Sparse Gaussian Process Regression Beyond Variational Inference

Martin Jankowiak  
Uber AI  
San Francisco, CA, USA  
jankowiak@uber.com

Geoff Pleiss  
Cornell University  
Dept. of Computer Science  
gp346@cornell.edu

Jacob R. Gardner  
Uber AI  
San Francisco, CA, USA  
jake.gardner@uber.com

Abstract

The combination of inducing point methods with stochastic variational inference has enabled approximate Gaussian Process (GP) inference on large datasets. Unfortunately, the resulting predictive distributions often exhibit substantially underestimated uncertainties. Worse still, in the regression case the predictive variance is typically dominated by observation noise, yielding uncertainty estimates that make little use of the input-dependent function uncertainty that makes GP priors attractive. In this work we propose a simple inference procedure that bypasses posterior approximations and instead directly targets the posterior predictive distribution. In an extensive empirical comparison with a number of alternative inference strategies on univariate and multivariate regression tasks, we find that the resulting predictive distributions exhibit significantly better calibrated uncertainties and higher log likelihoods—often by as much as half a nat or more per datapoint.

1 Introduction

Machine learning is finding increasing use in applications that make autonomous decisions based on predictive models. For example, machine learning can be used to drive dynamic load balancing in critical electrical systems, and autonomous vehicles use machine learning algorithms to detect and classify objects in unpredictable weather conditions and decide whether to brake. As machine learning models increasingly become deployed as components in larger decision making pipelines, it is essential that models be able to reason about uncertainty and risk. Techniques drawn from probabilistic machine learning offer the ability to deal with these challenges by offering predictive models with simple and interpretable probabilistic outputs.

Recent years have seen extensive use of variational inference (Jordan et al., 1999) as a workhorse inference algorithm for a variety of probabilistic models. The popularity of variational inference has been driven by a number of different factors, including: i) its amenability to data subsampling (Hoffman et al., 2013); ii) its applicability to black-box non-conjugate models (Kingma and Welling, 2013; Rezende et al., 2014); and iii) its suitability for GPU acceleration.

The many practical successes of variational inference come deployed as components in larger decision making pipelines, it is essential that models be able to reason about uncertainty and risk. Techniques drawn from probabilistic machine learning offer the ability to deal with these challenges by offering predictive models with simple and interpretable probabilistic outputs.
notwithstanding, it has long been recognized that variational inference often results in overconfident uncertainty estimates (see e.g. (Turner and Sahani, 2011)). This problem can be especially acute for Gaussian Process (GP) models (Bauer et al., 2016). In particular GP regressors fit with variational inference tend to apportion most of the predictive variance to the input-independent observation noise, making little use of the input-dependent function uncertainty that makes GP priors attractive in the first place. For a concrete demonstration of this phenomenon see the upper panel in Fig. 1. As we explain in Sec. 3.1, this tendency can be understood as resulting from the the asymmetry with which variational inference—through its reliance on Jensen’s inequality—treats the various contributions to the uncertainty in output space, in particular its asymmetric treatment of the observation noise.

In this work we propose a simple solution that corrects this undesirable behavior. By directly targeting the posterior predictive distribution—bypassing posterior approximations entirely—we formulate an objective function that treats the various contributions to predictive variance on an equal footing. The benefits of this approach are illustrated in the lower panel of Fig. 1, where the predictive uncertainty is much better at modelling heteroscedastic noise. The resulting predictive distributions, as we show empirically in Sec. 5, exhibit better calibrated uncertainties and higher log likelihoods than those obtained with variational inference.

2 Background

In Sec. 2.1-2.2 we review the basics of Gaussian Processes and inducing point methods. Then in Sec. 2.3 we review various approaches to GP inference that will serve as the baselines in our experiments. We also use this section to establish our notation.

2.1 Gaussian Process Regression

In probabilistic modeling Gaussian Processes offer powerful non-parametric function priors that are useful in a variety of regression and classification tasks (Rasmussen, 2003). For a given input space \( \mathbb{R}^d \) GPs are entirely specified by a covariance function or kernel \( k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R} \) and a mean function \( \mu : \mathbb{R}^d \to \mathbb{R} \). Different choices of \( \mu \) and \( k \) allow the modeler to encode prior information about the generative process. In the prototypical case of univariate regression the joint density takes the form

\[
p(y, f|X) = p(y|f, \sigma^2_{obs})p(f|X, \mu)
\]

where \( y \) are the real-valued targets, \( f \) are the latent function values, \( X = \{x_i\}_{i=1}^N \) are the \( N \) inputs with \( x_i \in \mathbb{R}^d \), and \( \sigma^2_{obs} \) is the variance of the Normal likelihood \( p(y|\cdot) \). The marginal likelihood takes the form

\[
p(y|X) = \int df \ p(y|f, \sigma^2_{obs})p(f|X)
\]

This marginal likelihood can be computed analytically, but doing so is computationally prohibitive for large datasets. This is because the cost scales as \( \mathcal{O}(X^3) \) from the terms involving \( K_{NN}^{-1} \) and \( \text{logdet} K_{NN} \) in Eqn. 2, where \( K_{NN} = k(X, X) \). This necessitates approximate methods when \( N \) is large.

2.2 Sparse Gaussian Processes

Over the past two decades significant progress has been made in scaling Gaussian Process inference to large datasets. The key technical innovation was the development of inducing point methods (Snelson and Ghahramani, 2006; Titsias, 2009; Hensman et al., 2013), which we now review. By introducing inducing variables \( u \) that are conditioned on variational parameters \( \{z_m\}_{m=1}^M \), where \( M = \dim(u) \ll N \) and with each \( z_m \in \mathbb{R}^d \), we augment the GP prior as follows

\[
p(f|X) \to p(f|u)p(u)
\]

where we have suppressed the dependence on \( X \) and \( Z \) in \( p(f|u, X, Z) \) and \( p(u|Z) \). We then appeal to Jensen’s inequality and lower bound the log joint density over the targets and inducing variables:

\[
\log p(y, u|X, Z) = \log \int df p(y|f)p(f|u)p(u)
\geq \mathbb{E}_{p(f|u)} [\log p(y|f) + \log p(u)]
= \sum_{i=1}^N \log \mathcal{N}(y_i | k_i^T K_{MM}^{-1} u, \sigma^2_{obs}) - \frac{1}{2 \sigma_{obs}^2} \text{Tr} K_{NN} + \log p(u)
\]

where \( k_i = k(x_i, Z) \), \( K_{MM} = k(Z, Z) \) and \( \tilde{K}_{NN} \) is given by

\[
\tilde{K}_{NN} = K_{NN} - K_{NM} K_{MM}^{-1} K_{MN}
\]

with \( K_{NM} = K_{MN}^T = k(X, Z) \). The essential characteristics of Eqn. 3 are that: i) it replaces expensive computations involving \( K_{NN} \) with cheaper computations like \( K_{MM}^{-1} \) that scale as \( \mathcal{O}(M^3) \); and ii) it is amenable to data subsampling, since the log likelihood and trace terms factorize as sums over datapoints \( (y_i, x_i) \).

2.3 Approximate GP Inference

We now describe how Eqn. 3 can be used to construct a variety of algorithms for scalable GP inference. We
limit ourselves to algorithms that satisfy two desiderata: i) support for data sub-sampling; and ii) the result of inference is a compact artifact that enables fast predictions at test time.

The rest of this section is organized as follows. In Sec. 2.3.1 we describe SVGP (Hensman et al., 2013)—currently the most popular method for scalable GP inference. In Sec. 2.3.2 we describe how Eqn. 3 can be leveraged in the context of MAP (maximum a posteriori) inference. Then in Sec. 2.3.3 we briefly review how ideas from robust variational inference can be applied to the GP setting.

### 2.3.1 SVGP

SVGP proceeds by introducing a multivariate Normal variational distribution $q(u) = \mathcal{N}(m, S)$. The parameters $m$ and $S$ are optimized using the ELBO (evidence lower bound), which is the expectation of Eqn. 3 w.r.t. $q(u)$ plus an entropy term $H(q(u))$:

$$L_{svgp} = \mathbb{E}_{q(u)}[\log p(y, u|X, Z)] - \text{KL}(q(u)||p(u))$$

$$= \sum_{i=1}^{N} \log \mathcal{N}(y_i k_i^T K_{MM}^{-1} m, \sigma^2_{obs}) - \frac{1}{2 \sigma_{obs}^2} \text{Tr} \tilde{K}_{NN}$$

$$- \frac{1}{2 \sigma_{obs}^2} \sum_{i=1}^{N} k_i^T K_{MM}^{-1} S K_{MM}^{-1} k_i - \text{KL}(q(u)||p(u))$$

where KL denotes the Kullback-Leibler divergence. Eqn. 5 can be rewritten more compactly as

$$L_{svgp} = \sum_{i=1}^{N} \left\{ \log \mathcal{N}(y_i k_i^T K_{MM}^{-1} m, \sigma^2_{obs}) - \frac{\sigma^2_f(x_i)^2}{2 \sigma^2_{obs}} \right\}$$

$$- \text{KL}(q(u)||p(u))$$

where $\sigma^2_f(x_i)^2 \equiv \text{Var}[f_i|x_i]$ denotes the latent function variance

$$\sigma^2_f(x_i)^2 = \tilde{K}_{ii} + k_i^T K_{MM}^{-1} S K_{MM}^{-1} k_i$$

$L_{svgp}$, which depends on $m, S, Z, \sigma^2_{obs}$ and the various kernel hyperparameters, can then be maximized with gradient methods. Below we refer to the resulting inference procedure as SVGP, with $\beta$-SVGP referring to the variant in which the KL divergence term in Eqn. 6 is scaled by a positive constant $\beta_{reg} > 0$. 

### 2.3.2 MAP

In contrast to SVGP, which maintains a distribution over the inducing variables $u$, MAP is a particle method in which we directly optimize a single point $u \in \mathbb{R}^M$ rather than a distribution over $u$. In particular we simply maximize Eqn. 3 evaluated at $u$. Note that the term $\log p(u)$ serves as a regularizer. In the following we refer to this inference procedure as MAP. To the best of our knowledge, it has not been considered before in the sparse GP literature.

### 2.3.3 Robust Gaussian Processes

Knoblauch et al. (2019); Knoblauch (2019) consider modifications to the typical variational objective (e.g. Eqn. 5), which consists of an expected log likelihood and a KL divergence term. In particular, they replace the expected log likelihood loss with an alternative divergence like the gamma divergence. This divergence raises the likelihood to a power

$$\log p(y|f) \rightarrow p(y|f)\gamma^{-1}$$

where typically $\gamma \in (1.0, 1.1)$. Empirically Knoblauch (2019) demonstrates that this modification can yield better performance than SVGP on regression tasks. We refer to this inference procedure as $\gamma$-Robust.

### 3 Regularized Predictive Distributions

We now describe the inference procedure that is the focus of this work. Crucially, unlike the inference strategies outlined in Sec. 2.3, our approach does not make use of the lower-bound energy surface in Eqn. 3.
To set the stage for our approach, we briefly review how SVGP arrives at a posterior predictive distribution. First, SVGP introduces a variational distribution \( q(f, u) \) and minimizes the KL divergence between \( q(f, u) \) and the posterior \( p(f, u | X, y) \). The posterior predictive can be formed using the marginal \( q(f) = \int du q(f, u) \) of the approximate posterior:

\[
p(y^* | x^*) = \int df^* df p(y^* | f^*) p(f^* | f, x^*) q(f)
\]  

(9)

Any deficiencies in the posterior approximation—e.g. overconfident uncertainty estimates in \( q(f) \)—will be inherited by the posterior predictive distribution \( p(y^* | x^*) \). A natural alternative is to directly target the posterior predictive distribution in Eqn. 9 from the outset.

In more detail, we introduce a parameterized family of distributions \( q(f) \), form the predictive distribution \( p(y^* | x^*) \) via Eqn. 9, and then choose a procedure to match \( p(y^* | x^*) \) to the empirical distribution \( p_{\text{data}}(y^* | x^*) \). There are several ways to realize this basic recipe. In the following we describe a simple variant that keeps the departure from SVGP to a minimum.

First, we use the same family of distributions \( q(f) \) that are used in SVGP. That is we introduce parameters \( \{m, S\} \) and form the distribution \( q(f) = \int du p(f | u) q(u) \) with \( q(u) = N(m, S) \). Second, we define an objective function based on the KL divergence between \( p(y^* | x^*) \) and \( p_{\text{data}}(y^* | x^*) \):

\[
L_{\text{pred}} = -\mathbb{E}_{p_{\text{data}}(y^* | x^*)} \text{KL}(p(y^* | x^*) || p(y^* | x^*))
\rightarrow \mathbb{E}_{p_{\text{data}}(y^* | x^*)} \left[ \log p(y^* | x^*) \right]
\]  

(10)

where \( p_{\text{data}}(y^* | x^*) \) is the empirical distribution over training data. In the second line we have dropped the entropy term \(-\mathbb{E}_{p_{\text{data}}(y^* | x^*)} \left[ \log p_{\text{data}}(y^* | x^*) \right]\), since it is a constant w.r.t. to the maximization problem. We obtain our final objective function by adding an optional regularization term modulated by a positive constant \( \beta_{\text{reg}} \geq 0 \):

\[
L_{\text{pred}}^{\beta_{\text{reg}}} = \mathbb{E}_{p_{\text{data}}(y^* | x^*)} \left[ \log p(y^* | x^*) \right] - \beta_{\text{reg}} \text{KL}(q(u) || p(u))
\]  

(11)

This objective can be expanded as

\[
L_{\text{pred}}^{\beta_{\text{reg}}} = \sum_{i=1}^{N} \log N(y_i | k_i^T K_{MM}^{-1} m, \sigma_{\text{obs}}^2 + \sigma_f(x_i)^2)
\rightarrow -\beta_{\text{reg}} \text{KL}(q(u) || p(u))
\]  

(12)

where \( \sigma_f(x_i)^2 = \text{Var}[f_i | x_i] \) is the latent function variance defined in Eqn. 7. The parameters \( m, S, Z \), as well as the observation noise \( \sigma_{\text{obs}} \) and kernel hyperparameters can then be optimized by maximizing Eqn. 11 using gradient methods (see Algorithm 1). In our experiments we refer to this approach—which employs a Normal distribution for \( q(u) \)—as \( \mathcal{N}\text{-Pred.} \).

When \( \beta_{\text{reg}} = 1 \) the form of the objective in Eqn. 11 can be motivated by a connection to Expectation Propagation (Minka, 2004); see Sec. D in the supplementary materials and (Li and Gal, 2017) for further discussion.

### 3.1 Discussion

What are the implications of directly fitting posterior predictive distributions? It is helpful to compare the objective function in Eqn. 11 to the SVGP objective in Eqn. 6. In both approaches the predictive variance \( \text{Var}[y^* | x^*] \) at an input \( x^* \) is given by the formula

\[
\text{Var}[y^* | x^*] = \sigma_{\text{obs}}^2 + \sigma_f(x^*)^2
\]  

(13)

where \( \sigma_f(x^*)^2 \) is the latent function variance at \( x^* \) (see Eqn. 7). In Eqn. 11 this leads to a data fit term

\[
L_{\text{pred}}^{\beta_{\text{reg}}} \rightarrow -\frac{1}{2} \frac{1}{\sigma_{\text{obs}}^2 + \sigma_f(x^*)^2} |y_* - \mu_f(x^*)|^2
\]  

(14)

where \( \mu_f(x^*) \) is the predictive mean function given by

\[
\mu_f(x^*) = k, K_{MM}^{-1} m
\]  

(15)

In contrast in SVGP the corresponding data fit term takes the form

\[
L_{\text{SVGP}} \rightarrow -\frac{1}{2} \frac{1}{\sigma_{\text{obs}}^2} |y_* - \mu_f(x^*)|^2
\]  

(16)

with \( \sigma_{\text{obs}}^2 \) in the denominator. Thus SVGP—and, more generally, variational inference for any regression model with a Normal likelihood—makes an arbitrary distinction between the observation variance \( \sigma_{\text{obs}}^2 \) and the latent function variance \( \sigma_f(x^*)^2 \), even though both terms contribute symmetrically to the total predictive variance in Eqn. 13. Moreover, this asymmetry will be inherited by any method that makes use of Eqn. 3, e.g. the MAP procedure described in Sec. 2.3.2.

If one is primarily interested in predictive performance—the typical case in machine learning—this asymmetric treatment of the two contributions to the predictive variance is troubling. As we will see in experiments (Sec. 5), the difference between Eqn. 14 and Eqn. 16 has dramatic consequences. In particular the data fit term in SVGP does nothing to encourage latent function variance \( \sigma_f(x^*) \). The consequence of this is that for many datasets \( \sigma_f(x^*) \ll \sigma_{\text{obs}} \); i.e. most of the predictive variance is explained by the input-independent observation noise.

---

5Arbitrary from the point of view of output space. Depending on the particular application and structure of the model, distinctions between different contributions to the predictive variance may be of interest.
In contrast, the data fit term in Eqn. 14 directly encourages large $\sigma_f(x^*)$, typically resulting in behavior opposite to that of SVGP, i.e. $\sigma_f(x^*) \gg \sigma_{\text{obs}}$. This is gratifying because—after having gone to the effort to introduce an input-dependent kernel and learn an appropriate geometry on the input space—we end up with predictive variances that actually make use of the input-dependent kernel.

3.2 Variants

A number of variants to the simple approach outlined above immediately suggest themselves. One possibility is to take the formal limit $\mathbf{S} \to 0$ in $q(\mathbf{u})$. In this limit $q(\mathbf{u})$ is a Dirac delta distribution, the predictive latent function variance $\sigma_f^2(x^*) \to K_{\star\star}$, and the number of parameters is now linear in $M$ instead of quadratic.\(^6\) Below we refer to this variant as $\delta$-Pred. Another possibility is to restrict the covariance matrix $\mathbf{S}$ in $q(\mathbf{u})$ to be diagonal; below we refer to this ‘mean field’ variant as $\mathcal{N}$-Pred-MF.

A number of other variants are also possible. For example we might replace the regularization term $\text{KL}(q(\mathbf{u})|p(\mathbf{u}))$ in Eqn. 11 with another divergence, for example the forward KL divergence $\text{KL}(p(\mathbf{u})|q(\mathbf{u}))$ or a Rényi divergence (Li and Turner, 2016). Alternatively we could use another divergence measure in Eqn. 10—e.g. the gamma divergence used in Sec. 2.3.3—to control the qualitative features of $p_{\text{data}}(y^|x^*)$ that we would like to capture in $p(y^|x^*)$. Another possibility is to use larger families of $q(\mathbf{u})$ distributions, e.g. a mixture of Dirac delta or Normal distributions. We leave the exploration of these and other variants to future work.

4 Related Work

The use of pseudo-inputs and inducing point methods to scale-up Gaussian Process inference has spawned a large literature, especially in the context of variational inference (Csató and Opper, 2002; Seeger et al., 2003; Quiñonero-Candela and Rasmussen, 2005; Snelson and Ghahramani, 2006; Titsias, 2009; Hensman et al., 2013, 2015a; Cheng and Boots, 2017). While variational inference remains the most popular inference algorithm in the scalable GP setting, researchers have also explored different variants of Expectation Propagation (Hernández-Lobato and Hernández-Lobato, 2016; Bui et al., 2017) as well as Stochastic gradient Hamiltonian Monte Carlo (Havasi et al., 2018) and other MCMC algorithms (Hensman et al., 2015b). For a recent review of scalable methods for GP inference we refer the reader to (Liu et al., 2018).

Our focus on the predictive distribution recalls (Snelson and Ghahramani, 2005), in which the authors construct parsimonious approximations to Bayesian predictive distributions. Their approach differs from the approach adopted here, since the posterior distribution is still computed (or approximated) as an intermediate step, whereas we completely bypass the posterior.

5 Experiments

In this section we compare the empirical performance of the Predictive approach described in Sec. 3 ($\mathcal{N}$-Pred, $\mathcal{N}$-Pred-MF, and $\delta$-Pred) to the baseline inference strategies described in Sec. 2.3. All our models use a prior mean of zero and a Matérn kernel with independent length scales for each input dimension.

5.1 Univariate regression

We consider a mix of robotics datasets in a multivariate regression task, with the input dimension in the range $\text{dim}(x) \in [10, 23]$ and $\text{dim}(\mathbf{x})\in[3, 380]$. For all but the two largest datasets we also compare to Exact GP inference, leveraging the conjugate gradient approach described in (Wang et al., 2019).

We summarize the results in Fig. 2-4. All three Predictive methods yield consistently lower negative log likelihoods (NLLs) than the scalable inference baselines. In general it is the case that $\mathcal{N}$-Pred outperforms $\mathcal{N}$-Pred-MF which outperforms $\delta$-Pred. Indeed on most datasets the Predictive methods outperform Exact GP inference. We hypothesize that this is at least partially due to the ability of inducing point methods to model heteroscedasticity, a “bonus” feature that was already noted in (Snelson and Ghahramani, 2006). Perhaps surprisingly, we also note that on most datasets MAP yields comparable log likelihoods to SVGP.

The results for root mean squared errors (RMSE) exhibit less divergence (see Fig. 3). Exact GP inference yields the lowest RMSEs for most datasets, with $\beta$-SVGP the best among the approximate inference methods. While the Predictive methods are competitive on RMSE, they perform somewhat less well on average than the scalable inference baselines. The best Predictive RMSE (usually $\delta$-Pred) is within $3.4\%$ (relative) of the best baseline method RMSE (usually $\beta$-SVGP) for 11 of the 12 datasets. We note that even

\(^6\)Additionally in the regularizer we make the replacement $-\beta_{\text{reg}} \text{KL}(q(\mathbf{u})|p(\mathbf{u})) \to \beta_{\text{reg}} \log p(\mathbf{u})$. 

when the Predictive RMSE is somewhat high, the corresponding NLL is still substantially lower than the scalable inference baselines. We hypothesize that this can be understood from the form of the Predictive objective function in Eqn. 11, which explicitly targets predictive log likelihoods. In particular Predictive GP models prefer large predictive uncertainty in regions of input space where good data fit is hard to achieve. Consequentially, there is less incentive to move the predictive mean function away from the prior in these re-
gions, which can then result in higher RMSEs. Among the three Predictive methods, we note that $N$-Pred-MF strikes a good balance, yielding low NLLs without sacrificing much on RMSE.

Strikingly, the Predictive GPs yield predictive variances that are in a qualitatively different regime than those resulting from the scalable inference baselines. Fig. 4 depicts the fraction of the overall predictive variance $\text{Var}[y \mid x]$ due to the observational noise $\sigma^2_{\text{obs}}$. The predictive variances from the baseline methods make comparatively little use of input-dependent function uncertainty, instead relying primarily on the observation noise. By contrast the variances of Predictive GPs are dominated by latent function uncertainty. This substantiates the discussion in Sec. 3.1. Additionally, this observation explains the similar log likelihoods exhibited by SVG and MAP; since neither method makes much use of latent function uncertainty, the uncertainty encoded in $q(u)$ is of secondary importance.\footnote{Note that MAP can be viewed as a degenerate limit of SVG in which $q(u)$ is a Dirac delta function.}

Finally we note that for most datasets the Predictive GPs prefer small values of $\beta_{\text{reg}}$. This suggests that choosing $M \ll N$ is largely sufficient for ensuring well-regularized models and that overfitting is not much of a concern in practice.

### 5.2 Multivariate regression

We consider a mix of robotics datasets in a multivariate regression task, with the input dimension in the range $\text{dim}(x) \in [10, 23]$ and the output dimension in the range $\text{dim}(y_{1:D}) \in [7, 18]$. Our multivariate regressor is a simple generalization of Eqn. 1, with independent GPs for each output dimension; see the supplementary materials for details. Each dataset has $N \sim 10^4$ to $N \sim 10^5$ datapoints, with three of the datasets collected from real-world robots and the rest generated using the MuJoCo physics simulator (Todorov et al., 2012). We exclude MAP from our comparison, since it yielded the worst log likelihoods on the univariate regression tasks in the previous section.

We summarize the results in Fig. 5-6. The performance of the Predictive GPs on these datasets largely mirrors the results of the univariate experiments. That is, the Predictive GPs yield consistently lower NLLs than the scalable inference baselines, with mean root mean squared errors (MRMSEs) that are generally worse, especially for $N$-Pred. Strikingly, on 6 out of 10 datasets the best Predictive method yields a NLL that is at least 0.5 nats lower than the scalable baselines.

See Table 1 for a summary of the results for both the univariate and multivariate regression tasks. For additional results on both sets of experiments see the Supplementary Materials.

### 5.3 Calibration in DKL Regression

In this section, we demonstrate that the Predictive approach offers an effective mechanism for calibrating deep neural networks for regression. To evaluate the potential of this approach, we utilize a real-world dataset of vehicular trip durations in two large cities. In this setting, uncertainty estimation is critical for managing risk when estimating transportation costs.

We compare three methods: i) deep kernel learning (Wilson et al., 2016) using SVG (SVG+DKL); ii) deep kernel learning using regularized predictive distributions ($N$-Pred+DKL; this paper); and iii) MCDropout (Gal and Ghahramani, 2016), a popular method for calibrating neural networks that does not rely on Gaussian Processes.

In Fig. 7 we visualize how well each of the three methods is calibrated as compared to the best possible calibration for a model with Normal predictive distributions. Overall $N$-Pred+DKL performs the best, with MCDropout outperforming SVG+DKL. Using regularized predictive distributions has a number of additional advantages over MCDropout. In particular, because the predictive variances can be computed analytically for Gaussian Process models,
Table 1: Average ranking of methods (lower is better). NCRPS is the (negative) Continuous Ranking Probability Score, a popular calibration metric for regression (Gneiting and Raftery, 2007). Rankings are averages across datasets and splits. See Sec. 5.1-5.2 for details.

| Datasets  | Metric | MAP | SVGP | β-SVGP | γ-Robust | δ-Pred | N-Pred-MF | N-Pred |
|-----------|--------|-----|------|--------|----------|--------|-----------|--------|
| Univariate| NLL    | 6.47| 5.50 | 4.54   | 5.48     | 3.00   | 1.98      | 1.02   |
|           | RMSE   | 4.15| 2.67 | 2.80   | 4.17     | 3.98   | 4.92      | 6.33   |
|           | NCRPS  | 6.53| 5.44 | 4.35   | 4.65     | 3.18   | 1.99      | 1.86   |
| Multivariate| NLL     | —   | 5.66 | 4.71   | 4.47     | 3.02   | 1.77      | 1.23   |
|           | MRMSE  | 2.04| 1.26 | 3.29   | 4.01     | 4.28   | 5.84      |        |

Figure 7: We visualize the calibration of three probabilistic deep learning models fit to trip data in two cities. For each method we depict the empirical CDF of the z-scores $z = (y^* - \mu_f(x^*)) / \sigma(x^*)$ computed on the test set $\{ (x^*_k, y^*_k) \}$. The “Ideal CDF” is the Normal CDF, which corresponds to the best possible calibration for a model with a Normal predictive distribution. See Sec. 5.3 for details.

the $N$-Pred+DKL model is significantly faster at test time than MCDropout, which requires forwarding data points through many sampled models (here 50). This is impractically slow for many applied settings, especially for large neural networks.

6 Discussion

Gaussian Process regression with a Normal likelihood represents a peculiar case in that: i) we can give an analytic formula for the exact posterior; but ii) it is impractical to compute for large datasets. In this work we have argued that if our goal is high quality predictive distributions, it is sensible to bypass posterior approximations and directly target the quantity of interest. While this may be a bad strategy for an arbitrary Bayesian model, in the case of GP regression inducing point methods provide a natural family of parametric predictive distributions whose capacity can be controlled to prevent overfitting. As we have shown empirically, the resulting predictive distributions exhibit significantly better calibrated uncertainties and higher log likelihoods than those obtained with variational inference, which tend to yield overconfident uncertainty estimates that make little use of the kernel.

An interesting question for future work is whether this approach can be applied to other likelihoods, especially those used in classification. On the face of it the classification setting seems challenging, since closed form expressions for predictive distributions are no longer available. One interesting strategy could be to define adaptive quadrature rules that yield high precision approximations of the pointwise predictive density.

References

Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding probabilistic sparse gaussian process approximations. In *Advances in neural information processing systems*, pages 1533–
Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1):973–978, 2019.

Thang D Bui, Josiah Yan, and Richard E Turner. A unifying framework for gaussian process pseudo-point approximations using power expectation propagation. *The Journal of Machine Learning Research*, 18(1):3649–3720, 2017.

Ching-An Cheng and Byron Boots. Variational inference for gaussian process models with linear complexity. In *Advances in Neural Information Processing Systems*, pages 5184–5194, 2017.

Andrzej Cichocki and Shun-ichi Amari. Families of alpha-beta-and gamma-divergences: Flexible and robust measures of similarities. *Entropy*, 12(6):1532–1568, 2010.

Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. *Neural computation*, 14(3):641–668, 2002.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

James Hensman, Alexander G Matthews, Maurizio Filippone, and Zoubin Ghahramani. Mcmc for variationally sparse gaussian processes. In *Advances in Neural Information Processing Systems*, pages 1648–1656, 2015b.

Daniel Hernández-Lobato and José Miguel Hernández-Lobato. Scalable gaussian process classification via expectation propagation. In *Artificial Intelligence and Statistics*, pages 168–176, 2016.

José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Daniel Hernández-Lobato, Thang Bui, and Richard Turner. Black-box α-divergence minimization. 2016.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Jeremias Knoblauch. Robust deep gaussian processes. *arXiv preprint arXiv:1904.02303*, 2019.

Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. Generalized variational inference. *arXiv preprint arXiv:1904.02063*, 2019.

Yingzhen Li and Yarin Gal. Dropout inference in bayesian neural networks with alpha-divergences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2052–2061. JMLR. org, 2017.

Yingzhen Li and Richard E Turner. Rényi divergence variational inference. In *Advances in Neural Information Processing Systems*, pages 1073–1081, 2016.

Yingzhen Li, José Miguel Hernández-Lobato, and Richard E Turner. Stochastic expectation propagation. In *Advances in neural information processing systems*, pages 2323–2331, 2015.

Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. When gaussian process meets big data: A review of scalable gps. *arXiv preprint arXiv:1807.01065*, 2018.

Alexander Graeme de Garis Matthews. *Scalable Gaussian process inference using variational methods*. PhD thesis, University of Cambridge, 2017.
Franziska Meier, Philipp Hennig, and Stefan Schaal. Incremental local gaussian regression. In *Advances in Neural Information Processing Systems*, pages 972–980, 2014.

Thomas Minka. Power ep. *Dep. Statistics, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep.*, 2004.

Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Hugh Salimbeni, Vincent Dutordoir, James Hensman, and Marc Peter Deisenroth. Deep gaussian processes with importance-weighted variational inference. *arXiv preprint arXiv:1905.05435*, 2019.

Matthias Seeger, Christopher Williams, and Neil Lawrence. Fast forward selection to speed up sparse gaussian process regression. Technical report, 2003.

Edward Snelson and Zoubin Ghahramani. Compact approximations to bayesian predictive distributions. In *Proceedings of the 22nd international conference on Machine learning*, pages 840–847. ACM, 2005.

Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006.

Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

R. E. Turner and M. Sahani. Two problems with variational expectation maximisation for time-series models. In D. Barber, T. Cemgil, and S. Chiappa, editors, *Bayesian Time series models*, chapter 5, pages 109–130. Cambridge University Press, 2011.

Sethu Vijayakumar and Stefan Schaal. Locally weighted projection regression: An o (n) algorithm for incremental real time learning in high dimensional space. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, volume 1, pages 288–293, 2000.

Ke Alexander Wang, Geoff Pleiss, Jacob R Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson. Exact gaussian processes on a million data points. *arXiv preprint arXiv:1903.08114*, 2019.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378, 2016.
A Experimental Details

We use zero mean functions and Matérn kernels with independent length scales for each input dimension throughout. All models and experiments are implemented using the GPyTorch framework (Gardner et al., 2018) and the Pyro probabilistic programming language (Bingham et al., 2019).

A.1 Heteroscedastic dataset

The training dataset used to generate Fig. 1 has \( N = 100 \) datapoints. The top panel depicts a regressor fit with SVGP, and the lower panel depicts a regressor fit with \( \mathcal{N}\text{-Pred} \). We used \( M = 25 \) inducing points and did 400 epochs of training with a mini-batch size of \( B = 20 \). For both methods we used \( \beta_{\text{reg}} = 1.0 \). Other details are as in Sec. A.2.

A.2 Univariate regression

We use the Adam optimizer for optimization with an initial learning rate of \( \ell = 0.01 \) that is progressively decimated over the course of training (Kingma and Ba, 2014). We use a mini-batch size of \( B = 10^4 \) for the Buzz, Song, 3Droad and Houselectric datasets and \( B = 10^3 \) for all other datasets. We train for 400 epochs except for the Houselectric dataset where we train for 200 epochs. Except for the Exact results, we do 10 train/test/validation splits on all datasets (always in the proportion 15:3:2, respectively). In particular for the Exact results we do 3 train/test/validation splits on the smaller datasets and one split for the two largest (3Droad and Song). All datasets are standardized in both input and output space; thus a predictive distribution concentrated at zero vector yields a root mean squared error of unity along each dimension. We use \( M = 500 \) inducing points initialized with k-means. We use the validation set to determine a small set of hyperparameters. In particular for \( \beta\text{-SVGP} \) we search over \( \beta_{\text{reg}} \in \{0.1, 0.3, 0.5, 1.0\} \). For \( \gamma\text{-Robust} \) we search over \( \{0.1, 0.3, 1.0, 1.07\} \). For both \( \beta\text{-Pred} \) and \( \delta\text{-Pred} \) we search over \( \beta_{\text{reg}} \in \{0.01, 0.05, 0.2, 1.0\} \).

A.3 Multivariate regression

We consider a model specified by its joint density

\[
p(y_{1:D}, f_{1:D} | X) = \prod_{i=1}^{D} p(y_i | f_i, \sigma_{\text{obs},i}^2) p(f_i | X, k_i)
\]

where \( D \) is the output dimension and each \( f_i \) is an independent GP with its own kernel \( k_i \), its own inducing locations \( Z_i \), and its own parameters \( m_i \) and \( S_i \). The likelihood factorizes along the \( D \) dimensions, and each dimension is endowed with its own observation noise \( \sigma_{\text{obs},i}^2 \).

Many of the robotics datasets we used are available online. Note that some of these datasets have been used in a number of papers, including references (Vijayakumar and Schaal, 2000; Meier et al., 2014; Cheng and Boots, 2017; Salimbeni et al., 2019).

We use the Adam optimizer for optimization with an initial learning rate of \( \ell = 0.01 \) that is progressively decimated over the course of training (Kingma and Ba, 2014). We use a mini-batch size of \( B = 1000 \) for the three datasets collected from real-world robots (Sarcos, Kuka, Baxter) and \( B = 2000 \) for all other datasets. We train for 400 epochs for all datasets and do 10 train/test/validation splits on all datasets (always in the proportion 15:3:2, respectively). All datasets are standardized in both input and output space; thus a predictive distribution concentrated at zero vector yields a root mean squared error of unity along each dimension. We use \( M = 500 \) inducing points initialized with k-means. We use the validation set to determine a small set of hyperparameters. In particular for \( \beta\text{-SVGP} \) we search over \( \beta_{\text{reg}} \in \{0.1, 0.3, 0.5, 1.0\} \). For \( \gamma\text{-Robust} \) we search over \( \{0.1, 0.3, 1.0, 1.07\} \). For both \( \beta\text{-Pred} \) and \( \delta\text{-Pred} \) we search over \( \beta_{\text{reg}} \in \{0.01, 0.05, 0.2, 1.0\} \).

A.4 DKL calibration

MC-Dropout has two hyperparameters that must be set by hand: a dropout proportion \( p \) and a prior variance inflation term \( \tau \). These were set by temporarily removing a portion of the training data as a validation set and performing a small grid search on each dataset over \( p \in [0.05, 0.1, 0.15, 0.2, 0.25] \) and \( \tau \in [0.05, 0.1, 0.25, 0.5, 0.75, 1.0] \). For the Predictive model, \( \beta \in [0.05, 0.2, 0.5, 1.0] \) was chosen in a similar fashion. The datasets for each of the two cities contain 30 features that encode various aspects of a trip like origin and destination location, time of day and week, as well as various rudimentary routing features.

For all three methods, we use the same five layer fully connected neural network, with hidden representation sizes of \([256, 256, 128, 128, 64]\) and ReLU nonlinearities. We use the Adam optimizer and use an initial learning rate of \( \ell = 0.01 \), which we drop by a factor of 0.1 at 100 and 150 epochs. We train for 200 epochs for all three methods. For the GP methods, we use \( M = 1024 \) inducing points, initialized by randomly selecting training data points and passing them through the initial feature extractor.  

See http://www.gaussianprocess.org/gpm1/data/  
http://github.com/hughesalimbeni/bayesian_benchmarks
Figure 8: We depict (negative) test continuous ranked probability scores (NCRPS) for twelve univariate regression datasets (lower is better). Results are averaged over ten random train/test/validation splits. See Sec. 5.1 for details.

Figure 9: We depict the mean fraction of the predictive variance that is due to the observation noise for ten multivariate regression datasets (as measured on the test set and averaged over output dimensions). Results are averaged over ten random train/test/validation splits. See Sec. 5.2 for details.

B Additional Experimental Results

In Fig. 8 we plot mean (negative) Continuous Ranked Probability Scores (Gneiting and Raftery, 2007) for the 12 univariate regression datasets. We see that the Predictive models achieve the lowest score on all datasets. Fig. 9 displays what fraction of the multi-output models’ variance comes from the likelihoods’ observation noise. Similarly to the univariate results, the SVGP variants make little use of the latent function. Conversely, the Predictive models rely less on the observation noise for predictive uncertainty.

C Whitened Sparse Gaussian Process Regression

The hyperparameters and variational parameters of the models can be learned by directly optimizing the objective functions in Eqn. 3 (for MAP), in Eqn. 5 (for SVGP), and Eqn. 11 (for the Predictive methods). In practice, we modify these optimization objectives using a transformation proposed by (Matthews, 2017). The “whitening transformation” is a simple change of variables:

$$u' = \Lambda_{MM}^{-1} u$$

where $$\Lambda_{ZZ}$$ is a matrix such that $$\Lambda_{ZZ} \Lambda_{ZZ}^{\top} = K_{ZZ}$$. (Typically, $$\Lambda_{ZZ}$$ is taken to be the Cholesky factor of $$K_{ZZ}$$.) Intuitively, this transformation is advantageous because it reduces the number of changing terms in the objective functions. In whitened coordinates the prior $$p(u')$$ is constant: $$p(u') = \mathcal{N}(0, \Lambda_{ZZ}^{-1} K_{ZZ} \Lambda_{ZZ}^{\top}) = \mathcal{N}(0, I)$$. Incorporating this transformation into Eqn. 3 gives us

$$\log p(y, u' | X, Z) \geq \mathbb{E}_{p(f|w)} [ \log p(y|f) + \log p(u') ] = \sum_{i=1}^{n} \log \mathcal{N}(y_i | k_i^T \Lambda_{ZZ}^{-1} u', \beta^{-1}) - \frac{1}{2} \text{Tr} \tilde{K}_{NN} - \frac{1}{2} \| u' \|_2^2 - \frac{M}{2} \log(2\pi).$$

Importantly, the prior term $$p(u')$$ in the modified objective does not depend on the inducing point locations $$Z$$ or kernel hyperparameters. In all experiments we use similarly modified objectives for better optimization. For MAP and $$\delta$$-Pred, we directly optimize the whitened variables $$u'$$. For $$\mathcal{N}$$-Pred, SVGP, and its variants, the whitened variational distribution is given as $$q(u') = \mathcal{N}(\Lambda_{ZZ}^{-1} m, \Lambda_{ZZ}^{-1} S \Lambda_{ZZ}^{-1})$$. We parameterize the mean with a vector $$m' = \Lambda_{ZZ}^{-1} m$$ and we parameterize the covariance with a lower triangular matrix $$LL^\top = \Lambda_{ZZ}^{-1} S \Lambda_{ZZ}^{-1}$$.
D  Connection to stochastic EP and the BB-α objective

Suppose we want to approximate the distribution

\[ p(\omega) = \frac{1}{Z} p_0(\omega) \prod_{i=1}^{N} f_n(\omega) \]  \hspace{1cm} (18)

where \( Z \) is an unknown normalizer. In the prototypical context of Bayesian modeling \( p_0(\omega) \) would be a prior distribution and \( f_n(\omega) \) would be a likelihood factor for the \( n \)th datapoint. Expectation propagation (EP) is a broad class of algorithms that can be used to approximate distributions like that in Eqn. 18 (Minka, 2004). In the following we give a brief review of a few variants of EP and describe a connection to the Predictive objective defined in Eqn. 11 in the main text.

Li et al. (2015) propose a particular variant of EP called Stochastic EP that reduces memory requirements by a factor of \( N \) by tying (i.e. sharing) factors together. In subsequent work Hernández-Lobato et al. (2016) present a version of Stochastic EP that is formulated in terms of an energy function, the so-called BB-α objective \( L_\alpha \), which is given by

\[ L_\alpha = -\frac{1}{\alpha} \sum_{i=1}^{N} \log E_{q(\omega)} \left[ \left( \frac{f_n(\omega)p_0(\omega)^{1/N}}{q(\omega)^{1/N}} \right)^{\alpha} \right] \]  \hspace{1cm} (19)

where \( q(\omega) \) is a so-called cavity distribution and \( \alpha \in [0,1] \). As shown in (Li and Gal, 2017), this can be rewritten as

\[ L_\alpha = R_\xi(\tilde{q}||p_0) - \frac{1}{\alpha} \sum_{i=1}^{N} \log E_q [f_n(\omega)^\alpha] \]  \hspace{1cm} (20)

where \( \tilde{q}(\omega) \) is defined by the equation

\[ q(\omega) = \frac{1}{Z_q} \tilde{q}(\omega) \left( \frac{\tilde{q}(\omega)}{p_0(\omega)} \right)^{\frac{\xi}{1-\xi}} \]  \hspace{1cm} (21)

and where \( \xi \equiv \frac{N}{N-\alpha} \) and \( R_\xi(\tilde{q}||p_0) \) is a Rényi divergence. Li and Gal (2017) then argue that, under suitable conditions, we have that as \( \frac{\alpha}{N} \to 0 \) this becomes

\[ L_\alpha \to KL(q||p_0) - \frac{1}{\alpha} \sum_{i=1}^{N} \log E_q [f_n(\omega)^\alpha] \]  \hspace{1cm} (22)

For the particular choice \( \alpha = 1 \) (so that we require \( N \to \infty \)) this then becomes

\[ L_{\alpha=1} \to KL(q||p_0) - \sum_{i=1}^{N} \log E_q [f_n(\omega)] \]  \hspace{1cm} (23)

The similarity of Eqn. 23 and Eqn. 11 is now manifest.

For further discussion of EP methods in the context of Gaussian Process inference we refer the reader to (Bui et al., 2017).