta}|\tau)\; d\boldsymbol{\theta}. \tag{4}$$

For many models, it will be hard to perform the above integration, which is why we refer to the ECP's construction as 'idealized.'

[1] We provide a detailed summary of and comparison to Simpson et al. (2017)'s framework in Section 6.

[2] We discuss bijectivity conditions for NNs in Section 5.



(a) KLD Priors (b) ECPs, $x = 1$ (c) Marginals, $x = 1$ (d) Marginals, $x = 0.25$

Figure 1: *ECP for Linear Regression.* Subfigure (a) shows each KLD prior: exponential ($\lambda = .5$), gamma ($\lambda = (.2, 2)$), and log-Cauchy ($\lambda = 1$). Subfigure (b) shows the corresponding ECP on τ. Subfigure (c) shows the marginal prior on β induced by each ECP from (b). Subfigure (c) shows the same marginals for $x = 0.25$. The horseshoe prior (Carvalho et al., 2009) (black dashed line) is shown for reference. The ECP adapts with the input feature, resulting in dynamic shrinkage properties.

Step #2: Define Divergence We next choose the divergence function. Following Simpson et al. (2017), we use the Kullback–Leibler divergence (KLD):

$$\mathrm{KL}\left[p(\mathbf{y}|\tau) \;||\; p_0(\mathbf{y})\right] = \int_{\mathbf{y}} p(\mathbf{y}|\tau) \; \log \frac{p(\mathbf{y}|\tau)}{p_0(\mathbf{y})} \, d\mathbf{y}. \quad (5)$$

We choose this particular KLD—using $p(\mathbf{y}|\tau)$ as the first argument—because it represents the bits lost when we approximate $p(\mathbf{y}|\tau)$ with the reference model. Switching the arguments would not be sensible since $\mathrm{KL}\left[p_0(\mathbf{y}) \;||\; p(\mathbf{y}|\tau)\right]$ quantifies the bits lost when $p(\mathbf{y}|\tau)$ approximates $p_0(\mathbf{y})$, which should be easy to do since $p(\mathbf{y}|\tau)$ is more expressive. Other choices of divergence (e.g. Hellinger) are possible, but in Section 4, we will require that the divergence be a convex function w.r.t. $p(\mathbf{y}|\tau)$.

Step #3: Reparameterize Lastly, we place a prior on the divergence: $\pi(\kappa) = \pi(\mathrm{KL}\left[p(\mathbf{y}|\tau) \;||\; p_0(\mathbf{y})\right])$. The support of $\pi(\kappa)$ should be $\mathbb{R}^{\geq 0}$ to match the KLD's codomain.[3] The degree to which $\pi(\kappa)$ favors $\kappa = 0$ represents our preference for the simpler reference model. As $\pi(\kappa)$ allocates more density away from zero, the functional regularization is relaxed. Performing the change of variables then yields the ECP on τ:

$$p\left(\tau\right) = \pi\left(\mathrm{KL}\left[p(\mathbf{y}|\tau) \;||\; p_0(\mathbf{y})\right]\right) \times \left| \frac{\partial \; \mathrm{KL}\left[p(\mathbf{y}|\tau) \;||\; p_0(\mathbf{y})\right]}{\partial \; \tau} \right|. \quad (6)$$

The crucial characteristics of the ECP are that it compares the models holistically and in data space. Using the evidence functions (step #1) allows the ECP to directly assess how τ affects $\mathbf{y}$. Hence the ECP embraces the philosophy that the prior can only be understood in the context of the likelihood (Gelman et al., 2017). Computing the KLD (step #2) then provides a functional comparison of how the models allocate probability in data space. This data-space behavior is our ultimate concern for black-box models.

3.1 Example: Linear Regression

We continue discussion of the ECP with a concrete example. Consider the linear model $\mathbb{E}[y|x, \beta] = x\beta$, where $y \in \mathbb{R}$ denotes a scalar response, $x \in \mathbb{R}$ its covariate / feature, and $\beta \in \mathbb{R}$ the model parameter. While this is undoubtedly a simple example, the priors we later describe for NNs will have commonalities. Let us now step through the ECP derivation. We choose the first-level prior to be normal, $p(\beta|\tau) = \mathrm{N}(0, \tau)$, and the reference prior[4] to be 'the spike,' $p_0(\beta) = \delta(|\beta - 0|)$. The ECP for τ is then:

$$p(\tau; x) = \pi\left(\mathrm{KL}\left[\mathrm{N}(0, \sigma_y^2 + x^2\tau) \;||\; \mathrm{N}(0, \sigma_y^2)\right]\right) \times \left| \frac{\partial \; \mathrm{KL}\left[\mathrm{N}(0, \sigma_y^2 + x^2\tau) \;||\; \mathrm{N}(0, \sigma_y^2)\right]}{\partial \; \tau} \right| \quad (7)$$

where σ_y^2 is the response noise. In this case—and for conditional models in general—the ECP is a function of the features x and any other independent variables. Other default priors such as the *g-prior* (Zellner, 1986) and Jeffreys prior (Jeffreys, 1946) have this dependence as well, which reflects their holistic natures.

Choosing $\pi(\kappa)$ We next discuss the choice of $\pi(\kappa)$ and its effect on the resulting ECP. Figure 1(b) shows three ECPs, each defined by a different choice of KLD prior (1(a)): exponential (green), gamma (purple), and log-Cauchy (red). The choice of $\pi(\kappa)$ is significant. First considering the exponential prior, it clearly favors

[3] If $\pi(\kappa)$'s support is $\mathbb{R}^+$, then we assume a small constant is added to the KLD so that it never evaluates to zero.

[4] We use 'reference prior' to refer to the prior for the reference model, not to Bernardo (1979)'s class of objective priors.

$\tau > 0$ since the density function decays to zero at the origin. We can interpret this behavior in the context of the reference and original models as a strict preference for $p(\mathbf{y}|\tau)$. At the other extreme is the gamma prior: it has a mode at $\tau = 0$ and then quickly decays as τ increases. Thus, the gamma strictly prefers $p_0(\mathbf{y})$. Last we have the log-Cauchy, which we chose due to its heavy tail. Heavy-tailed priors have been well-validated for robust regression since they allow the shrinkage to be ignored under sufficient counter-evidence (Carvalho et al., 2009). A similar logic can be applied to the KLD: perhaps the reference model is too simplistic and $p(\mathbf{y}|\tau)$ is drastically superior. If so, we want any preference for the reference model to be forgotten. Figure 1(b) shows that the log-Cauchy results in an ECP with two modes, one at $\tau = 0$ and another at $\tau \approx .15$. The log-Cauchy is able to balance its preferences for $p(\mathbf{y}|\tau)$ and $p_0(\mathbf{y})$, interpolating between the exponential and gamma's single-mindedness.

Marginal Priors and Feature Dependence It is perhaps more intuitive to examine the marginal prior on β induced by the ECP: $p(\beta) = \int p(\beta|\tau)p(\tau)d\tau$. Figure 1(c) shows the marginal prior for the three ECPs considered above and compares them to the horseshoe prior (Carvalho et al., 2009) (black dashed line). The three priors behave as expected from looking at $p(\tau)$: the gamma shrinks the hardest and the log-Cauchy has the heaviest tails. Yet, recall that the ECP also depends on x. So far we have assumed $x = 1$, but in Figure 1(d) we show the same marginal priors for $x = 0.25$. This change in x results in drastically different ECPs. As $x \to 0$, the ECP (no matter the choice of $\pi(\kappa)$) becomes heavier tailed, allowing more deviation from the reference model. This behavior is natural since, when x is small, large β values are necessary to substantially change the model's predictions. See the appendix for more discussion, including the ECP for multivariate regression.

4 PREDICTIVE COMPLEXITY PRIORS

We now move on to our primary contribution: deriving a prior that has the same holistic, function-space properties as the ECP but is tractable for models such as NNs. As mentioned earlier, the primary weakness of the ECP is the difficulty of step #1: integrating over θ. In this section, we propose modifications to the ECP derivation that result in a tractable and scalable alternative. We call the resulting prior a *predictive complexity prior* (PredCP).

KLD Upper Bound As the primary obstacle is integrating over $p(\theta|\tau)$, we make headway by defining the PredCP using the following upper bound on the ECP's KLD:

$$\begin{aligned}\mathrm{KL}[p(\mathbf{y}|\tau)||p_0(\mathbf{y})] &= \mathrm{KL}\left[\mathbb{E}_{\theta|\tau}\left[p(\mathbf{y}|\theta)\right]||p_0(\mathbf{y})\right] \\ &\leq \mathbb{E}_{\theta|\tau}\mathrm{KL}\left[p(\mathbf{y}|\theta)||p_0(\mathbf{y})\right].\end{aligned} \tag{8}$$

We arrive at the upper bound via the strict convexity of $\mathrm{KL}\left[p(\mathbf{y}|\theta) \,||\, p_0(\mathbf{y})\right]$ and Jensen's inequality. The bound reverses the order in which marginalization and divergence computation are done for the ECP. This reversal makes the PredCP more practical since its KLD is taken between the data models. These are usually simple distributions (e.g. categorical, Gaussian) that afford a closed-form KLD. Unfortunately, the expectation over $\theta|\tau$ may still not be analytically available. We recommend evaluating the integral using a *differentiable, non-centered* Monte Carlo (MC) approximation (Kingma and Welling, 2014). Doing so ensures the KLD's derivative w.r.t. τ is well-defined. For supervised learning, a downside of the upper bound is that the dependencies between predictions are lost. Having the data model factorize across feature observations—$p(\mathbf{Y}|\boldsymbol{X}, \theta) = \prod_n p(\mathbf{y}|\boldsymbol{x}_n, \theta)$—results in the KLD becoming a point-wise sum. This is not an issue for the unsupervised case since there is no concept of features.

Mini-Batching For supervised learning, evaluating the PredCP requires a sum over all feature observations, which will be computationally costly for large data sets. Therefore we recommend the PredCP be evaluated with mini-batches. Moreover, we compute the KLD's mean across the mini-batch, not the sum. In doing so we assume that the batch's mean KLD represents an unbiased estimate of the full-data mean KLD. We use the mean KLD primarily for practical purposes: it is easier to set $\pi(\kappa)$'s parameters since they do not have to account for the batch size.

PredCP Final Form Below we give the final form of the PredCP for supervised learning, combining the point-mass reference prior, the KLD upper bound, and mini-batching:

$$\begin{aligned}p(\tau;\boldsymbol{X}_B) = &\left|\frac{1}{B}\sum_{b=1}^{B}\frac{\partial\ \mathbb{E}_{\theta|\tau}\mathrm{KL}_b}{\partial\ \tau}\right| \times \\ &\pi\left(\frac{1}{B}\sum_{b=1}^{B}\mathbb{E}_{\theta|\tau}\mathrm{KL}\left[p(\mathbf{y}|\boldsymbol{x}_b, \theta)\ ||\ p(\mathbf{y}|\boldsymbol{x}_b, \boldsymbol{\theta}_0)\right]\right)\end{aligned} \tag{9}$$

where b indexes the B-sized batch and $\mathbb{E}_{\theta|\tau}\mathrm{KL}_b$ is shorthand for the expected KLD for the bth instance. The PredCP encourages stronger shrinkage than the ECP, which is expected due to the upper bound. For a given τ, the PredCP deems the models to be more discrepant than the ECP would for the same τ. This is an appropriate inductive bias for the PredCP since it will be

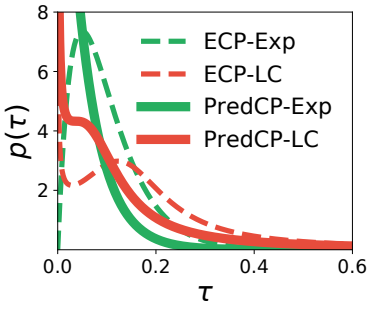


Figure 2: *PredCP for Linear Regression.* The ECP (dashed lines) vs the PredCP (solid lines) for the exponential and log-Cauchy KLD priors ($x = 1$).

used for large models that often require strong regularization. In Figure 2, we compare the ECP (dashed lines) and the corresponding PredCP (solid lines) for the linear regression example. The PredCP's inductive bias is evident in the leftward shift of the density functions. This shift can change the PredCP's behavior considerably in comparison to the corresponding ECP. The exponential's PredCP has a mode at $\tau = 0$ whereas its ECP decays to zero at the origin.

5 APPLICATIONS OF THE PREDCP

We now demonstrate the PredCP's utility for modern machine learning. We consider two applications: depth-selection for Bayesian NNs (Dikov and Bayer, 2019; Nalisnick et al., 2019; Antorán et al., 2020) and sharing statistical strength across tasks for meta-learning (Chen et al., 2019). Both of these applications exhibit the PredCP's ability to enable Bayesian reasoning across the model's macro-structures (e.g. layers) while still being a tractable and proper prior.

Bijectivity Conditions for Neural Networks Before moving on to these two applications, we first address some technical conditions. Recall that for the PredCP to be a proper prior (i.e. integrate to one), the expected KLD must be differentiable and bijective w.r.t. τ. The former is easy to satisfy by using a non-centered MC approximation, as mentioned above. It is not obvious if the latter is satisfied by NNs. One could check the condition via brute force, by using numerical integration. Since τ is a scalar, the numerical solution should be stable and makes for a good unit test. Yet, we show in the appendix that bijectivity is satisfied for feedforward NNs with ReLU activations and Gaussian or categorical observation models. No architectural modifications are necessary.

5.1 Depth Selection for Neural Networks

PredCPs allow us to perform Bayesian reasoning over the depth of a NN. First assume the NN to be a residual network (resnet) (He et al., 2016); later we will address the traditional feedforward case. Since we wish to isolate the effect of depth, we choose the reference model to have one fewer layer ($l-1$ layers) than the model of interest (l layers). The KLD between these models will then capture the extra capacity afforded by the additional layer.

More formally, for an arbitrary layer l, the prior on the (square) weight matrix $\mathbf{W}_l \in \mathbb{R}^{D_h \times D_h}$ for the reference and original models are: $p_0(\mathbf{W}_l) = \delta\left(|\mathbf{W}_l - \mathbf{0}|\right)$, $p(\mathbf{W}_l|\tau_l) = \mathrm{N}(\mathbf{0},\ \tau_l \mathbf{\Sigma}_l)$ where τ_l is again the parameter of interest. Integrating over p_0 sets $\mathbf{W}_k = \mathbf{0}$ for $k \geq l$ for the reference model and $k > l$ for the original model. The resnet then maps the hidden layers directly to the output layer, thereby allowing the PredCP to compare the predictions when using $l-1$ vs l layers. We can define the PredCP for all layers by applying the above priors recursively from the bottom-up:

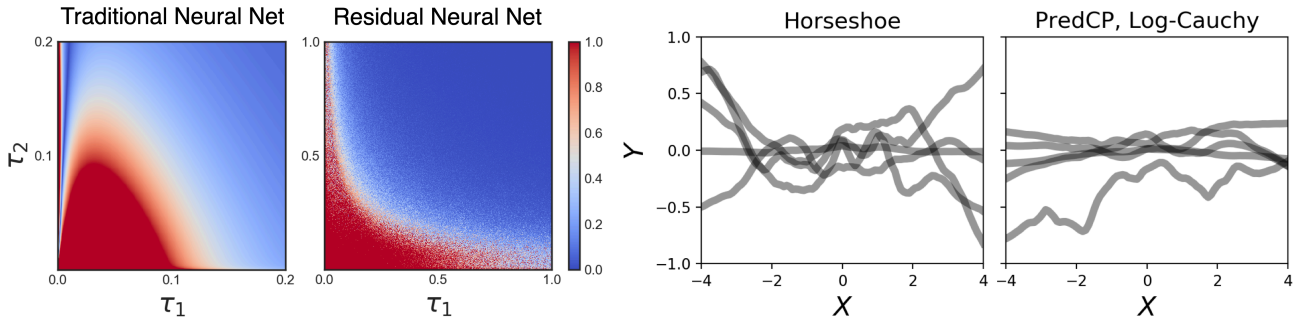
$$
\begin{aligned}
p(\tau_1, \ldots, \tau_L) &= p(\tau_1) \prod_{l=2}^{L} p(\tau_l | \tau_1, \ldots, \tau_{l-1}) \\
&= \prod_{l=1}^{L} \pi\left(\mathbb{D}(\tau_l; \tau_{1:l-1})\right) \left| \frac{\partial\ \mathbb{D}(\tau_l; \tau_{1:l-1})}{\partial\ \tau_l} \right|
\end{aligned}
\tag{10}
$$

where the divergence function is

$$
\begin{aligned}
&\mathbb{D}(\tau_l; \tau_{1:l-1}) = \\
&\mathbb{E}_{\{\mathbf{W}_j | \tau_j\}_{j=1}^{l}} \mathrm{KL}\left[p(\mathbf{y} | \{\mathbf{W}_j\}_{j=1}^{l}) \ ||\ p(\mathbf{y} | \{\mathbf{W}_k\}_{k=1}^{l-1}) \right].
\end{aligned}
$$

Computing the full prior requires L forward propagations, each evaluating a progressively deeper network with the hidden units at layer l serving as the last hidden layer. In practice, we cache the forward propagation required to evaluate the original model for τ_l and use it as the reference model when evaluating the prior for τ_{l+1}. Nearly the same procedure can be applied to non-residual networks, except that the residual connections can no longer be relied upon to transport the hidden units to the output layer. Rather, the network must be 'short circuited,' with the final hidden units being directly multiplied with the output weights.

Figure 3(a) shows the joint density function $\pi(\tau_1, \tau_2)$ for both traditional and residual NNs. The PredCP's capability for depth selection is conspicuous for the traditional NN (left). The high density region (red) touches the x-axis but not the y-axis except near the origin. This implies that τ_2 cannot grow unless $\tau_1 > 0$, meaning that the first layer is activated. For resnets (right), the density's L-shape means that either layer can be active while the other is inactive ($\tau \approx 0$), which



(a) Joint Density Function (b) Functions Sampled from Resnet

Figure 3: *Depth-Wise PredCP*. Subfigure (a) shows $\pi(\tau_1, \tau_2)$ for traditional (left) and residual (right) NNs. Subfigure (b) shows functions sampled from a resnet with a PredCP and a horseshoe prior for comparison. The KLD prior is Log-Cauchy$(0, 1)$ in both cases.

is made possible by the skip connection. Yet the density's bias towards the x-axis suggests that the resnet-PredCP still prefers to activate τ_1's layer before activating τ_2's.

Further intuition can be had by examining the depth-wise PredCP for resnets with a Gaussian data model. Denote a hidden layer for the nth observation as $\mathrm{h}_{n,l} = \mathrm{h}_{n,l-1} + f_l(\mathrm{h}_{n,l-1}\mathbf{W}_l)$ and the output weights as $\mathbf{W}_o \sim \mathrm{N}(\mathbf{0}, \mathbf{\Sigma})$. We assume $\mathbf{W}_l$ is parameterized as $\sqrt{\tau}_l \tilde{\mathbf{W}}_l$, $\tilde{\mathbf{W}}_l \sim \mathrm{N}(\mathbf{0}, \mathbf{\Sigma})$. The expected KLD for computing $\pi(\tau_l | \tau_{1:l-1})$ (Equation 10) is then:

$$\begin{aligned} &\mathbb{E}_{\{\mathbf{W}_j | \tau_j\}_{j=1}^{l}} \mathrm{KL}\left[p(\mathbf{y} | \{\mathbf{W}_j\}_{j=1}^{l}) || p(\mathbf{y} | \{\mathbf{W}_k\}_{k=1}^{l-1})\right] \\ &= \frac{\tau_l}{2\sigma_y^2} \frac{1}{N} \sum_{n=1}^{N} \mathrm{Var}_{\tilde{\mathbf{W}}, \mathbf{W}_o}\left[f_l(\mathrm{h}_{n,l-1}\tilde{\mathbf{W}}_l)\mathbf{W}_o\right] \end{aligned} \tag{11}$$

where σ_y^2 denotes the response noise and f_l is any positively homogeneous activation function (such as the ReLU). The crucial term $\mathrm{Var}[\mathbf{f}_l \mathbf{W}_o]$ represents the variance that the lth layer's transformation term contributes to the resnet's prediction for $\boldsymbol{x}_n$. The expression makes clear that the PredCP is performing functional regularization: a zero-favoring $\pi(\kappa)$ will encourage this variance to be small. Dropout has been shown to curb the variance of hidden units in a similar way (Baldi and Sadowski, 2013). Figure 3(b) shows functions sampled from a resnet with a horseshoe prior (left) and a depth-wise log-Cauchy PredCP (right). The PredCP's samples are closer to linear due to the regularization of the predictive variance. Yet, recall that the log-Cauchy is heavy-tailed and therefore allows some functions to stray from the origin, as we see one sample has done.

5.2 Hierarchical Modeling for Meta-Learning

Meta-learning is another natural application for the PredCP as it can control the degree to which information is pooled across tasks. Following the approach of Chen et al. (2019), we use the generative model: $\mathcal{D}_t \sim p(\mathcal{D}_t | \boldsymbol{\theta}_t)$, $\boldsymbol{\theta}_t \sim \mathrm{N}(\boldsymbol{\phi}, \tau\mathbb{I})$ where t indexes the task, $\mathcal{D}_t$ is data for the tth task, $\boldsymbol{\theta}_t$ are local parameters specific to the tth task, and $\{\boldsymbol{\phi}, \tau\}$ are global *meta-parameters*. The scale τ controls local adaptation, and as $\tau \to 0^+$, the task structure becomes irrelevant. This hierarchical meta-learning model is perfectly suited for a PredPC as the global parameters $\boldsymbol{\phi}$ define a natural reference model:

$$p_0(\mathcal{D}_t) = \int_{\boldsymbol{\theta}_t} p(\mathcal{D}_t | \boldsymbol{\theta}_t)\ \delta(|\boldsymbol{\theta}_t - \boldsymbol{\phi}|)\ d\boldsymbol{\theta}_t = p(\mathcal{D}_t | \boldsymbol{\phi}).$$

Computing the KLD between the original and reference models then quantifies the information lost when ignoring the task structure. The prior is written as:

$$\begin{aligned} p(\tau) \ = \ &\left| \frac{1}{T} \sum_t \frac{\partial\ \mathbb{E}_{\boldsymbol{\theta}_t | \tau} \mathrm{KL}_t}{\partial\ \tau} \right| \ \times \\ &\pi\left(\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\theta}_t | \tau} \mathrm{KL}\left[p_+(\mathcal{D}_t | \boldsymbol{\theta}_t)\ ||\ p_0(\mathcal{D}_t | \boldsymbol{\phi})\right] \right) \end{aligned} \tag{12}$$

where $\mathbb{E}_{\boldsymbol{\theta}_t | \tau} \mathrm{KL}_t$ is shorthand for the expected KLD on the tth task. In the experiments, we follow Chen et al. (2019)'s modular specification by applying the PredCP layer-wise. Doing so allows the feature-extracting shallow layers to adapt to a different degree than the classification-based final layer.

6 RELATED WORK

Penalized Complexity Prior Our work is directly inspired by and extends Simpson et al. (2017)'s *penalized complexity prior* (PCP). Let $p_0(\boldsymbol{\theta})$ denote a 'base' prior and $p_+(\boldsymbol{\theta} | \tau)$ an 'extended' prior, with τ controlling p_+'s expressivity. Simpson et al. (2017) define a prior for τ by placing a prior on the root-KLD and

Table 1: *Logistic Regression.* 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.

			Variational Inference			Markov Chain Monte Carlo		
Data Set	N_{train}	D	Half-Cauchy	**ECP**	**PredCP**	Half-Cauchy	**ECP**	**PredCP**
`allaml`	51	7129	$-0.43_{\pm.01}$	$\mathbf{-0.32}_{\pm.01}$	$\mathbf{-0.32}_{\pm.01}$	$-0.19_{\pm.02}$	$\mathbf{-0.17}_{\pm.02}$	$\mathbf{-0.17}_{\pm.02}$
`colon`	44	2000	$\mathbf{-0.61}_{\pm.02}$	$-0.63_{\pm.03}$	$-0.66_{\pm.02}$	$-0.54_{\pm.05}$	$\mathbf{-0.52}_{\pm.05}$	$-0.54_{\pm.04}$
`breast`	82	9	$-0.60_{\pm.01}$	$\mathbf{-0.58}_{\pm.01}$	$\mathbf{-0.58}_{\pm.01}$	$-0.55_{\pm.02}$	$-0.55_{\pm.01}$	$-0.55_{\pm.02}$

changing variables:

$$p(\tau) = \pi\left(\sqrt{2\mathrm{KL}\left[p_+(\boldsymbol{\theta}|\tau) \,||\, p_0(\boldsymbol{\theta})\right]}\right)\left|\frac{\partial\sqrt{2\mathrm{KL}}}{\partial\,\tau}\right|.$$

The major difference between the PCP and our PredCP is in how the KLD is formulated. Simpson et al. (2017) compute the divergence *between priors* $p(\boldsymbol{\theta})$ whereas we use the divergence *between data models* $p(\mathbf{y}|\boldsymbol{\theta})$. This crucial change is necessary since the PCP is hard to define for NNs and similarly complicated models. As the priors *and* the divergence are defined in $\boldsymbol{\theta}$-space, specifying the PCP still requires intimate knowledge of the parameters, running into the same challenges of high-dimensionality and unidentifiability. Because of our modification to $\mathbf{y}$-space, this makes the computation harder since we need to marginalize $\boldsymbol{\theta}$ to work with $p(\mathbf{y}|\tau)$. This difficulty necessitated the KLD upper bound in Equation 8. Another benefit of comparing the models in $\mathbf{y}$-space is that we can easily use point-mass priors for $\boldsymbol{\theta}$. Simpson et al. (2017) also do this, but because their KLD is defined on $\boldsymbol{\theta}$, they need to take limits. PCPs have been used as priors for P-splines (Ventrucci and Rue, 2016), distributional regression (Klein and Kneib, 2016), autoregressive processes (Sørbye and Rue, 2017), mixed effects models (Ventrucci et al., 2019), and Gaussian random fields (Fuglstad et al., 2019).

Functional Priors for Bayesian NNs Our work is also motivated by recent efforts to rethink prior specification for Bayesian NNs. As we truly care about the distribution over predictive functions, specifying *functional priors* has received much attention of late (Ma et al., 2019; Hafner et al., 2019; Flam-Shepherd et al., 2018; Louizos et al., 2019). The hope is that it is easier to reason about our preferences for functions than for parameters. However, existing functional priors introduce cumbersome byproducts into the Bayesian workflow. Placing a functional prior on a NN requires either taking infinite width limits (Pearce et al., 2019), optimizing divergences involving stochastic processes (Flam-Shepherd et al., 2017; Sun et al., 2019), or pretraining (Flam-Shepherd et al., 2017; Nalisnick and Smyth, 2018; Atanov et al., 2019). Our framework, on the other hand, uses reparameterization to obtain a proper prior on the parameters, creating no complications for traditional Bayesian inference.

7 EXPERIMENTS

We evaluate the PredCP on regression, classification, and few-shot learning tasks under a variety of algorithms for posterior inference. The experimental details are provided in the appendix. In all cases, we use relatively small data sets so that the prior's influence is not overwhelmed by the likelihood's.

Logistic Regression We first report an experiment in which the ECP can be computed and posteriors obtained with high-fidelity. We use the logistic regression model: $\mathrm{y} \sim \mathrm{Bernoulli}(f(\boldsymbol{x}\boldsymbol{\beta}))$, $\beta_d \sim \mathrm{N}(0, \lambda_d^2\tau^2)$, $\lambda_d \sim \mathrm{C}^+(0,1)$ where f denotes the logistic function and C^+ a half-Cauchy prior. We compare three priors for τ: $\mathrm{C}^+(0,1)$, which is the default prior recommended by Gelman (2006) and Carvalho et al. (2009), the ECP (via probit approximation (Bishop, 2006)), and the PredCP. The log-Cauchy$(0,1)$ is the KLD prior. We use `Stan` (Carpenter et al., 2017) to obtain the full posterior $p(\boldsymbol{\beta}, \boldsymbol{\lambda}, \tau|\boldsymbol{X}, \boldsymbol{y})$, performing both variational inference (Normal mean-field approximation) (Kucukelbir et al., 2017) and Hamiltonian MC. We test the priors on three small medical data sets (Golub et al., 1999; Alon et al., 1999; Patrício et al., 2018) so that the prior strongly influences the posterior. Furthermore, two of the data sets are high-dimensional (2000+) in order to test if the PredCP can prevent overfitting. Table 1 reports the predictive log-likelihood on the test set averaged over 20 splits. The ECP and PredCP have comparable performance and outperform the half-Cauchy in four of six cases and with one tie.

Neural Networks We next report results using resnets for regression: $\mathrm{y} \sim \mathrm{N}(\mathrm{y}|\boldsymbol{x}, \{\mathbf{W}_l\}_{l=1}^3)$, $\mathrm{w}_{l,i,j} \sim \mathrm{N}(0, \lambda_{l,i}^2\tau_l)$, $\lambda_{l,i} \sim \Gamma^{-1}(3,3)$ where l indexes layers, i rows of the weight matrix, and j columns. This prior has two forms of Bayesian regularization. The row-wise scale $\lambda_{l,i}$ implements *automatic relevance determination* (ARD) (MacKay, 1994; Neal, 1994), which controls the effective width. The layer-wise scale τ_l performs *automatic depth determination* (ADD) (Nalisnick et al.,

Table 2: *ARD-ADD Resnet.* We report test set RMSE for UCI benchmarks, comparing the PredCP against a shrinkage prior (Nalisnick et al., 2019) 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	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

Table 3: *Few-Shot Learning.* We report test set accuracy for the PredCP, comparing it to a shrinkage prior, a uniform prior (Chen et al., 2019), and non-Bayesian MAML.

	FEWSHOT-CIFAR100		MINI-IMAGENET	
	1-SHOT	5-SHOT	1-SHOT	5-SHOT
MAML	35.6 ± 1.8	50.3 ± 0.9	46.8 ± 1.9	58.4 ± 0.9
σ-MAML + uniform prior	39.3 ± 1.8	51.0 ± 1.0	47.7 ± 0.7	60.1 ± 0.8
σ-MAML + shrinkage prior	40.9 ± 1.9	52.7 ± 0.9	48.5 ± 1.9	60.9 ± 0.7
σ-MAML + **PredCP**	$\mathbf{41.2} \pm 1.8$	$\mathbf{52.9} \pm 0.9$	$\mathbf{49.3} \pm 1.8$	$\mathbf{61.9} \pm 0.9$

2019), as it controls the effective depth. We again compare three strategies for setting τ. The first is to use a fixed scale ($\tau = \tau_0$), thereby removing ADD. We performed light cross-validation for τ_0, reporting the better of $\tau_0 = \{0.1, 1.0\}$. The second is to use a shrinkage prior. Nalisnick et al. (2019) use a cross-validated inverse gamma prior, and we report their results as the strong baseline. For our method we use the PredCP with a log-Cauchy$(0, 1)$ KLD prior because it performed well for logistic regression. For posterior inference, we use Bayes-by-backprop (Blundell et al., 2015) for the weights and variational EM (Wu et al., 2019; Nalisnick et al., 2019) for the scales λ and τ. The maximization step cannot be performed analytically for the PredCP, as it can for the inverse gamma, and so we perform iterative gradient-based optimization. Again, we use relatively small data sets to ensure the prior remains influential: results on UCI benchmarks (Dua and Graff, 2019; Hernández-Lobato and Adams, 2015) are reported in Table 2. Using the PredCP results in the best test set root-mean-square error (RMSE) for three of the seven benchmarks (`boston`, `energy`, `yacht`) and in one tie (`kin8nm`).

Few-Shot Learning Our final experiment evaluates the PredCP for few-shot learning. We follow Chen et al. (2019)'s experimental framework, using the hierarchical model $\mathcal{D}_t \sim p(\mathcal{D}_t|\theta_t)$, $\theta_t \sim \mathrm{N}(\phi, \tau\mathbb{I})$ (described in Section 5) and their σ-MAML algorithm for optimization. In essence, σ-MAML performs MAP estimation for θ_t, ϕ, and τ. The classifier is the standard four-layer convolutional NN (Finn et al., 2017). We experimentally compare four different priors, each applied layer-wise (again following Chen et al. (2019)). The first baseline is $\theta_t \sim \mathbb{1}$ (improper uniform), which corresponds to standard MAML (Finn et al., 2017). The second baseline is Chen et al. (2019)'s model, which uses the improper uniform prior for the meta-parameters: $\phi_l, \tau_l \sim \mathbb{1}$. For a third baseline, we extend Chen et al. (2019)'s model by placing a shrinkage prior on τ_l, cross-validating over half-Cauchy, log-Cauchy, exponential, and gamma-exponential mixture distributions. Finally, our proposal is to place a PredCP on τ_l. We cross-validate over the same four shrinkage priors for $\pi(\kappa)$. We evaluated all models on the few-shot CIFAR100 (Oreshkin et al., 2018) and mini-ImageNet (Vinyals et al., 2016) classification benchmarks, using the standard 5-way 1-shot and 5-shot protocols. Table 3 reports the results. The PredCP consistently improves the mean accuracy across all experiments, albeit with some statistical overlap.

8 CONCLUSIONS

We proposed a novel prior termed the *predictive complexity prior* (PredCP). This prior is constructed procedurally and provides functional regularization. We found the PredCP to improve generalization across a range of small-data tasks that require careful regularization. The log-Cauchy$(0, 1)$ served as a good default KLD prior. We hope that this framework will inspire new directions in prior specification in machine learning, with an emphasis on building priors from reference models and exploiting compositional structure for adaptive regularization.

There are several directions for future work. One potential avenue is to improve the divergence computation.

For instance, it would be good to re-introduce predictive correlations into the divergence, as these were lost when switching to the upper bound (Equation 8). The resulting resnet-PredCP could then account for more interesting local structure in the predictive function, not just its point-wise variance. For another example, the Monte Carlo approximation could be stabilized, possibly with the use of control variates. Other divergence functions could also be explored, including symmetric ones such as the Hellinger distance. Scaling the PredCP to large neural networks is another open challenge, as the cost of our layer-wise PredCP increases with depth. Lastly, it would be interesting to investigate if the PredCP mitigates the misspecification issues described by Wenzel et al. (2020) or improves uncertainty estimation on out-of-distribution data.

Acknowledgements

This work was generously funded by Samsung Research (Samsung Electronics Co., Seoul, Republic of Korea). We thank John Bronskill for his help with the meta-learning experiments. The experiments were performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3), operated by the University of Cambridge Research Computing Service.

References

Uri Alon, Naama Barkai, Daniel A. Notterman, Kurt Gish, Suzanne Ybarra, Daniel Mack, and Arnold J. Levine. Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays. In *Proceedings of the National Academy of Sciences*, 1999.

Javier Antorán, James Urquhart Allingham, and José Miguel Hernández-Lobato. Variational Depth Search in ResNets. *ICLR Workshop on Neural Architecture Search*, 2020.

Andrei Atanov, Arsenii Ashukha, Kirill Struminsky, Dmitriy Vetrov, and Max Welling. The Deep Weight Prior. In *Proceedings of the International Conference on Learning Representations*, 2019.

Pierre Baldi and Peter J. Sadowski. Understanding Dropout. In *Advances in Neural Information Processing Systems*, 2013.

Jose M Bernardo. Reference Posterior Distributions for Bayesian Inference. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1979.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Networks. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Bob Carpenter, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A Probabilistic Programming Language. *Journal of Statistical Software*, 2017.

Carlos M. Carvalho, Nicholas G. Polson, and James G. Scott. Handling Sparsity via the Horseshoe. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, 2009.

George Casella. An Introduction to Empirical Bayes Data Analysis. *The American Statistician*, 1985.

Yutian Chen, Abram L. Friesen, Feryal Behbahani, David Budden, Matthew W. Hoffman, Arnaud Doucet, and Nando de Freitas. Modular Meta-Learning with Shrinkage. *NeurIPS Workshop on Meta-Learning*, 2019.

Georgi Dikov and Justin Bayer. Bayesian Learning of Neural Network Architectures. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, 2019.

Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Daniel Flam-Shepherd, James Requeima, and David Duvenaud. Mapping Gaussian Process Priors to Bayesian Neural Networks. *NeurIPS Workshop on Bayesian Deep Learning*, 2017.

Daniel Flam-Shepherd, James Requeima, and David Duvenaud. Characterizing and Warping the Function Space of Bayesian Neural Networks. *NeurIPS Workshop on Bayesian Deep Learning*, 2018.

Geir-Arne Fuglstad, Daniel Simpson, Finn Lindgren, and Håvard Rue. Constructing Priors that Penalize the Complexity of Gaussian Random Fields. *Journal of the American Statistical Association*, 2019.

Andrew Gelman. Prior Distributions for Variance Parameters in Hierarchical Models. *Bayesian Analysis*, 2006.

Andrew Gelman, Daniel Simpson, and Michael Betancourt. The Prior Can Often Only Be Understood in the Context of the Likelihood. *Entropy*, 2017.

Todd R. Golub, Donna K. Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P. Mesirov, Hilary Coller, Mignon L. Loh, James R. Downing, and Mark A. Caligiuri. Molecular Classification of Cancer: Class Discovery and Class Pre-

diction by Gene Expression Monitoring. *Science*, 1999.

Danijar Hafner, Dustin Tran, Timothy Lillicrap, Alex Irpan, and James Davidson. Noise Contrastive Priors for Functional Uncertainty. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Jonathan Heek and Nal Kalchbrenner. Bayesian Inference for Large Scale Image Classification. *ArXiv e-Prints*, 2019.

José Miguel Hernández-Lobato and Ryan Adams. Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Harold Jeffreys. An Invariant Form for the Prior Probability in Estimation Problems. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 1946.

Diederik Kingma and Max Welling. Efficient Gradient-Based Inference Through Transformations Between Bayes Nets and Neural Nets. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Nadja Klein and Thomas Kneib. Scale-Dependent Priors for Variance Parameters in Structured Additive Distributional Regression. *Bayesian Analysis*, 2016.

Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, and David Blei. Automatic Differentiation Variational Inference. *The Journal of Machine Learning Research*, 2017.

Christos Louizos, Xiahan Shi, Klamer Schutte, and Max Welling. The Functional Neural Process. In *Advances in Neural Information Processing Systems*, 2019.

Chao Ma, Yingzhen Li, and José Miguel Hernández-Lobato. Variational Implicit Processes. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

David MacKay. Bayesian Non-Linear Modeling for the Prediction Competition. *Maximum Entropy and Bayesian Methods*, 1994.

Eric Nalisnick and Padhraic Smyth. Learning Priors for Invariance. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, 2018.

Eric Nalisnick, José Miguel Hernández-Lobato, and Padhraic Smyth. Dropout as a Structured Shrinkage Prior. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Radford M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1994.

Boris Oreshkin, Pau Rodríguez López, and Alexandre Lacoste. TADAM: Task Dependent Adaptive Metric for Improved Few-Shot Learning. In *Advances in Neural Information Processing Systems*, 2018.

Miguel Patrício, José Pereira, Joana Crisóstomo, Paulo Matafome, Manuel Gomes, Raquel Seiça, and Francisco Caramelo. Using Resistin, Glucose, Age and BMI to Predict the Presence of Breast Cancer. *BMC Cancer*, 2018.

Tim Pearce, Russell Tsuchida, Mohamed Zaki, Alexandra Brintrup, and Andy Neely. Expressive Priors in Bayesian Neural Networks: Kernel Combinations and Periodic Functions. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*, 2019.

Juho Piironen and Aki Vehtari. Sparsity Information and Regularization in the Horseshoe and Other Shrinkage Priors. *Electronic Journal of Statistics*, 2017.

Christian Robert. *The Bayesian Choice*. Springer, 2001.

Daniel Simpson, Håvard Rue, Andrea Riebler, Thiago G. Martins, and Sigrunn H. Sørbye. Penalising Model Component Complexity: A Principled, Practical Approach to Constructing Priors. *Statistical Science*, 2017.

Sigrunn Holbek Sørbye and Håvard Rue. Penalised Complexity Priors for Stationary Autoregressive Processes. *Journal of Time Series Analysis*, 2017.

Shengyang Sun, Guodong Zhang, Jiaxin Shi, and Roger Grosse. Functional Variational Bayesian Neural Networks. In *Proceedings of the International Conference on Learning Representations*, 2019.

Massimo Ventrucci and Håvard Rue. Penalized Complexity Priors for Degrees of Freedom in Bayesian P-Splines. *Statistical Modelling*, 2016.

Massimo Ventrucci, Daniela Cocchi, Gemma Burgazzi, and Alex Laini. PC Priors for Residual Correlation Parameters in One-Factor Mixed Models. *Statistical Methods & Applications*, 2019.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, and Daan Wierstra. Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems*, 2016.

Florian Wenzel, Kevin Roth, Bastiaan S. Veeling, Jakub Świątkowski, Linh Tran, Stephan Mandt,

Jasper Snoek, Tim Salimans, Rodolphe Jenatton, and Sebastian Nowozin. How Good is the Bayes Posterior in Deep Neural Networks Really? In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Anqi Wu, Sebastian Nowozin, Edward Meeds, Richard E. Turner, José Miguel Hernández-Lobato, and Alexander L. Gaunt. Deterministic Variational Inference for Robust Bayesian Neural Networks. In *Proceedings of the International Conference on Learning Representations*, 2019.

A. Zellner. On Assessing Prior Distributions and Bayesian Regression Analysis with g-Prior Distributions. *Bayesian Inference and Decision Techniques*, 1986.

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical Stochastic Gradient MCMC for Bayesian Deep Learning. In *Proceedings of the International Conference on Learning Representations*, 2020.