Closed-Form Coordinate Ascent Variational Inference for Student-t Process Regression with Student-t Likelihood

Keisuke Onoue **Takatomi Kubo** **Kazushi Ikeda**

Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma, Nara 630-0192, JAPAN
onoue.keisuke.ok2@naist.ac.jp, {takatomi-k, kazushi}@is.naist.jp

Abstract

Combining a Student-t Process prior with a Student-t likelihood yields a doubly robust regression model whose intractable posterior has prevented its practical use. We introduce the first tractable variational inference framework for this model. Leveraging the Student-t distribution's scale-mixture representation, we design a structured variational family that affords an analytic evidence lower bound. To overcome the non-conjugacy of this family, which precludes closed-form updates, we devise a novel projection-based optimization: we find the optimum in a simpler, factorized family and analytically project it back onto our structured one. The framework is extended to a scalable sparse, stochastic setting. Empirical results demonstrate strong performance, particularly in the full-batch setting, establishing this robust model as a practical and powerful tool.

1 INTRODUCTION

Gaussian Processes (GPs) (Williams and Rasmussen, 2006) have established themselves as a cornerstone of non-parametric Bayesian regression. Their flexibility, inherent ability to quantify uncertainty, and the capacity to encode prior beliefs through kernel functions make them highly effective even with small datasets. Consequently, GPs have been extensively applied in diverse domains, including robotics (Deisenroth et al., 2013), geostatistics (Diggle et al., 1998), and computational biology (Romero et al., 2013).

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

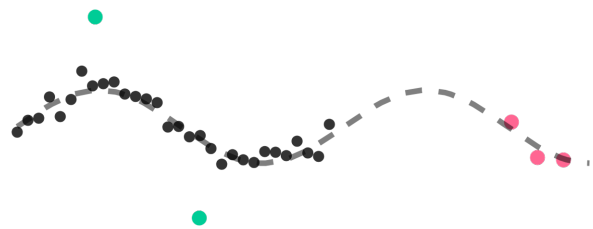


Figure 1: Conceptual illustration of outliers on inputs and outputs. The doubly robust TPRT model automatically discounts target outliers (green dots) through Student-t likelihood and input outliers (pink dots) through the TP prior, accurately capturing the underlying latent function (dashed line) from the group of inlying observations (black dots).

However, the predictive accuracy of GPs can be severely compromised by the presence of outliers. This vulnerability stems from the Gaussian assumptions for both the prior and the likelihood, whose thin tails are sensitive to extreme observations. A natural and powerful strategy to enhance robustness is to replace these Gaussian components with their heavy-tailed counterparts, namely the Student-t distribution. Combining a Student-T Process (TP) prior (Shah et al., 2014), which provides resilience to outliers in the inputs, with a Student-t likelihood for robustness to outliers in the outputs, creates a "doubly robust" regression model, hereafter referred to as the TPRT (T-Process Regression with T-likelihood) model, ideally suited for real-world data (Andrade, 2023) (as conceptually illustrated in Figure 1).

Despite its theoretical appeal, this powerful combination introduces a formidable computational barrier. The TP prior is not conjugate to the Student-t likelihood, rendering the posterior distribution of the latent function analytically intractable. This intractability has been the primary obstacle preventing this highly robust model from being a practical, widely-

used tool for data analysis. Consequently, inference for this model necessitates the use of approximation techniques.

While various approximation methods have been explored, they often face a trade-off between computational efficiency and the expressiveness of the posterior approximation. For instance, the principal method applied directly to this doubly robust model is limited by a restrictive parametric form (Tang et al., 2017). In parallel, more general variational inference (VI) frameworks for TPs have been developed, but they inherently require costly sampling procedures to optimize their objective function, where accuracy is directly tied to the number of Monte Carlo (MC) samples drawn (Xu and Zeng, 2024; Xu et al., 2024). Some of these frameworks are further constrained by the parametric form of their variational posterior (Xu et al., 2024). A scalable inference framework that can efficiently capture the complex posterior of the doubly robust TP model without such compromises has remained an open challenge.

In this work, we overcome this challenge by introducing the first tractable VI framework for the doubly robust TP regression (TPR) model that admits closed-form coordinate ascent updates. Our approach is built upon the scale-mixture representation of the Student-t distribution, which allows us to design a structured variational family that preserves the critical dependency between the latent function and its global scaling variable. To address the non-conjugacy that prevents closed-form updates, we devise a novel, efficient optimization step based on analytic projection: we find the optimum within a simpler, factorized family and then project it back onto our more expressive structured family.

Our main contributions are as follows:

- We propose a new structured VI framework for the doubly robust TP model, which yields an analytically tractable evidence lower bound (ELBO) and predictive distribution.
- We develop a novel projection-based optimization method that enables efficient, closed-form coordinate ascent updates, sidestepping the challenges of non-conjugacy.
- We extend the framework to a sparse and stochastic setting, enabling scalable inference for large datasets.
- Through extensive experiments, we demonstrate that our method achieves state-of-the-art predictive accuracy and faster convergence, particularly in the full-batch setting, establishing the doubly robust TP model as a practical and powerful tool.

Our source code and datasets are publicly available[1].

2 RELATED WORK

Our work integrates and extends several lines of research in robust regression and approximate Bayesian inference (ABI) (Alquier, 2020) for TPs.

2.1 Robust Regression Model Architectures

The TP is a natural generalization of the GP for robust modeling. Early work established its properties, such as its derivation from a hierarchical GP with an Inverse-Wishart prior over the covariance or Inverse-Gamma prior over the scale of kernel function and its status as the most general elliptical process with an analytic density (Shah et al., 2014).

A primary challenge in TPR is handling observation noise. The TPRT model combines a TP prior with an independent Student-t likelihood, explicitly modeling robustness to both input and target outliers (Tang et al., 2017). To preserve tractability, a widely adopted alternative is to absorb the noise variance into the TP's kernel, e.g., using $\mathbf{K}' = \mathbf{K}_{XX} + \sigma^2\mathbf{I}$ as the scale matrix of a single MVT distribution (TPRK model) (Shah et al., 2014). However, this model's robustness can be limited; it was shown that for some common kernels, the predictive mean is identical to that of GP regression (GPR), compromising its effectiveness against outliers (Tang et al., 2016). Our work therefore focuses on developing a scalable inference scheme for the more robust but intractable TPRT model.

2.2 Approximate Inference for Student-t Processes

The intractability of the TPRT model necessitates approximate inference. To date, this specific model has been principally addressed using Laplace Approximation (LA) (Tang et al., 2017). While faster than Markov chain Monte Carlo methods (Shah et al., 2014), this approach is limited by its reliance on a single Gaussian to approximate the posterior, which may fail to capture the true posterior shape.

In parallel, significant progress has been made in developing general-purpose VI frameworks for TPs that approximate the posterior directly with a Student-t distribution, i.e., $q(\mathbf{f}) = \mathcal{ST}(\mathbf{f}; \dots)$ (Xu and Zeng, 2024; Xu et al., 2024). However, this powerful approach faces two key analytical challenges. First, the Kullback-Leibler (KL) divergence term in the ELBO, $\mathrm{KL}(q(\mathbf{f})||p(\mathbf{f}))$, lacks a closed-form solution as

[1] https://github.com/k-onoue/variational-tpr.git

both distributions are Student-t, requiring approximations via MC estimation or local variational bounds. Second, and more fundamentally, the expected log-likelihood term of the ELBO, $\mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{y}|\mathbf{f})]$, is also analytically intractable. This is because the expectation is taken with respect to the Student-t distribution q, a difficulty that arises for nearly any likelihood, including both Gaussian and Student-t. Consequently, optimizing the ELBO in such a framework typically relies on estimating gradients via MC sampling from the variational posterior.

Thus, existing ABI approaches for TPs are either confined to the less expressive LA for the TPRT model, or rely on general-purpose VI that necessitates approximations for both the KL and likelihood terms of the ELBO. Consequently, a tractable ABI framework that leverages the TPRT model's hierarchical scale-mixture structure to achieve a fully analytic ELBO and thus sampling-free, closed-form updates, along with its scalable extension, has remained an open challenge.

Our work directly addresses this gap.

3 PRELIMINARIES

3.1 The Doubly Robust TPRT Model

A general framework for non-linear regression is formulated through a latent variable model. Given a set of input-output pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, we model the observed output y_i using a latent function $f(\mathbf{x}_i)$ and an additive noise term ε_i:

$$y_i = f(\mathbf{x}_i) + \varepsilon_i.$$

The conventional approach, GPR, assumes a GP prior for the latent function f and i.i.d. Gaussian noise for ε_i. However, the thin tails of the Gaussian distribution make GPR sensitive to outliers in both the inputs (affecting f) and the outputs (affecting ε_i).

To enhance robustness, Gaussian components can be replaced by their Student-t counterparts. This doubly robust TPRT regression model, combines a TP prior for the latent function with an independent Student-t likelihood for the observation noise. A TP is a stochastic process where any finite collection of function values $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^\top$ follows a multivariate Student-t distribution (MVT).

Definition 1. *A random vector $\mathbf{f} \in \mathbb{R}^N$ follows a multivariate Student-t distribution, $\mathcal{ST}(\mathbf{f}; \nu, \boldsymbol{\mu}, \mathbf{K})$, with degrees of freedom ν, mean vector $\boldsymbol{\mu}$, and scale matrix $\mathbf{K}$, if its probability density function (PDF) is:*

$$p(\mathbf{f}) = \frac{\Gamma\left(\frac{\nu+N}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)(\nu\pi)^{\frac{N}{2}}|\mathbf{K}|^{\frac{1}{2}}} \times \left(1 + \frac{1}{\nu}(\mathbf{f}-\boldsymbol{\mu})^\top \mathbf{K}^{-1}(\mathbf{f}-\boldsymbol{\mu})\right)^{-\frac{\nu+N}{2}}$$

A key property of the Student-t distribution, which is central to our proposed method, is that it can be expressed as a scale-mixture of Gaussians (Gelman, 2006). Specifically, a random vector $\mathbf{f}$ following an MVT distribution as in Definition 1 can be generated hierarchically by first drawing a scalar precision variable r from a Gamma distribution:

$$r \sim \text{Gamma}\left(r; \frac{\nu}{2}, \frac{\nu}{2}\right), \tag{1a}$$

$$\mathbf{f}|r \sim \mathcal{N}(\mathbf{f}; \boldsymbol{\mu}, r^{-1}\mathbf{K}). \tag{1b}$$

This representation transforms the heavy-tailed distribution into a conditionally Gaussian system, which is often more amenable to inference algorithms.

Despite this useful property, inference in the TPRT model remains a significant challenge. The TP prior is not conjugate to the Student-t likelihood, and the convolution of two Student-t distributions is not analytic. This intractability necessitates the use of approximate inference techniques.

3.2 VI and Coordinate Ascent

VI is a powerful technique for approximating intractable posterior distributions (Bishop and Nasrabadi, 2006), which have been widely explored in complex Bayesian models such as Sparse GP models (Titsias, 2009; Hensman et al., 2013, 2015; Wenzel et al., 2019; Nadew et al., 2024). The core idea of VI is to reframe the inference problem as an optimization problem. We first posit a family of tractable distributions, $q(\boldsymbol{\theta}; \boldsymbol{\phi})$, over the latent variables $\boldsymbol{\theta}$, parameterized by variational parameters $\boldsymbol{\phi}$. Then, we find the optimal parameters $\boldsymbol{\phi}^*$ that make $q(\boldsymbol{\theta}; \boldsymbol{\phi}^*)$ closest to the true posterior $p(\boldsymbol{\theta}|\mathbf{y})$. Closeness is typically measured by the KL divergence, $\text{KL}(q||p)$.

Minimizing this KL divergence is equivalent to maximizing the ELBO, $\mathcal{L}(q)$:

$$\begin{aligned}\mathcal{L}(q) &= \mathbb{E}_{q(\boldsymbol{\theta})}[\log p(\mathbf{y}, \boldsymbol{\theta})] - \mathbb{E}_{q(\boldsymbol{\theta})}[\log q(\boldsymbol{\theta})] \\ &= \log p(\mathbf{y}) - \text{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathbf{y})).\end{aligned}$$

The ELBO is a lower bound on the log marginal likelihood, $\log p(\mathbf{y})$. Maximizing it thus makes our approximation q a better fit for the posterior.

Mean-Field Assumption and Coordinate Ascent. The choice of the variational family for q is critical. The mean-field approximation assumes that the latent variables can be partitioned into disjoint sets, and the variational distribution factorizes over these partitions: $q(\boldsymbol{\theta}) = \prod_{i=1}^{M} q_i(\boldsymbol{\theta}_i)$. This factorization can enhance the tractability. The traditional algorithm to maximize the ELBO under this assumption is Coordinate Ascent VI (CAVI). In CAVI, we iteratively optimize the parameters for each factor $q_j(\boldsymbol{\theta}_j)$ while holding the others fixed. The optimal solution for a given factor q_j^* takes the form:

$$\log q_j^*(\boldsymbol{\theta}_j) = \mathbb{E}_{i \neq j}[\log p(\mathbf{y}, \boldsymbol{\theta})] + \text{const.} \tag{2}$$

The expectation is taken with respect to all other factors $q_i(\boldsymbol{\theta}_i)$ for $i \neq j$. A key benefit arises when the model exhibits conditional conjugacy. If the complete conditional distribution $p(\boldsymbol{\theta}_j | \boldsymbol{\theta}_{-j}, \mathbf{y})$ belongs to the exponential family, then the optimal variational distribution $q_j^*(\boldsymbol{\theta}_j)$ will belong to the same exponential family. This allows the update for its parameters to be computed in closed form.

4 A CLOSED-FORM CAVI FRAMEWORK FOR TPRT

In this section, we introduce our novel VI framework for the TPRT model. We first reformulate the generative model using data augmentation and then detail our structured posterior and the efficient, closed-form update procedure.

4.1 Augmented Generative Model

We begin with the TPRT model. The prior over the latent function $\mathbf{f}$ is a zero-mean TP with degrees of freedom ν_f and a kernel matrix $\mathbf{K}_{XX} = k(\mathbf{X}, \mathbf{X})$. The likelihood for the observations $\mathbf{y}$ is an independent Student-t distribution with degrees of freedom ν_ε and scale σ:

$$\begin{aligned} \mathbf{f} &\sim \mathcal{ST}(\mathbf{f}; \nu_f, \mathbf{0}, \mathbf{K}_{XX}), \\ y_i | f_i &\sim \mathcal{ST}(y_i; \nu_\varepsilon, f_i, \sigma^2) \quad (i = 1, \ldots, N). \end{aligned}$$

Using the scale-mixture representation of the Student-t distribution, we can rewrite this model in an equivalent hierarchical form using Gamma-distributed auxiliary variables r and $\boldsymbol{\lambda} = \{\lambda_i\}_{i=1}^N$:

$$\begin{aligned} r &\sim \text{Gamma}\left(r; \frac{\nu_f}{2}, \frac{\nu_f}{2}\right), \\ \mathbf{f} | r &\sim \mathcal{N}(\mathbf{f}; \mathbf{0}, r^{-1}\mathbf{K}_{XX}), \\ \lambda_i &\sim \text{Gamma}\left(\lambda_i; \frac{\nu_\varepsilon}{2}, \frac{\nu_\varepsilon}{2}\right), \\ y_i | f_i, \lambda_i &\sim \mathcal{N}(y_i; f_i, \lambda_i^{-1}\sigma^2) \quad (i = 1, \ldots, N). \end{aligned}$$

This augmented representation is beneficial for VI because all conditional distributions in the joint probability are now in the exponential family.

4.2 Structured Variational Posterior

We propose a structured mean-field variational family (Saul and Jordan, 1995) for the posterior distribution over the latent variables $q(\mathbf{f}, r, \boldsymbol{\lambda})$. We assume independence between the global variable set $(\mathbf{f}, r)$ and the local variables $\boldsymbol{\lambda}$, but crucially, we maintain the dependency between $\mathbf{f}$ and r:

$$q(\mathbf{f}, r, \boldsymbol{\lambda}) = q(\mathbf{f}, r) \prod_{i=1}^{N} q(\lambda_i).$$

Here, $q(\mathbf{f}, r)$ is a Gaussian-Gamma distribution and each $q(\lambda_i)$ is a Gamma distribution, parameterized as follows:

$$\begin{aligned} q(\mathbf{f}, r) &= \mathcal{N}(\mathbf{f}; \mathbf{m}, r^{-1}\mathbf{S})\text{Gamma}(r; \alpha_r, \beta_r), \\ q(\lambda_i) &= \text{Gamma}(\lambda_i; \alpha_{\lambda_i}, \beta_{\lambda_i}). \end{aligned}$$

This structural assumption is more expressive than a fully factorized approximation $q(\mathbf{f})q(r)q(\boldsymbol{\lambda})$ and is key to deriving an analytic predictive distribution. Given this family, the ELBO decomposes as:

$$\begin{aligned} \mathcal{L}(q) = &\sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{f}, r) \prod_{i=1}^{N} q(\lambda_i)}[\log p(y_i | f_i, \lambda_i)] \\ &- \text{KL}(q(\mathbf{f}, r) || p(\mathbf{f}, r)) - \sum_{i=1}^{N} \text{KL}(q(\lambda_i) || p(\lambda_i)). \end{aligned}$$

Each component of the ELBO is analytically tractable; a detailed derivation is provided in Appendix B.2.

4.3 Closed-Form Updates via Projection

We maximize the ELBO with respect to the variational parameters by coordinate ascent, alternating between updates for $q(\lambda_i)$ and $q(\mathbf{f}, r)$.

Update for $q(\lambda_i)$. The optimal distribution for each local variable $q(\lambda_i)$ is conditionally conjugate. The update rules are derived from the standard CAVI equation. Letting m_i be the i-th element of the variational mean $\mathbf{m}$ and S_{ii} be the i-th diagonal element of the variational covariance $\mathbf{S}$ from $q(\mathbf{f}, r)$, the updates are:

$$\begin{aligned} \alpha_{\lambda_i} &\leftarrow \frac{\nu_\varepsilon + 1}{2}, \\ \beta_{\lambda_i} &\leftarrow \frac{\nu_\varepsilon}{2} + \frac{1}{2\sigma^2}\left((y_i - m_i)^2 + \frac{\beta_r}{\alpha_r - 1} S_{ii}\right). \end{aligned}$$

Approximate Update for $q(\mathbf{f}, r)$. The primary challenge is that the true conditional posterior $p(\mathbf{f}, r|\mathbf{y}, \boldsymbol{\lambda})$ is not in the Gaussian-Gamma family, meaning a direct CAVI update for $q(\mathbf{f}, r)$ is not available in closed form. We devise a novel two-step procedure to find an efficient, closed-form approximate update.

First, we compute the optimal parameters for a temporary, fully factorized distribution,

$$\tilde{q}(\mathbf{f}, r) = \tilde{q}(\mathbf{f})\tilde{q}(r) = \mathcal{N}(\mathbf{f}; \tilde{\mathbf{m}}, \tilde{\mathbf{S}})\mathrm{Gamma}(r; \tilde{\alpha}_r, \tilde{\beta}_r).$$

This distribution $\tilde{q}$ serves as an analytically tractable proxy for the true optimal conditional posterior that we would ideally use as the target in a CAVI update.

Second, we update our structured variational posterior $q(\mathbf{f}, r)$ to be as close as possible to this tractable target $\tilde{q}$. This is achieved by solving the following minimization problem:

$$\arg\min_{q} \mathrm{KL}(\tilde{q}(\mathbf{f}, r)||q(\mathbf{f}, r)), \tag{3}$$

which is known as moment projection (Murphy, 2012). As derived in Appendix E, this projection problem has a closed-form solution. The resulting parameter updates for $q(\mathbf{f}, r)$ are:

$$\begin{aligned}
\mathbf{m} &\leftarrow \tilde{\mathbf{m}}, \\
\mathbf{S} &\leftarrow \mathbb{E}_{\tilde{q}(r)}[r] \cdot \tilde{\mathbf{S}} = \frac{\tilde{\alpha}_r}{\tilde{\beta}_r}\tilde{\mathbf{S}}, \\
\alpha_r &\leftarrow \tilde{\alpha}_r, \\
\beta_r &\leftarrow \tilde{\beta}_r,
\end{aligned}$$

where the parameters for the target $\tilde{q}$ are computed via standard CAVI updates detailed in Appendix B.3. This two-step process provides a principled and efficient closed-form update for our expressive structured posterior.

4.4 Overall Optimization Procedure

The complete inference procedure follows a Variational Expectation-Maximization (EM) algorithm (Jordan et al., 1999). We group the model hyperparameters (e.g., kernel parameters, degrees of freedom) into a single vector $\boldsymbol{\theta}$, and the variational parameters (e.g., $\mathbf{m}, \mathbf{S}, \alpha_r, \beta_r$) into a vector $\boldsymbol{\phi}$. Given the training data $(\mathbf{X}, \mathbf{y})$, we alternate between optimizing $\boldsymbol{\phi}$ (E-step) and $\boldsymbol{\theta}$ (M-step). The procedure is summarized in Algorithm 1.

4.5 A Conceptual View: Alternating Scale and GP Inference

While our framework is formally derived through coordinate ascent on the augmented model, it offers an intuitive interpretation that bridges the gap between heavy-tailed processes and Gaussian modeling. Conceptually, the E-step can be viewed as an alternating process between *scale estimation* and *GP inference*. Specifically, for any fixed values of the auxiliary precision variables r and $\boldsymbol{\lambda}$, the conditional model of the latent function $\mathbf{f}$ is exactly a GP. The updates for $q(\lambda_i)$ and $q(r)$ essentially estimate the local and global scale of the noise and the latent function, respectively, while the projection-based update for $q(\mathbf{f})$ performs inference on the resulting scaled GP. This perspective clarifies why our predictive mean and covariance retain the familiar forms of GPR.

Algorithm 1 Variational EM Algorithm for the TPRT Model

Require: Data $\{(\mathbf{X}, \mathbf{y})\}$, learning rate η

1: Initialize hyperparameters $\boldsymbol{\theta}$ and variational parameters $\boldsymbol{\phi}$.
2: **while** ELBO has not converged **do**
3: **E-step**:
  a. Update local variational parameters $\{\alpha_{\lambda_i}, \beta_{\lambda_i}\}_{i=1}^N$.
  b. Update global variational parameters $\{\mathbf{m}, \mathbf{S}, \alpha_r, \beta_r\}$ using the projection method.
4: **M-step**: Update hyperparameters $\boldsymbol{\theta}$ via gradient ascent on the ELBO:
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta\nabla_{\boldsymbol{\theta}}\mathcal{L}(q)$.
5: **end while**
6: **return** Optimized parameters $\boldsymbol{\phi}^*$ and $\boldsymbol{\theta}^*$.

4.6 Analytic Predictive Distribution

A key benefit of our structured posterior is that the predictive distribution for the latent function value f_* at a new test point $\mathbf{x}_*$ can be computed analytically. The true predictive posterior is approximated by marginalizing over our learned variational posterior $q(\mathbf{f}, r)$:

$$\begin{aligned}
p(f_*|\mathbf{y}) &= \int p(f_*|\mathbf{f}, r)p(\mathbf{f}, r|\mathbf{y})d\mathbf{f}dr \\
&\approx \int p(f_*|\mathbf{f}, r)q(\mathbf{f}, r)d\mathbf{f}dr =: q(f_*).
\end{aligned}$$

Remarkably, this integral is tractable and results in a univariate Student-t distribution, $q(f_*) = \mathcal{ST}(f_*; \nu_*, \mu_*, \sigma_*^2)$. We define the relevant kernel covariances as $\mathbf{k}_{*X} = k(\mathbf{x}_*, \mathbf{X})$ and $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$. The parameters are then given by:

$$\begin{aligned}
\nu_* &= 2\alpha_r, \\
\mu_* &= \mathbf{k}_{*X}\mathbf{K}_{XX}^{-1}\mathbf{m}, \\
\sigma_*^2 &= \frac{\beta_r}{\alpha_r}\left(k'_{**} + \mathbf{k}_{*X}\mathbf{K}_{XX}^{-1}\mathbf{S}\mathbf{K}_{XX}^{-1}\mathbf{k}_{X*}\right),
\end{aligned}$$

where $k'_{**} = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_{*X}\mathbf{K}_{XX}^{-1}\mathbf{k}_{X*}$. Note that while the predictive distribution for the latent function f_* is analytic, the predictive distribution for a new observation y_* requires a further approximation as it involves a convolution of two Student-t distributions. The detailed derivation is given in Appendix B.4.

5 SCALABLE EXTENSION WITH A SPARSE MODEL

The inference framework developed in Section 4 has a computational complexity of $\mathcal{O}(N^3)$, making it unsuitable for large datasets. To address this scalability challenge, we introduce two key extensions. First, we develop a sparse version of our model using inducing points to reduce the computational complexity with respect to the number of data points. Second, we adapt our update rules for stochastic optimization, allowing the model to be trained efficiently on mini-batches of data.

5.1 Sparse Variational Model with Inducing Points

Following established sparse GP literature (Titsias, 2009), we introduce a set of M inducing variables $\mathbf{u} = f(\mathbf{Z})$ at locations $\mathbf{Z}$. The joint prior over the function values $\mathbf{f}$ at the training data locations and $\mathbf{u}$ is a multivariate Student-t distribution (Xu and Zeng, 2024), which can be equivalently written in its scale-mixture form:

$$p(\mathbf{f}, \mathbf{u}) = \int \mathcal{N}\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{u}\end{bmatrix}; \mathbf{0}, r^{-1}\begin{bmatrix}\mathbf{K}_{XX} & \mathbf{K}_{XZ}\\ \mathbf{K}_{ZX} & \mathbf{K}_{ZZ}\end{bmatrix}\right) \times \text{Gamma}\left(r; \frac{\nu_f}{2}, \frac{\nu_f}{2}\right) dr.$$

To approximate the true posterior $p(\mathbf{f}, \mathbf{u}, r, \boldsymbol{\lambda}|\mathbf{y})$, we construct a variational posterior $q(\mathbf{f}, \mathbf{u}, r, \boldsymbol{\lambda}) = q(\mathbf{f})q(\mathbf{u}, r)\prod q(\lambda_i)$ and set the conditional component of the full variational distribution equal to the true conditional prior, $q(\mathbf{f}|\mathbf{u}, r) := p(\mathbf{f}|\mathbf{u}, r)$. This choice leads to a simplified ELBO where the term involving $p(\mathbf{f}|\mathbf{u}, r)$ cancels out:

$$\mathcal{L}(q) = \sum_{i=1}^{N} \mathbb{E}_{p(\mathbf{f}|\mathbf{u},r)q(\mathbf{u},r)\prod_{i=1}^{N} q(\lambda_i)}[\log p(y_i|f_i, \lambda_i)] - \text{KL}(q(\mathbf{u}, r)||p(\mathbf{u}, r)) - \sum_{i=1}^{N} \text{KL}(q(\lambda_i)||p(\lambda_i)).$$

The update equations follow the same projection-based procedure as the base model, with complexity reduced to $\mathcal{O}(NM^2)$. The full derivations are provided in Appendix C.

5.2 Stochastic Optimization

To scale our sparse model to very large datasets, we employ stochastic optimization for both the Variational E-step and M-step, following the principles of stochastic VI (SVI) (Hoffman et al., 2013). This allows the model to be trained efficiently on mini-batches of data.

In the E-step, we compute an approximate update for the global variational parameters of $q(\mathbf{u}, r)$. For each mini-batch, we first find the optimal parameters for the proxy distribution $\tilde{q}$ and then project them to obtain the target parameters for our structured posterior $q(\mathbf{u}, r)$. These target parameters, converted to their natural parameter representation $\boldsymbol{\eta}_{\text{target}}$, define the direction of an approximate natural gradient step. As the coordinate ascent update corresponds to a natural gradient step of unit length in the conjugate exponential family (Hoffman et al., 2013), the global parameters are updated using weighted averaging:

$$\boldsymbol{\eta}_{\text{global}}^{(t+1)} \leftarrow (1 - \rho_t)\boldsymbol{\eta}_{\text{global}}^{(t)} + \rho_t \boldsymbol{\eta}_{\text{target}},$$

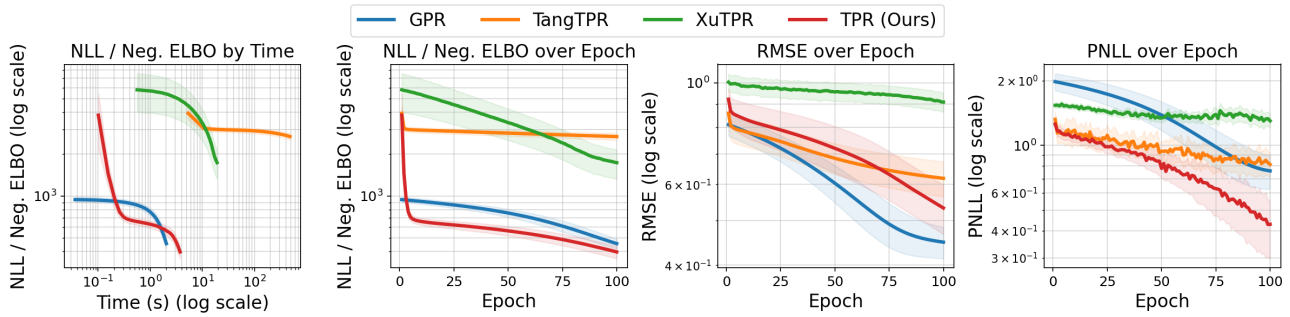
where ρ_t is the learning rate. In the M-step, the model hyperparameters $\boldsymbol{\theta}$ are updated by taking a gradient step on a stochastic estimator of the ELBO, computed using the same mini-batch. The necessary derivations, including the natural parameterization of the Gaussian-Gamma distribution, are detailed in Appendix C and D.

6 EXPERIMENTS

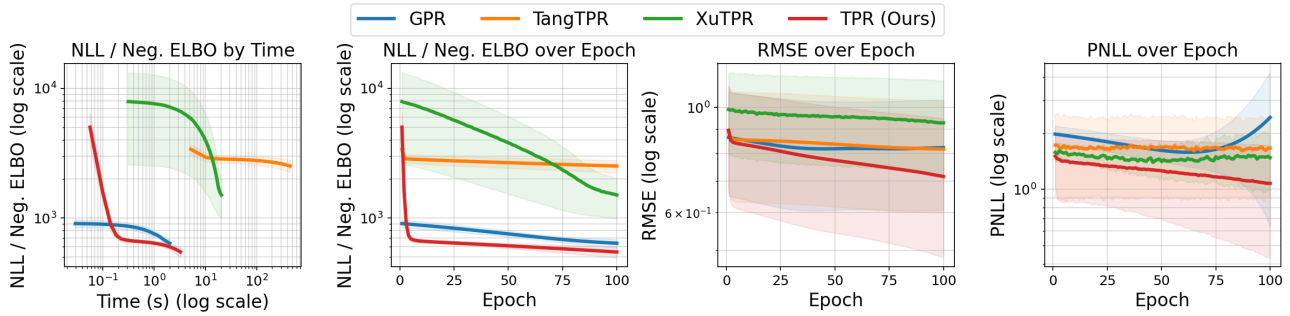
6.1 Experimental Setup

To comprehensively evaluate the performance of our proposed method, we conducted experiments on a wide range of benchmark datasets drawn from previous works on TPR (Tang et al., 2017; Xu and Zeng, 2024). The datasets were categorized into two groups based on their scale: one for full-batch models and another for sparse models. All input features and target values were standardized before training.

The suitability of these benchmarks for evaluating a robust model like TPRT is well-established. Prior analyses have confirmed that these datasets exhibit non-Gaussian properties indicative of naturally occurring outliers. For instance, Tang et al. (2017) demonstrated through kernel density estimation that the input features of several benchmarks possess heavy tails. Similarly, Xu and Zeng (2024) showed that the target variables of datasets like Yacht and Taxi are highly skewed and heavy-tailed. This established presence of natural outliers makes these datasets an ideal testbed for assessing the performance of our inference method,



(a) Performance on the original Boston dataset.



(b) Performance on the Boston dataset with artificial outliers.

Figure 2: Learning curves for all full-batch models on the Boston benchmark (a) and with artificial outliers (b). Within the figure, the four panels of each row show: (from left to right) the model's objective function (NLL/Neg. ELBO) vs. wall-clock time, the objective function vs. epochs, RMSE vs. epochs, and PNLL vs. epochs. The solid lines represent the mean over 10 folds, with shaded areas indicating one standard deviation. These plots demonstrate that our TPR model (red) is competitive on clean data and achieves superior accuracy and faster convergence in the presence of outliers for a dataset.

as it is applied to a model specifically designed for such challenging conditions.

To further test the limits of robustness, we also evaluated performance on artificially contaminated data. For input outliers, we used the "Neal with input outliers" dataset, following (Tang et al., 2017). For all other datasets, we focused on robustness to output outliers by creating contaminated versions of the target values following the procedure from (Xu and Zeng, 2024): 5% of the samples were perturbed by adding three times the standard deviation of the target values. In the following result tables (e.g., Table 1), these output-contaminated datasets are denoted by "Yes" in the "Out." (Outlier) column to distinguish them from the original ("No") or input-contaminated ("In") settings.

All experiments were evaluated using a 10-fold cross-validation scheme. Detailed specifications of each benchmark task are provided in Appendix F.

Baselines. We compare our proposed method against several key baselines. In the full-batch setting, we compare against: (1) **GPR**, a baseline GPR model with a Gaussian likelihood, serving as a non-robust baseline; (2) **TangTPR** (Tang et al., 2017), a doubly robust TPRT model that employs an LA for inference; and (3) **XuTPR** (Xu and Zeng, 2024), a VI-based TPRT model that uses a Student-t variational posterior and MC estimation for optimization, which we adapted to a full-batch setting. Our proposed full-batch model is denoted as **TPR**.

In the sparse setting, we compare against: (1) **SparseGPR**, a baseline variational sparse GP model with a Gaussian likelihood; and (2) **XuSparseTPR** (Xu and Zeng, 2024), the sparse VI-based TPRT model as originally proposed. Our proposed sparse model is denoted as **SparseTPR**.

Evaluation Metrics and Optimization. Predictive performance is primarily evaluated using Root Mean Squared Error (RMSE) and Predictive Negative Log-Likelihood (PNLL). Objective function values (negative log marginal likelihood for GPR/TangTPR, negative ELBO for VI-based methods) are also reported to visualize optimization progress but are

Table 1: Comprehensive performance comparison. The left side shows full-batch models, and the right side shows sparse models. We report the mean value over 10-fold cross-validation for RMSE and PNLL. Values are shown to two decimal places, except where three digits are needed to show performance gaps; values ≥ 10 are rounded to integers. The **best**-performing model for each task is highlighted in **bold**.

Full-batch Models

		RMSE				PNLL			
Task	**Out.**	GPR	Tang	Xu	Ours	GPR	Tang	Xu	Ours
Bost.	No	**0.45**	0.62	0.91	0.53	0.76	0.82	1.30	**0.43**
	Yes	0.82	0.82	0.93	**0.72**	2.43	1.66	1.48	**1.08**
Diab.	No	1.15	1.05	1.01	**0.97**	14	2.91	**1.63**	1.95
	Yes	1.58	1.05	1.03	**1.01**	4.19	2.47	2.23	**1.93**
ELE	No	1.70	0.96	0.72	**0.67**	108	3.27	1.38	**0.82**
	Yes	2.96	0.81	0.89	**0.67**	640	3.00	1.31	**0.88**
MPG	No	0.55	0.53	0.67	**0.40**	2.00	0.90	0.93	**0.32**
	Yes	0.77	0.65	0.70	**0.52**	3.05	1.65	**1.03**	1.25
Mach.	No	**0.657**	0.77	0.659	0.665	46	0.98	0.31	**0.13**
	Yes	2.59	0.80	0.73	**0.68**	181	1.04	0.90	**0.36**
Neal	No	0.52	0.49	0.66	**0.45**	9.40	0.77	0.31	**-0.12**
	In.	0.94	0.470	0.73	**0.468**	23	0.90	0.21	**-0.12**
	Out.	5.75	0.46	0.63	**0.42**	218	0.50	0.56	**-0.07**
Yacht	No	**0.04**	0.22	0.64	0.18	**-2.20**	-0.03	0.30	-0.65
	Yes	1.52	0.51	0.57	**0.40**	2657	0.37	0.62	**-0.14**

Sparse Models

		RMSE			PNLL		
Task	**Out.**	GPR	Xu	Ours	GPR	Xu	Ours
Bike	No	**0.25**	0.29	0.66	3.73	**-0.21**	0.82
	Yes	0.46	**0.44**	0.73	41	1.20	**0.95**
Conc.	No	0.321	0.328	**0.315**	1.45	**0.09**	0.19
	Yes	0.63	0.47	**0.46**	38	1.04	**0.64**
Elev.	No	**0.30**	0.32	0.33	28	**0.13**	0.34
	Yes	0.48	**0.45**	0.47	394	1.05	**0.75**
Ener.	No	0.058	0.053	**0.045**	-0.91	-0.99	**-1.12**
	Yes	0.60	0.26	**0.25**	47	**-0.50**	0.43
Kin8.	No	**0.305**	0.307	0.53	2.88	**0.15**	0.83
	Yes	0.51	**0.45**	0.62	35	**0.42**	1.08
Prot.	No	**0.69**	0.75	0.79	20	**0.95**	1.22
	Yes	**0.76**	0.79	0.82	71	**1.29**	1.41
Taxi	No	0.02	0.11	**0.01**	-0.03	-1.27	**-1.35**
	Yes	0.39	0.28	**0.22**	25272	**-0.78**	19

not used for direct performance comparison across different model classes due to their inherent differences. To ensure stable training, all model hyperparameters were optimized using Maximum a Posteriori (MAP) estimation with weakly informative priors. Our proposed models (TPR, SparseTPR), TangTPR, and SparseGPR employ an alternating optimization scheme, where variational parameters (E-step) and hyperparameters (M-step) are updated in separate steps. For the M-step, all models used the Adam optimizer (Kingma, 2014) with a fixed learning rate of 0.01. For the E-step, the variational updates for SparseGPR and our SparseTPR used a natural gradient-based step with a fixed learning rate of 0.1. The optimization for non-sparse, full-batch models was run for 100 iterations, and for sparse, mini-batch models, 1000 epochs. For all sparse models, we used a fixed mini-batch size of 1024 and set the number of inducing points to 256. Further details on specific priors and implementation are provided in Appendix G and H.

6.2 Results on Full-Batch Regression Tasks

We first present the results for our full-batch model (TPR). Representative learning curves for the Boston benchmark are shown in Figure 2 to visually illustrate our key findings. The complete quantitative results are available in Table 6, with learning curves for all tasks shown in Figure 4.

Our proposed TPR outperformed the baselines, achieving the lowest RMSE on nearly all benchmark tasks. It was surpassed only by GPR on three datasets without artificially introduced outliers: Boston, Machine CPU, and Yacht. However, even on these clean datasets, our TPR consistently achieves lower PNLL than GPR and TangTPR (see Table 1), indicating superior uncertainty quantification and reliability of its predictions. On all other tasks, including all benchmarks with explicitly added input or output outliers, our method achieved superior predictive accuracy in both RMSE and PNLL . Table 1 provides a clear visual summary of this trade-off, showing GPR's competitiveness on the original, clean data and our model's decisive advantage in convergence and accuracy in the presence of outliers. The massive values of PNLL for GPR in outlier settings (e.g., ELE or Yacht) highlight its significant failure to accommodate extreme observations, whereas our TPR maintains stable and low PNLL values across all conditions.

Furthermore, our method exhibits faster convergence compared to other TPR models. The plots in Figure 2 visually confirm this, showing that our model consistently achieves lower objective function values, RMSE and PNLL in fewer iterations and less wall-clock time than TangTPR and XuTPR.

6.3 Results on Sparse Regression Tasks

Next, we evaluate our scalable sparse model (SparseTPR) on large-scale datasets. Detailed results are presented in Table 7 and Figure 5.

While the performance gap is less pronounced than in the full-batch setting, our SparseTPR remains highly competitive and robust. It achieved the highest number of first-place rankings in terms of RMSE, securing the best performance on tasks such as Concrete, Energy, and notably, the largest dataset, Taxi, in both original and outlier-contaminated versions. Regarding PNLL, while our method is still competitive, XuSparseTPR occasionally shows lower values in certain settings (e.g., Bike). This trade-off suggests that while our projection-based updates are highly effective for mean estimation, the approximation in the SVI setting may leave room for further refinement in uncertainty quantification.

A key finding is the enhanced robustness of SparseTPR. When comparing performance on a dataset with and without artificial outliers, our model consistently showed a smaller degradation in RMSE compared to SparseGPR and XuSparseTPR. This suggests that our framework maintains its robustness advantage even in the stochastic, sparse setting, effectively mitigating the impact of outliers.

6.4 Discussion

The experimental results validate the effectiveness of our proposed variational inference framework for the doubly robust TPRT model. This success is particularly evident in the robustness evaluation using PNLL, where our method consistently outperforms baselines even in full-batch scenarios where RMSE gains are marginal.

The remarkable success in the full-batch setting can be attributed to our structured variational posterior. By leveraging data augmentation, we designed a Gaussian-Gamma variational family that retains the crucial dependency between the latent function and its global precision variable. This allows our model to have a representational capacity comparable to a direct Student-t posterior, while the projection-based CAVI update provides an efficient, closed-form optimization path. This combination of an expressive posterior and an efficient update scheme is a primary driver of the superior predictive density and faster convergence observed. As suggested during the review process, this inference procedure can be intuitively interpreted as alternating between estimating scaling variables and performing Gaussian process inference.

In the sparse setting, while still competitive, the performance margin of SparseTPR was smaller. While our method achieved top-tier RMSE in many cases, it was surpassed in PNLL by sampling-based methods in several tasks. We hypothesize this is related to the interaction between our approximate update and the stochastic optimization. In SVI, the gradient of the data-dependent term in the ELBO must be scaled to form an unbiased estimator. Our closed-form update approximates the natural gradient, and this scaling may amplify the approximation error introduced by the moment projection step, especially when computed on a small mini-batch. In the full-batch setting, this scaling factor is unity, and the error is solely from the approximation itself, allowing the model to better capture the complex posterior shape.

7 CONCLUSION

We proposed a closed-form coordinate ascent variational inference framework for the doubly robust TPRT model. Our method leverages a scale-mixture representation and a novel projection-based approximation to derive an efficient, closed-form optimization procedure for a structured variational posterior. Our experiments on a comprehensive suite of benchmarks demonstrated that this approach significantly improves both predictive accuracy and convergence speed over existing methods in the presence of outliers. The framework proved especially effective in the full-batch setting. Future work includes refining the stochastic optimization for the sparse model to mitigate the potential error amplification from mini-batch scaling. Furthermore, extending this efficient inference framework to other TP-based models, such as multi-output regression and classification, is a promising direction.

Acknowledgements

We deeply appreciate Dr. Jian Xu at RIKEN iTHEMS and RIKEN AIP for his insightful advice at an early stage of this work, and Assoc. Prof. Ryosuke Kojima at Kyoto University and RIKEN BDR for his dedicated mentorship to the first author.

This work was supported by JST, CREST Grant Number JPMJCR22P2, Japan.

References

Alquier, P. (2020). Approximate bayesian inference.

Andrade, J. A. A. (2023). On the robustness to outliers of the student-t process. *Scandinavian Journal of Statistics*, 50(2):725–749.

Asuncion, A., Newman, D., et al. (2007). Uci machine learning repository.

Balandat, M., Karrer, B., Jiang, D., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. (2020). Botorch: A framework for efficient monte-carlo bayesian optimization. *Advances in neural information processing systems*, 33:21524–21538.

Bishop, C. M. and Nasrabadi, N. M. (2006). *Pattern recognition and machine learning*. Springer.

Deisenroth, M. P., Fox, D., and Rasmussen, C. E. (2013). Gaussian processes for data-efficient learning in robotics and control. *IEEE transactions on pattern analysis and machine intelligence*, 37(2):408–423.

Derrac, J., Garcia, S., Sanchez, L., and Herrera, F. (2015). Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *J. Mult. Valued Logic Soft Comput*, 17:255–287.

Diggle, P. J., Tawn, J. A., and Moyeed, R. A. (1998). Model-based geostatistics. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 47(3):299–350.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by browne and draper). *Bayesian Analysis*, 1(3):515–534.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*.

Hensman, J., Matthews, A., and Ghahramani, Z. (2015). Scalable variational gaussian process classification. In *Artificial intelligence and statistics*, pages 351–360. PMLR.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *the Journal of machine Learning research*, 14(1):1303–1347.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lee, S. Y. (2022). The use of a log-normal prior for the student t-distribution. *Axioms*, 11(9):462.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Nadew, Y. Y., Fan, X., and Quinn, C. J. (2024). Conditionally-conjugate gaussian process factor analysis for spike count data via data augmentation. *arXiv preprint arXiv:2405.11683*.

Neal, R. M. (1997). Monte carlo implementation of gaussian process models for bayesian regression and classification. *arXiv preprint physics/9701026*.

Romero, P. A., Krause, A., and Arnold, F. H. (2013). Navigating the protein fitness landscape with gaussian processes. *Proceedings of the National Academy of Sciences*, 110(3):E193–E201.

Saul, L. and Jordan, M. (1995). Exploiting tractable substructures in intractable networks. *Advances in neural information processing systems*, 8.

Shah, A., Wilson, A., and Ghahramani, Z. (2014). Student-t processes as alternatives to gaussian processes. In *Artificial intelligence and statistics*, pages 877–885. PMLR.

Tang, Q., Niu, L., Wang, Y., Dai, T., An, W., Cai, J., and Xia, S.-T. (2017). Student-t process regression with student-t likelihood. In *International Joint Conference on Artificial Intelligence 2017*, pages 2822–2828. Association for the Advancement of Artificial Intelligence (AAAI).

Tang, Q., Wang, Y., and Xia, S.-T. (2016). Student-t process regression with dependent student-t noise. In *ECAI 2016*, pages 82–89. IOS Press.

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR.

Wenzel, F., Galy-Fajou, T., Donner, C., Kloft, M., and Opper, M. (2019). Efficient gaussian process classification using pòlya-gamma data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5417–5424.

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*. MIT press Cambridge, MA.

Xu, J. and Zeng, D. (2024). Sparse variational student-t processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16156–16163.

Xu, J., Zeng, D., and Paisley, J. (2024). Information geometry and beta link for optimizing sparse variational student-t processes. *arXiv preprint arXiv:2408.06699*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **[Yes]** The mathematical setting and the TPRT model are detailed in Section 3.1. Our proposed variational family and inference algorithm are described in Section 4.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **[Yes]** The computational complexity of the full-batch model ($\mathcal{O}(N^3)$) is stated at the beginning of Section 5, and the complexity of the sparse model ($\mathcal{O}(NM^2)$) is given at the end of Section 5.1.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **[Yes]** We provide an anonymized URL to our source code at the end of Section 1. Appendix H provides further implementation details.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **[Yes]** Our primary theoretical contribution is the derivation of a tractable inference framework. The assumptions, including the scale-mixture representation and our structured mean-field variational posterior, are explicitly stated in Sections 4.1 and 4.2.
   (b) Complete proofs of all theoretical results. **[Yes]** The complete derivations for our closed-form updates, ELBO, and predictive distribution are provided in the main text (Section 4) and detailed extensively in the appendices (e.g., Appendix B.2, E, B.3).
   (c) Clear explanations of any assumptions. **[Yes]** We explain the motivation for our key assumption, the structured variational posterior, in Section 4.2, highlighting its benefits for expressiveness and tractability.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **[Yes]** Our anonymized source code is provided via a URL (Section 1). All datasets are publicly available from standard repositories (cited in Appendix F), and we provide specific preprocessing details for the Taxi dataset.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **[Yes]** We provide all training details in Section 6.1. Hyperparameter choices and priors are detailed in Appendix G.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **[Yes]** Our primary evaluation metric, RMSE, is defined in Section 6.1. All tables and figure captions explicitly state that results are reported as mean $\pm$ standard deviation over a 10-fold cross-validation scheme.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **[Yes]** A complete description of the computing infrastructure used for our experiments is provided in Appendix H.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. **[Yes]** We use public benchmark datasets and cite their original sources in Appendix F.
   (b) The license information of the assets, if applicable. **[No]** We use standard, publicly available academic datasets (UCI, KEEL), which are common benchmarks in the field. While we cite their sources, we do not explicitly state the license for each one in the paper.
   (c) New assets either in the supplemental material or as a URL, if applicable. **[Yes]** Our source code is provided as a new asset via an anonymized URL, as mentioned in Section 1.
   (d) Information about consent from data providers/curators. **[Not Applicable]** The datasets used are public benchmarks from established repositories and do not contain sensitive user information requiring specific consent.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **[Not Applicable]** The datasets used are standard regression benchmarks and do not contain personally identifiable information or offensive content.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. **[Not Applicable]** Our research does not involve crowdsourcing or human subjects.

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **[Not Applicable]** Our research does not involve crowdsourcing or human subjects.

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **[Not Applicable]** Our research does not involve crowdsourcing or human subjects.

Supplementary Material

Table of Contents

A Graphical Models 14

B Detailed Derivations for TPRT 14

C Detailed Derivations for Sparse TPRT 17

D Exponential Family Representation of the Gaussian-Gamma Distribution 20

E Projection from Factorized to Structured Distribution 21

F Specifications of Benchmark Tasks 21

G Specifications of Baseline Models 22

H Experimental Details 23

I Ablation Study on an LA-Based TPR Baseline (Tang et al., 2017) 23

J Ablation Study on an VI-Based TPR Baseline (Xu et al., 2024) 25

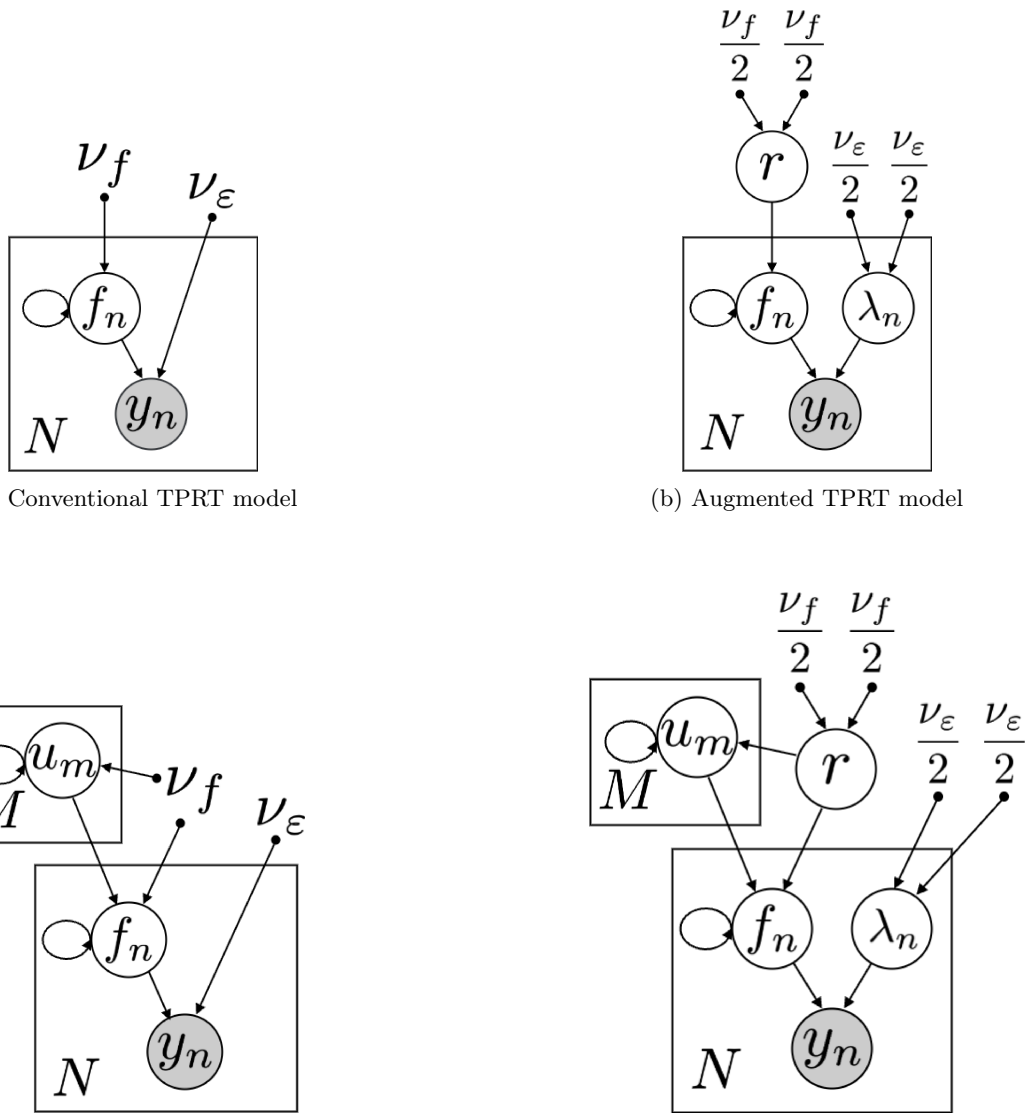
K Detailed Experimental Results 27

L Ablation Study on Optimization Iterations 36

A GRAPHICAL MODELS

Figure 3 illustrates the graphical models for the different variants of the TPRT discussed in this work. The models are presented in both their conventional form (left column) and their equivalent augmented, scale-mixture form (right column), which is the basis for our tractable inference framework.

The top row shows the non-sparse models, while the bottom row introduces inducing variables $\mathbf{u}_m$ for the sparse approximation. In the augmented models (b) and (d), the degrees of freedom parameters (ν_f, ν_ε) are replaced by Gamma-distributed auxiliary variables $(r, \boldsymbol{\lambda}_n)$, transforming the Student-t distributions into conditionally Gaussian ones.



(a) Conventional TPRT model

(b) Augmented TPRT model

(c) Conventional Sparse TPRT model

(d) Augmented Sparse TPRT model

Figure 3: Graphical models for the TPRT variants. **(Top Row)**: Non-sparse models. **(Bottom Row)**: Sparse models with inducing variables u_m. **(Left Column)**: Conventional representation with direct Student-t distributions. **(Right Column)**: Equivalent augmented representation using the scale-mixture of Gaussians with auxiliary variables r and λ_n. This augmented form is used for our proposed inference method.

B DETAILED DERIVATIONS FOR TPRT

This appendix provides the detailed derivations for the TPRT model.

B.1 Moments of the Latent Function Value under the Variational Posterior

We begin by deriving the first two moments of the latent function values, f_i, under the variational posterior. The full variational posterior has the form $q(\mathbf{f}, r, \boldsymbol{\lambda}) = q(\mathbf{f}, r) \prod_{i=1}^{N} q(\lambda_i)$. As the moments of f_i only depend on the structured component $q(\mathbf{f}, r)$, we define it as a Gaussian-Gamma distribution:

$$q(\mathbf{f}, r) = \mathcal{N}(\mathbf{f}; \mathbf{m}, r^{-1}\mathbf{S})\mathrm{Gamma}(r; \alpha_r, \beta_r).$$

Expectation $\mathbb{E}_q[f_i]$. The expectation of the i-th latent function value is taken with respect to $q(\mathbf{f}, r)$ and is simply the i-th element of the mean vector:

$$\mathbb{E}_q[f_i] = \mathbb{E}_{q(\mathbf{f},r)}[f_i] = m_i.$$

Variance $\mathrm{Var}_q(f_i)$. We use the law of total variance. As the conditional expectation $\mathbb{E}_{q(\mathbf{f}|r)}[f_i] = m_i$ is not a function of r, the variance of the expectation is zero. The variance is thus given by the expectation of the conditional variance:

$$\begin{aligned}\mathrm{Var}_q(f_i) &= \mathbb{E}_{q(r)}[\mathrm{Var}_{q(\mathbf{f}|r)}(f_i)] + \mathrm{Var}_{q(r)}[\mathbb{E}_{q(\mathbf{f}|r)}(f_i)] \\ &= \mathbb{E}_{q(r)}[r^{-1}S_{ii}] + \mathrm{Var}_{q(r)}[m_i] \\ &= S_{ii} \cdot \mathbb{E}_{q(r)}[r^{-1}],\end{aligned}$$

where $\mathbb{E}_{q(r)}[r^{-1}] = \beta_r/(\alpha_r - 1)$ for $\alpha_r > 1$.

B.2 ELBO

The ELBO for the non-sparse model is $\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{y}|\mathbf{f}, \boldsymbol{\lambda})] - \mathrm{KL}(q(\mathbf{f}, r)||p(\mathbf{f}, r)) - \sum_{i=1}^{N} \mathrm{KL}(q(\lambda_i)||p(\lambda_i))$.

The first term, the expected log-likelihood, is computed using the moments above:

$$\begin{aligned}\mathbb{E}_q[\log p(\mathbf{y}|\mathbf{f}, \boldsymbol{\lambda})] &= \sum_{i=1}^{N} \mathbb{E}_q\left[-\frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2}\log\lambda_i - \frac{\lambda_i}{2\sigma^2}(y_i - f_i)^2\right] \\ &= \sum_{i=1}^{N}\left(-\frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2}\mathbb{E}_{q(\lambda_i)}[\log\lambda_i] - \frac{\mathbb{E}_{q(\lambda_i)}[\lambda_i]}{2\sigma^2}\mathbb{E}_{q(\mathbf{f},r)}[(y_i - f_i)^2]\right) \\ &= \sum_{i=1}^{N}\left(-\frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2}(\psi(\alpha_{\lambda_i}) - \log\beta_{\lambda_i}) - \frac{\alpha_{\lambda_i}}{2\sigma^2\beta_{\lambda_i}}\left((y_i - m_i)^2 + S_{ii}\mathbb{E}_{q(r)}[r^{-1}]\right)\right).\end{aligned}$$

The KL divergence $\mathrm{KL}(q(\mathbf{f}, r)||p(\mathbf{f}, r))$ between the variational posterior and the prior is that between two Gaussian-Gamma distributions. The remaining KL terms are between Gamma distributions.

B.3 CAVI Updates

We derive the coordinate ascent updates. The update for $q(\lambda_i)$ is conjugate, while the update for $q(\mathbf{f}, r)$ requires our two-step approximation.

Update for $q(\lambda_i)$. The optimal distribution $q^*(\lambda_i)$ is conditionally conjugate. Its form is found by isolating the terms dependent on λ_i in the expected log-joint probability:

$$\begin{aligned}\log q^*(\lambda_i) &\propto \mathbb{E}_{q(\mathbf{f},r)}\left[\log p(y_i|f_i, \lambda_i)\right] + \log p(\lambda_i) \\ &\propto \left(\frac{\nu_\varepsilon + 1}{2} - 1\right)\log\lambda_i - \left(\frac{\nu_\varepsilon}{2} + \frac{\mathbb{E}_{q(\mathbf{f},r)}[(y_i - f_i)^2]}{2\sigma^2}\right)\lambda_i.\end{aligned}$$

From this form, we can recognize that $q^*(\lambda_i)$ is a Gamma distribution. By matching its log-probability density, $\log\mathrm{Gamma}(\lambda_i; \alpha, \beta) \propto (\alpha - 1)\log\lambda_i - \beta\lambda_i$, we find the updated parameters:

$$\begin{aligned}\alpha_{\lambda_i} &\leftarrow \frac{\nu_\varepsilon + 1}{2}, \\ \beta_{\lambda_i} &\leftarrow \frac{\nu_\varepsilon}{2} + \frac{1}{2\sigma^2}\left((y_i - m_i)^2 + S_{ii}\mathbb{E}_{q(r)}[r^{-1}]\right).\end{aligned}$$

Approximate Update for $q(\mathbf{f}, r)$. As a direct CAVI update for the structured posterior $q(\mathbf{f}, r)$ is not available in closed form, we employ a two-step procedure. First, we find the optimal parameters for a temporary, fully factorized proxy distribution, $\tilde{q}(\mathbf{f}, r) = \tilde{q}(\mathbf{f})\tilde{q}(r) = \mathcal{N}(\mathbf{f}; \tilde{\mathbf{m}}, \tilde{\mathbf{S}})\text{Gamma}(r; \tilde{\alpha}_r, \tilde{\beta}_r)$.

The optimal Gaussian factor $\tilde{q}^*(\mathbf{f})$ is found by taking the expectation with respect to the other variational distributions:

$$\begin{aligned}
\log \tilde{q}^*(\mathbf{f}) &\propto \mathbb{E}_{q(\boldsymbol{\lambda})}\left[\log p(\mathbf{y}|\mathbf{f}, \boldsymbol{\lambda})\right] + \mathbb{E}_{\tilde{q}(r)}\left[\log p(\mathbf{f}|r)\right] \\
&\propto \mathbb{E}_{q(\boldsymbol{\lambda})}\left[\sum_{i=1}^{N}\left(-\frac{\lambda_i}{2\sigma^2}(y_i - f_i)^2\right)\right] + \mathbb{E}_{\tilde{q}(r)}\left[-\frac{r}{2}\mathbf{f}^\top \mathbf{K}_{XX}^{-1}\mathbf{f}\right] \\
&\propto -\frac{1}{2}\mathbf{f}^\top\left(\mathbb{E}_{\tilde{q}(r)}[r]\mathbf{K}_{XX}^{-1} + \frac{1}{\sigma^2}\text{diag}(\mathbb{E}_{q(\boldsymbol{\lambda})}[\boldsymbol{\lambda}])\right)\mathbf{f} + \mathbf{f}^\top\left(\frac{1}{\sigma^2}\text{diag}(\mathbb{E}_{q(\boldsymbol{\lambda})}[\boldsymbol{\lambda}])\mathbf{y}\right).
\end{aligned}$$

This expression is a quadratic function of $\mathbf{f}$, which corresponds to the log-density of a Gaussian distribution. By completing the square, we can identify the parameters and obtain the update rules for the proxy distribution:

$$\begin{aligned}
\tilde{\mathbf{S}} &\leftarrow \left(\mathbb{E}_{\tilde{q}(r)}[r]\mathbf{K}_{XX}^{-1} + \frac{1}{\sigma^2}\text{diag}(\mathbb{E}_{q(\boldsymbol{\lambda})}[\boldsymbol{\lambda}])\right)^{-1}, \\
\tilde{\mathbf{m}} &\leftarrow \frac{1}{\sigma^2}\tilde{\mathbf{S}}\,\text{diag}(\mathbb{E}_{q(\boldsymbol{\lambda})}[\boldsymbol{\lambda}])\,\mathbf{y}.
\end{aligned}$$

Similarly, the optimal Gamma factor $\tilde{q}^*(r)$ is found from:

$$\begin{aligned}
\log \tilde{q}^*(r) &\propto \mathbb{E}_{\tilde{q}(\mathbf{f})}[\log p(\mathbf{f}|r)] + \log p(r) \\
&\propto \mathbb{E}_{\tilde{q}(\mathbf{f})}\left[\frac{N}{2}\log r - \frac{r}{2}\mathbf{f}^\top\mathbf{K}_{XX}^{-1}\mathbf{f}\right] + \left(\frac{\nu_f}{2} - 1\right)\log r - \frac{\nu_f}{2}r \\
&\propto \left(\frac{N + \nu_f}{2} - 1\right)\log r - \left(\frac{\nu_f}{2} + \frac{1}{2}\mathbb{E}_{\tilde{q}(\mathbf{f})}[\mathbf{f}^\top\mathbf{K}_{XX}^{-1}\mathbf{f}]\right)r.
\end{aligned}$$

From this form, we recognize that $\tilde{q}^*(r)$ is a Gamma distribution. By comparing coefficients with the log-density of a Gamma distribution, we get the updates:

$$\begin{aligned}
\tilde{\alpha}_r &\leftarrow \frac{N + \nu_f}{2}, \\
\tilde{\beta}_r &\leftarrow \frac{\nu_f}{2} + \frac{1}{2}\left(\tilde{\mathbf{m}}^\top\mathbf{K}_{XX}^{-1}\tilde{\mathbf{m}} + \text{Tr}(\mathbf{K}_{XX}^{-1}\tilde{\mathbf{S}})\right).
\end{aligned}$$

Finally, we project the optimized proxy distribution $\tilde{q}(\mathbf{f}, r)$ back to our structured family $q(\mathbf{f}, r)$ using the solution from Appendix E. This yields the final updates for the structured posterior's parameters:

$$\begin{aligned}
\mathbf{m} &\leftarrow \tilde{\mathbf{m}}, \\
\mathbf{S} &\leftarrow \mathbb{E}_{\tilde{q}(r)}[r] \cdot \tilde{\mathbf{S}} = \frac{\tilde{\alpha}_r}{\tilde{\beta}_r}\tilde{\mathbf{S}}, \\
\alpha_r &\leftarrow \tilde{\alpha}_r, \\
\beta_r &\leftarrow \tilde{\beta}_r,
\end{aligned}$$

This completes the two-step approximate CAVI update for $q(\mathbf{f}, r)$.

B.4 Predictive Distribution

A key benefit of our structured posterior is that the predictive distribution for the latent function value f_* at a new test point $\mathbf{x}_*$ can be computed analytically. The approximation is the marginalization of the latent variables with respect to our variational posterior:

$$q(f_*) = \int p(f_*|\mathbf{f}, r)q(\mathbf{f}, r)d\mathbf{f}dr = \int\left[\int p(f_*|\mathbf{f}, r)q(\mathbf{f}|r)d\mathbf{f}\right]q(r)dr. \tag{4}$$

The inner integral over $\mathbf{f}$ involves the product of two Gaussians: the conditional prior $p(f_*|\mathbf{f}, r) = \mathcal{N}(f_*; \mathbf{k}_{*X}\mathbf{K}_{XX}^{-1}\mathbf{f}, r^{-1}k'_{**})$ and the variational posterior $q(\mathbf{f}|r) = \mathcal{N}(\mathbf{f}; \mathbf{m}, r^{-1}\mathbf{S})$. The result of this marginalization is another Gaussian, $q(f_*|r) = \mathcal{N}(f_*; \mu_{f_*}, \sigma^2_{f_*})$, with parameters:

$$\mu_{f_*} = \mathbb{E}_{q(\mathbf{f}|r)}[\mathbf{k}_{*X}\mathbf{K}_{XX}^{-1}\mathbf{f}] = \mathbf{k}_{*X}\mathbf{K}_{XX}^{-1}\mathbf{m}, \tag{5}$$

$$\sigma^2_{f_*} = \mathbb{E}_{q(\mathbf{f}|r)}[r^{-1}k'_{**}] + \mathrm{Var}_{q(\mathbf{f}|r)}[\mathbf{k}_{*X}\mathbf{K}_{XX}^{-1}\mathbf{f}] = r^{-1}\left(k'_{**} + \mathbf{k}_{*X}\mathbf{K}_{XX}^{-1}\mathbf{S}\mathbf{K}_{XX}^{-1}\mathbf{k}_{X*}\right). \tag{6}$$

Here, $\mathbf{k}_{*X} = k(\mathbf{x}_*, \mathbf{X})$ and $k'_{**} = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_{*X}\mathbf{K}_{XX}^{-1}\mathbf{k}_{X*}$.

The outer integral is then a Gaussian-Gamma mixture:

$$q(f_*) = \int \mathcal{N}(f_*; \mu_{f_*}, \sigma^2_{f_*})\mathrm{Gamma}(r; \alpha_r, \beta_r)dr. \tag{7}$$

This integral is the definition of a univariate Student-t distribution, $q(f_*) = \mathcal{ST}(f_*; \nu_*, \mu_*, \sigma^2_*)$, with the following parameters:

$$\nu_* = 2\alpha_r, \tag{8}$$

$$\mu_* = \mathbf{k}_{*X}\mathbf{K}_{XX}^{-1}\mathbf{m}, \tag{9}$$

$$\sigma^2_* = \frac{\beta_r}{\alpha_r}\left(k'_{**} + \mathbf{k}_{*X}\mathbf{K}_{XX}^{-1}\mathbf{S}\mathbf{K}_{XX}^{-1}\mathbf{k}_{X*}\right). \tag{10}$$

Note that while the predictive distribution for the latent function f_* is analytic, the predictive distribution for a new observation y_* involves a convolution of two Student-t distributions, which requires a further approximation.

C DETAILED DERIVATIONS FOR SPARSE TPRT

This appendix provides the detailed derivations for the sparse TPRT and its stochastic optimization.

C.1 Moments of the Latent Function Value under the Variational Posterior

The expected log-likelihood term in the ELBO requires the first two moments of the latent function values, f_i, under the variational distribution q. The structure of our variational distribution is $q(\mathbf{f}, \mathbf{u}, r, \boldsymbol{\lambda}) = p(\mathbf{f}|\mathbf{u}, r)q(\mathbf{u}, r)\prod_i q(\lambda_i)$, where the structured component is $q(\mathbf{u}, r) = \mathcal{N}(\mathbf{u}; \mathbf{m}, r^{-1}\mathbf{S})\mathrm{Gamma}(r; \alpha_r, \beta_r)$.

We use the shorthand notation $\mathbf{A} \coloneqq \mathbf{K}_{XZ}\mathbf{K}_{ZZ}^{-1}$ and $\tilde{\mathbf{K}} \coloneqq \mathbf{K}_{XX} - \mathbf{A}\mathbf{K}_{ZX}$. The i-th row of $\mathbf{A}$ is $\mathbf{a}_i^\top$, and the i-th diagonal element of $\tilde{\mathbf{K}}$ is $\tilde{K}_{ii}$. The conditional distribution of f_i from the GP prior is $p(f_i|\mathbf{u}, r) = \mathcal{N}(f_i; \mathbf{a}_i^\top\mathbf{u}, r^{-1}\tilde{K}_{ii})$.

Expectation $\mathbb{E}_q[f_i]$. Using the law of total expectation, the mean of f_i under the full variational distribution q is:

$$\mathbb{E}_q[f_i] = \mathbb{E}_{q(\mathbf{u},r)}\left[\mathbb{E}_{p(f_i|\mathbf{u},r)}[f_i]\right] = \mathbb{E}_{q(\mathbf{u},r)}[\mathbf{a}_i^\top\mathbf{u}] = \mathbf{a}_i^\top\mathbf{m}.$$

The mean depends only on the variational mean of the inducing variables.

Variance $\mathrm{Var}_q(f_i)$. Using the law of total variance, we have:

$$\begin{aligned}\mathrm{Var}_q(f_i) &= \mathbb{E}_{q(\mathbf{u},r)}[\mathrm{Var}_{p(f_i|\mathbf{u},r)}(f_i)] + \mathrm{Var}_{q(\mathbf{u},r)}[\mathbb{E}_{p(f_i|\mathbf{u},r)}(f_i)] \\ &= \mathbb{E}_{q(\mathbf{u},r)}[r^{-1}\tilde{K}_{ii}] + \mathrm{Var}_{q(\mathbf{u},r)}[\mathbf{a}_i^\top\mathbf{u}].\end{aligned}$$

The first term is $\tilde{K}_{ii}\mathbb{E}_{q(r)}[r^{-1}]$. The second term can be expanded again:

$$\begin{aligned}\mathrm{Var}_{q(\mathbf{u},r)}(\mathbf{a}_i^\top\mathbf{u}) &= \mathbb{E}_{q(r)}[\mathrm{Var}_{q(\mathbf{u}|r)}(\mathbf{a}_i^\top\mathbf{u})] + \mathrm{Var}_{q(r)}[\mathbb{E}_{q(\mathbf{u}|r)}(\mathbf{a}_i^\top\mathbf{u})] \\ &= \mathbb{E}_{q(r)}\left[\mathbf{a}_i^\top(r^{-1}\mathbf{S})\mathbf{a}_i\right] + \mathrm{Var}_{q(r)}[\mathbf{a}_i^\top\mathbf{m}] = (\mathbf{a}_i^\top\mathbf{S}\mathbf{a}_i)\mathbb{E}_{q(r)}[r^{-1}].\end{aligned}$$

Combining the components gives the total variance:

$$\mathrm{Var}_q(f_i) = \left(\tilde{K}_{ii} + \mathbf{a}_i^\top\mathbf{S}\mathbf{a}_i\right)\mathbb{E}_{q(r)}[r^{-1}],$$

where $\mathbb{E}_{q(r)}[r^{-1}] = \beta_r/(\alpha_r - 1)$ for $\alpha_r > 1$.

C.2 ELBO

Given our variational family where $q(\mathbf{f}|\mathbf{u}, r) = p(\mathbf{f}|\mathbf{u}, r)$, the ELBO simplifies to:

$$\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{y}|\mathbf{f}, \boldsymbol{\lambda})] - \mathrm{KL}(q(\mathbf{u}, r)||p(\mathbf{u}, r)) - \sum_{i=1}^{N} \mathrm{KL}(q(\lambda_i)||p(\lambda_i)).$$

The expected log-likelihood term is computed using the moments derived above:

$$\begin{aligned}
\mathbb{E}_q[\log p(\mathbf{y}|\mathbf{f}, \boldsymbol{\lambda})] &= \sum_{i=1}^{N} \mathbb{E}_q \left[-\frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2}\log\lambda_i - \frac{\lambda_i}{2\sigma^2}(y_i - f_i)^2 \right] \\
&= \sum_{i=1}^{N} \left(-\frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2}\mathbb{E}_{q(\lambda_i)}[\log\lambda_i] - \frac{\mathbb{E}_{q(\lambda_i)}[\lambda_i]}{2\sigma^2} \left((y_i - \mathbb{E}_q[f_i])^2 + \mathrm{Var}_q(f_i) \right) \right) \\
&= \sum_{i=1}^{N} \left(-\frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2}(\psi(\alpha_{\lambda_i}) - \log\beta_{\lambda_i}) - \frac{\alpha_{\lambda_i}}{2\sigma^2\beta_{\lambda_i}} \left((y_i - \mathbb{E}_q[f_i])^2 + \mathrm{Var}_q(f_i) \right) \right).
\end{aligned}$$

C.3 CAVI Updates

We derive the coordinate ascent updates. As in the non-sparse case, the update for $q(\mathbf{u}, r)$ requires a two-step approximation.

Update for $q(\lambda_i)$. The update for $q(\lambda_i)$ is conditionally conjugate and identical in form to the non-sparse model, using the sparse model's moments for f_i:

$$\begin{aligned}
\log q^*(\lambda_i) &\propto \mathbb{E}_q\left[\log p(y_i|f_i, \lambda_i)\right] + \log p(\lambda_i) \\
&\propto \left(\frac{\nu_\varepsilon + 1}{2} - 1 \right) \log\lambda_i - \left(\frac{\nu_\varepsilon}{2} + \frac{\mathbb{E}_q[(y_i - f_i)^2]}{2\sigma^2} \right) \lambda_i.
\end{aligned}$$

This corresponds to a Gamma distribution, yielding the updates:

$$\begin{aligned}
\alpha_{\lambda_i} &\leftarrow \frac{\nu_\varepsilon + 1}{2}, \\
\beta_{\lambda_i} &\leftarrow \frac{\nu_\varepsilon}{2} + \frac{1}{2\sigma^2}\left((y_i - \mathbb{E}_q[f_i])^2 + \mathrm{Var}_q(f_i) \right).
\end{aligned}$$

Approximate Update for $q(\mathbf{u}, r)$. We first compute the updates for a fully factorized proxy distribution, $\tilde{q}(\mathbf{u}, r) = \tilde{q}(\mathbf{u})\tilde{q}(r) = \mathcal{N}(\mathbf{u}; \tilde{\mathbf{m}}, \tilde{\mathbf{S}})\mathrm{Gamma}(r; \tilde{\alpha}_r, \tilde{\beta}_r)$.

The optimal factor $\tilde{q}^*(\mathbf{u})$ is found from terms involving $\mathbf{u}$:

$$\begin{aligned}
\log\tilde{q}^*(\mathbf{u}) &\propto \mathbb{E}_{\tilde{q}(r)}[\log p(\mathbf{u}|r)] + \sum_{i=1}^{N} \mathbb{E}_q[\log p(y_i|f_i, \lambda_i)] \\
&\propto -\frac{\mathbb{E}_{\tilde{q}(r)}[r]}{2}\mathbf{u}^\top \mathbf{K}_{ZZ}^{-1}\mathbf{u} + \sum_{i=1}^{N} \mathbb{E}_{q(\lambda_i)} \left[-\frac{\lambda_i}{2\sigma^2}(y_i - \mathbf{a}_i^\top \mathbf{u})^2 \right] \\
&\propto -\frac{1}{2}\mathbf{u}^\top \left(\mathbb{E}_{\tilde{q}(r)}[r]\mathbf{K}_{ZZ}^{-1} + \frac{1}{\sigma^2}\mathbf{A}^\top \mathrm{diag}(\mathbb{E}_q[\boldsymbol{\lambda}])\mathbf{A} \right) \mathbf{u} + \mathbf{u}^\top \left(\frac{1}{\sigma^2}\mathbf{A}^\top \mathrm{diag}(\mathbb{E}_q[\boldsymbol{\lambda}])\mathbf{y} \right).
\end{aligned}$$

This quadratic form in $\mathbf{u}$ corresponds to a Gaussian distribution, giving the proxy updates:

$$\begin{aligned}
\tilde{\mathbf{S}}^{-1} &\leftarrow \mathbb{E}_{\tilde{q}(r)}[r]\mathbf{K}_{ZZ}^{-1} + \frac{1}{\sigma^2}\mathbf{A}^\top \mathrm{diag}(\mathbb{E}_q[\boldsymbol{\lambda}])\mathbf{A}, \\
\tilde{\mathbf{m}} &\leftarrow \tilde{\mathbf{S}} \left(\frac{1}{\sigma^2}\mathbf{A}^\top \mathrm{diag}(\mathbb{E}_q[\boldsymbol{\lambda}])\mathbf{y} \right).
\end{aligned}$$

The optimal factor $\tilde{q}^*(r)$ is found from terms involving r:

$$\begin{aligned}\log \tilde{q}^*(r) &\propto \mathbb{E}_{\tilde{q}(\mathbf{u})}[\log p(\mathbf{u}|r)] + \log p(r) \\ &\propto \mathbb{E}_{\tilde{q}(\mathbf{u})}\left[\frac{M}{2}\log r - \frac{r}{2}\mathbf{u}^\top \mathbf{K}_{ZZ}^{-1}\mathbf{u}\right] + \left(\frac{\nu_f}{2} - 1\right)\log r - \frac{\nu_f}{2} r \\ &\propto \left(\frac{\nu_f + M}{2} - 1\right)\log r - \left(\frac{\nu_f}{2} + \frac{1}{2}\mathbb{E}_{\tilde{q}(\mathbf{u})}[\mathbf{u}^\top \mathbf{K}_{ZZ}^{-1}\mathbf{u}]\right) r.\end{aligned}$$

This corresponds to a Gamma distribution, yielding the proxy updates:

$$\begin{aligned}\tilde{\alpha}_r &\leftarrow \frac{\nu_f + M}{2}, \\ \tilde{\beta}_r &\leftarrow \frac{\nu_f}{2} + \frac{1}{2}\left(\mathrm{Tr}(\mathbf{K}_{ZZ}^{-1}\tilde{\mathbf{S}}) + \tilde{\mathbf{m}}^\top \mathbf{K}_{ZZ}^{-1}\tilde{\mathbf{m}}\right).\end{aligned}$$

Finally, we project the optimized proxy $\tilde{q}(\mathbf{u}, r)$ back to our structured family $q(\mathbf{u}, r)$ using the solution from Appendix E. The resulting updates are:

$$\begin{aligned}\mathbf{m} &\leftarrow \tilde{\mathbf{m}}, \\ \alpha_r &\leftarrow \tilde{\alpha}_r, \\ \beta_r &\leftarrow \tilde{\beta}_r, \\ \mathbf{S} &\leftarrow \mathbb{E}_{\tilde{q}(r)}[r] \cdot \tilde{\mathbf{S}} = \frac{\tilde{\alpha}_r}{\tilde{\beta}_r}\tilde{\mathbf{S}}.\end{aligned}$$

C.4 Stochastic Optimization

To scale our model to large datasets, we employ stochastic optimization. The ELBO is approximated on a mini-batch of size B by scaling the sum over data-dependent terms by N/B. This provides an unbiased estimator for computing gradients in the M-step.

The E-step is also adapted for this stochastic setting. The local variational parameters $\{q(\lambda_i)\}_{i \in \mathbf{B}}$ are updated using only the current mini-batch, following the standard CAVI update.

For the global variational parameters of $q(\mathbf{u}, r)$, we perform an approximate natural gradient step. First, for each mini-batch, we compute the parameters for the fully factorized proxy distribution $\tilde{q}(\mathbf{u}, r)$, scaling the data-dependent terms as follows:

$$\begin{aligned}\tilde{\mathbf{S}}^{-1} &\leftarrow \mathbb{E}_{\tilde{q}(r)}[r]\mathbf{K}_{ZZ}^{-1} + \frac{N}{B}\frac{1}{\sigma^2}\mathbf{A_B}^\top \mathrm{diag}(\mathbb{E}_q[\boldsymbol{\lambda}_{\boldsymbol{B}}])\mathbf{A_B}, \\ \tilde{\mathbf{m}} &\leftarrow \tilde{\mathbf{S}}\left(\frac{N}{B}\frac{1}{\sigma^2}\mathbf{A_B}^\top \mathrm{diag}(\mathbb{E}_q[\boldsymbol{\lambda}_{\boldsymbol{B}}])\mathbf{y_B}\right), \\ \tilde{\alpha}_r &\leftarrow \frac{\nu_f + M}{2}, \\ \tilde{\beta}_r &\leftarrow \frac{\nu_f}{2} + \frac{1}{2}\left(\mathrm{Tr}(\mathbf{K}_{ZZ}^{-1}\tilde{\mathbf{S}}) + \tilde{\mathbf{m}}^\top \mathbf{K}_{ZZ}^{-1}\tilde{\mathbf{m}}\right).\end{aligned}$$

These proxy parameters are then projected to obtain the target parameters for our structured posterior. Both the current global parameters and the target parameters are converted to their natural parameter representation, $\boldsymbol{\eta}_{\text{global}}^{(t)}$ and $\boldsymbol{\eta}_{\text{target}}$. The global parameters are then updated using Polyak averaging with a learning rate ρ_t:

$$\boldsymbol{\eta}_{\text{global}}^{(t+1)} \leftarrow (1 - \rho_t)\boldsymbol{\eta}_{\text{global}}^{(t)} + \rho_t \boldsymbol{\eta}_{\text{target}}.$$

C.5 Predictive Distribution

The predictive distribution for a new latent function value f_* at a test point $\mathbf{x}_*$ is also analytic in the sparse model. The approximation involves marginalizing the inducing variables $\mathbf{u}$ and the global precision r with respect to their variational posterior $q(\mathbf{u}, r)$:

$$q(f_*) = \int p(f_*|\mathbf{u}, r)q(\mathbf{u}, r)d\mathbf{u}dr = \int \left[\int p(f_*|\mathbf{u}, r)q(\mathbf{u}|r)d\mathbf{u}\right] q(r)dr. \tag{11}$$

The inner integral over $\mathbf{u}$ is a product of two Gaussians: the conditional prior from the sparse GP, $p(f_*|\mathbf{u}, r) = \mathcal{N}(f_*; \mathbf{k}_{*Z}\mathbf{K}_{ZZ}^{-1}\mathbf{u}, r^{-1}k'_{**})$, and the variational posterior $q(\mathbf{u}|r) = \mathcal{N}(\mathbf{u}; \mathbf{m}, r^{-1}\mathbf{S})$. The result is another Gaussian, $q(f_*|r) = \mathcal{N}(f_*; \mu_{f_*}, \sigma^2_{f_*})$, with parameters:

$$\mu_{f_*} = \mathbb{E}_{q(\mathbf{u}|r)}[\mathbf{k}_{*Z}\mathbf{K}_{ZZ}^{-1}\mathbf{u}] = \mathbf{k}_{*Z}\mathbf{K}_{ZZ}^{-1}\mathbf{m}, \tag{12}$$

$$\sigma^2_{f_*} = \mathbb{E}_{q(\mathbf{u}|r)}[r^{-1}k'_{**}] + \mathrm{Var}_{q(\mathbf{u}|r)}[\mathbf{k}_{*Z}\mathbf{K}_{ZZ}^{-1}\mathbf{u}] = r^{-1}\left(k'_{**} + \mathbf{k}_{*Z}\mathbf{K}_{ZZ}^{-1}\mathbf{S}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z*}\right). \tag{13}$$

Here, $\mathbf{k}_{*Z} = k(\mathbf{x}_*, \mathbf{Z})$ and $k'_{**} = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_{*Z}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z*}$.

The outer integral becomes a Gaussian-Gamma mixture:

$$q(f_*) = \int \mathcal{N}(f_*|\mu_{f_*}, \sigma^2_{f_*})\mathrm{Gamma}(r|\alpha_r, \beta_r)dr. \tag{14}$$

This integral defines a univariate Student-t distribution, $q(f_*) = \mathcal{ST}(f_*; \nu_*, \mu_*, \sigma^2_*)$, with the following parameters:

$$\nu_* = 2\alpha_r, \tag{15}$$

$$\mu_* = \mathbf{k}_{*Z}\mathbf{K}_{ZZ}^{-1}\mathbf{m}, \tag{16}$$

$$\sigma^2_* = \frac{\beta_r}{\alpha_r}\left(k'_{**} + \mathbf{k}_{*Z}\mathbf{K}_{ZZ}^{-1}\mathbf{S}\mathbf{K}_{ZZ}^{-1}\mathbf{k}_{Z*}\right). \tag{17}$$

As in the non-sparse case, predicting a new observation y_* requires a further approximation for the convolution of two Student-t distributions.

D EXPONENTIAL FAMILY REPRESENTATION OF THE GAUSSIAN-GAMMA DISTRIBUTION

The Gaussian-Gamma distribution, a member of the exponential family, is central to our natural gradient updates. Its probability density function $p(\mathbf{u}, r|\mathbf{m}, \mathbf{S}, \alpha, \beta) = \mathcal{N}(\mathbf{u}; \mathbf{m}, r^{-1}\mathbf{S})\mathrm{Gamma}(r; \alpha, \beta)$ can be written in the canonical form $p(\mathbf{z}|\boldsymbol{\eta}) = h(\mathbf{z})\exp(\boldsymbol{\eta}^\top T(\mathbf{z}) - A(\boldsymbol{\eta}))$. The log-PDF is:

$$\log p(\mathbf{u}, r) = \mathrm{const} + \left(\alpha + \frac{M}{2} - 1\right)\log r - \beta r - \frac{r}{2}(\mathbf{u} - \mathbf{m})^\top\mathbf{S}^{-1}(\mathbf{u} - \mathbf{m}).$$

Expanding this reveals the natural parameters $\boldsymbol{\eta}$ corresponding to the sufficient statistics $T(\mathbf{u}, r) = (r\mathbf{u}, r\mathbf{u}\mathbf{u}^\top, r, \log r)^\top$:

$$\begin{aligned}
\boldsymbol{\eta}_1 &= \mathbf{S}^{-1}\mathbf{m}, \\
\boldsymbol{\eta}_2 &= -\frac{1}{2}\mathbf{S}^{-1}, \\
\eta_3 &= -\beta - \frac{1}{2}\mathbf{m}^\top\mathbf{S}^{-1}\mathbf{m}, \\
\eta_4 &= \alpha + \frac{M}{2} - 1.
\end{aligned}$$

The inverse transformation from natural parameters back to the standard parameters $(\mathbf{m}, \mathbf{S}, \alpha, \beta)$ is given by:

$$\begin{aligned}
\mathbf{S} &= (-2\boldsymbol{\eta}_2)^{-1}, \\
\mathbf{m} &= (-2\boldsymbol{\eta}_2)^{-1}\boldsymbol{\eta}_1, \\
\alpha &= \eta_4 + 1 - \frac{M}{2}, \\
\beta &= -\eta_3 + \frac{1}{4}\boldsymbol{\eta}_1^\top\boldsymbol{\eta}_2^{-1}\boldsymbol{\eta}_1.
\end{aligned}$$

E PROJECTION FROM FACTORIZED TO STRUCTURED DISTRIBUTION

This section details the projection of a fully factorized Normal-Gamma distribution onto the structured Gaussian-Gamma family by minimizing the Kullback-Leibler (KL) divergence. This is the core of our closed-form approximate update for the structured variational posterior $q(\mathbf{u}, r)$.

Let the target distribution be the factorized proxy $\tilde{q}(\mathbf{u}, r)$ with known parameters:

$$\tilde{q}(\mathbf{u}, r) = \mathcal{N}(\mathbf{u}; \tilde{\mathbf{m}}, \tilde{\mathbf{S}})\text{Gamma}(r; \tilde{\alpha}_r, \tilde{\beta}_r).$$

Our goal is to find the best approximation within the structured family $q(\mathbf{u}, r)$:

$$q(\mathbf{u}, r; \mathbf{m}, \mathbf{S}, \alpha_r, \beta_r) = \mathcal{N}(\mathbf{u}; \mathbf{m}, r^{-1}\mathbf{S})\text{Gamma}(r; \alpha_r, \beta_r).$$

We find the optimal parameters $(\mathbf{m}, \mathbf{S}, \alpha_r, \beta_r)$ by minimizing $\text{KL}(\tilde{q}||q)$, which is equivalent to maximizing the expectation $\mathbb{E}_{\tilde{q}}[\log q(\mathbf{u}, r)]$.

The objective function separates into two independent optimization problems:

$$\underset{\mathbf{m},\mathbf{S},\alpha_r,\beta_r}{\arg\max}\ \mathbb{E}_{\tilde{q}}[\log q(\mathbf{u}, r)] = \underset{\mathbf{m},\mathbf{S}}{\arg\max}\, \mathbb{E}_{\tilde{q}}[\log \mathcal{N}(\mathbf{u}; \mathbf{m}, r^{-1}\mathbf{S})] + \underset{\alpha_r,\beta_r}{\arg\max}\, \mathbb{E}_{\tilde{q}}[\log \text{Gamma}(r; \alpha_r, \beta_r)].$$

The second term is maximized when the parameters of the Gamma distributions are identical. Thus, the optimal Gamma parameters are:

$$\alpha_r = \tilde{\alpha}_r,$$
$$\beta_r = \tilde{\beta}_r.$$

For the Normal parameters, we maximize the objective $\mathcal{L}(\mathbf{m}, \mathbf{S}) = \mathbb{E}_{\tilde{q}}[\log \mathcal{N}(\mathbf{u}; \mathbf{m}, r^{-1}\mathbf{S})]$. Expanding the log-pdf and taking the expectation with respect to $\tilde{q}(\mathbf{u}, r) = \tilde{q}(\mathbf{u})\tilde{q}(r)$ yields:

$$\begin{aligned}\mathcal{L}(\mathbf{m}, \mathbf{S}) &= \mathbb{E}_{\tilde{q}}\left[\text{const} - \frac{1}{2}\log|\mathbf{S}| + \frac{M}{2}\log r - \frac{r}{2}(\mathbf{u}-\mathbf{m})^\top \mathbf{S}^{-1}(\mathbf{u}-\mathbf{m})\right] \\ &= \text{const} - \frac{1}{2}\log|\mathbf{S}| - \frac{1}{2}\mathbb{E}_{\tilde{q}(r)}[r] \cdot \mathbb{E}_{\tilde{q}(\mathbf{u})}\left[(\mathbf{u}-\mathbf{m})^\top \mathbf{S}^{-1}(\mathbf{u}-\mathbf{m})\right]. \end{aligned} \tag{18}$$

This expression is maximized with respect to $\mathbf{m}$ when it is set to the mean of $\mathbf{u}$ under $\tilde{q}(\mathbf{u})$:

$$\mathbf{m} = \mathbb{E}_{\tilde{q}(\mathbf{u})}[\mathbf{u}] = \tilde{\mathbf{m}}.$$

Substituting $\mathbf{m} = \tilde{\mathbf{m}}$ into Eq. (18) and using the trace identity $\mathbb{E}[(\mathbf{x}-\boldsymbol{\mu})^\top \mathbf{A}(\mathbf{x}-\boldsymbol{\mu})] = \text{Tr}(\mathbf{A}\text{Cov}(\mathbf{x}))$, the objective becomes:

$$\mathcal{L}(\mathbf{S}) = \text{const} - \frac{1}{2}\log|\mathbf{S}| - \frac{1}{2}\mathbb{E}_{\tilde{q}(r)}[r] \cdot \text{Tr}(\mathbf{S}^{-1}\tilde{\mathbf{S}}).$$

Taking the derivative with respect to $\mathbf{S}^{-1}$ and setting it to zero gives the optimal $\mathbf{S}$:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{S}^{-1}} = \frac{1}{2}\mathbf{S} - \frac{1}{2}\mathbb{E}_{\tilde{q}(r)}[r] \cdot \tilde{\mathbf{S}} = \mathbf{0} \quad \Longrightarrow \quad \mathbf{S} = \mathbb{E}_{\tilde{q}(r)}[r] \cdot \tilde{\mathbf{S}}.$$

F SPECIFICATIONS OF BENCHMARK TASKS

We selected a suite of benchmark tasks based on the experiments in (Tang et al., 2017) and (Xu and Zeng, 2024). This suite comprises real-world datasets sourced from the UCI Machine Learning Repository (Asuncion et al., 2007) and the KEEL dataset collection (Derrac et al., 2015), and synthetic data from the work of Neal (1997). The specifications for each dataset are detailed in Table 2 and Table 3.

For every dataset listed below (with the exception of "Neal with input outliers"), we also generated a corresponding version with output outliers for robustness evaluation, as described in Section 6 of the main text.

Table 2: Specifications of datasets used for non-sparse models.

Dataset	Number of Samples	Number of Features	Source
Neal	100	1	(Neal, 1997)
Neal with input outliers	100	1	(Neal, 1997)
Diabetes	43	2	KEEL
Machine CPU	209	6	KEEL
MPG	398	8	UCI
ELE	495	2	KEEL
Yacht	308	6	UCI
Boston	506	14	UCI

Table 3: Specifications of datasets used for sparse models.

Dataset	Number of Samples	Number of Features	Source
Concrete	1,030	9	UCI
Bike	17,389	16	UCI
Energy	768	8	UCI
Elevator	16,599	18	UCI
Protein	45,730	9	UCI
Kin8nm	8,192	8	UCI
Taxi	200,000	7	UCI

Taxi Dataset Preprocessing. The Taxi dataset required specific preprocessing due to its size and raw format. We used a subset consisting of the first 200,000 rows from the original data file. The following steps were performed:

- **Feature Engineering:** The ‘tpep_pickup_datetime‘ and ‘tpep_dropoff_datetime‘ columns were used to engineer a new feature, ‘trip_duration‘, representing the ride duration in seconds. The original datetime columns were subsequently removed.
- **Categorical Encoding:** The categorical feature ‘store_and_fwd_flag‘ was converted to a numerical format by mapping ‘N‘ to 0 and ‘Y‘ to 1.

From the processed columns, we selected the 7 features specified in Table 3 for our regression task. The ‘fare_amount‘ column was used as the target variable.

G SPECIFICATIONS OF BASELINE MODELS

To ensure numerical stability and prevent pathologically large hyperparameter values, we employed Maximum a Posteriori (MAP) estimation for the hyperparameters of all models. This approach, by incorporating regularization through priors, also imparts a degree of robustness even to the baseline GPR and SparseGPR models. Across all experimental settings, we consistently employed the Radial Basis Function kernel (Williams and Rasmussen, 2006) defined as:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\ell^2}\right), \tag{19}$$

where σ_f^2 denotes the outputscale and ℓ represents the lengthscale parameter. The specific, weakly informative priors were chosen based on common practices and recommendations from existing literature. The priors for kernel parameters and the likelihood scale were guided by the defaults in the BoTorch framework (Balandat et al., 2020). For the degrees of freedom parameter of the Student-t distributions, we adopted a Log-Normal prior, a choice supported by the analysis in Lee (2022). The priors used are as follows:

- Kernel lengthscale: $\text{Gamma}(3.0, 6.0)$
- Kernel outputscale: $\text{Gamma}(2.0, 0.15)$
- Likelihood scale parameter: $\text{LogNormal}(-4.0, 1.0)$
- Degrees of freedom for Student-t distributions: $\text{LogNormal}(1.0, 1.0)$

GPR and SparseGPR. For the Gaussian Process-based models, the hyperparameters optimized via MAP estimation were the kernel 'lengthscale', 'outputscale', and the Gaussian likelihood's noise parameter.

TP-based Models. The Student-t Process-based models evaluated in our experiments, which include the TangTPR and XuTPR baselines as well as our proposed TPR and SparseTPR, share common hyperparameters. For all these models, the kernel's 'outputscale' was fixed to 1.0, considering their model formulation, and was not treated as an optimizable parameter. The remaining hyperparameters were optimized via MAP estimation using their corresponding priors listed above.

H EXPERIMENTAL DETAILS

All experiments were conducted under two distinct hardware configurations, depending on the model type.

Experiments for the **non-sparse models** were run on nodes with 4 cores of an Intel Xeon Gold 6230R CPU and 8 GB of memory. A timeout of 3600 seconds (1 hour) was set for each run.

Experiments for the **sparse models** used nodes equipped with 4 cores of an Intel Xeon Gold 6230R CPU, 16 GB of memory, and one NVIDIA A100 GPU. The timeout for these runs was extended to 14,400 seconds (4 hours).

All experimental runs were completed within their respective time limits.

The software environment was based on Ubuntu 22.04.5 LTS and Python 3.12.2. The core implementation relies on key libraries including PyTorch (v2.8.0), NumPy (v2.3.3), scikit-learn (v1.7.2), and linear-operator (v0.6).

I ABLATION STUDY ON AN LA-BASED TPR BASELINE

To ensure a fair and robust comparison in our main experiments, we conducted an ablation study to determine the optimal hyperparameter configuration for the TangTPR baseline (Tang et al., 2017). The TangTPR model relies on an LA for inference, which itself requires an iterative optimization process to find the mode of the posterior distribution. This study investigates the impact of the number of optimization steps used within this approximation procedure on the final model performance. We tested three settings for the number of internal Adam optimization steps: 1, 10, and 100. All other experimental conditions, including datasets and hyperparameter priors, were identical to those described in Section 6.

The results, presented in Table 4, show that increasing the number of optimization steps generally leads to improved predictive accuracy (lower RMSE and lower PNLL) and a better model fit. While a setting of 10 steps often provides a good balance, the configuration with 100 steps consistently achieves the most competitive or superior performance across the majority of tasks, especially on datasets with outliers.

Based on this analysis, we selected 100 optimization steps as the fixed setting for the TangTPR baseline in all main experiments. This choice is justified by its consistently strong performance, establishing it as the most robust configuration for a fair comparison against our proposed method.

Table 4: Ablation study on the number of internal optimization steps for the LA in the TangTPR baseline. We report the mean and standard deviation over 10 folds for RMSE and Predictive Negative Log-Likelihood (PNLL). The **best**-performing configuration for each task and outlier setting is highlighted in **bold**.

		RMSE			**PNLL**		
Task	**Outlier**	Iter. 1	Iter. 10	Iter. 100	Iter. 1	Iter. 10	Iter. 100
Boston	No	$0.769_{(0.065)}$	$\mathbf{0.607}_{(0.101)}$	$0.619_{(0.109)}$	$1.059_{(0.202)}$	$\mathbf{0.803}_{(0.154)}$	$0.836_{(0.224)}$
	Yes	$\mathbf{0.798}_{(0.457)}$	$0.805_{(0.426)}$	$0.817_{(0.434)}$	$1.469_{(1.716)}$	$\mathbf{1.381}_{(1.179)}$	$2.180_{(2.796)}$
Diabetes	No	$\mathbf{0.938}_{(0.446)}$	$0.966_{(0.407)}$	$1.054_{(0.423)}$	$2.364_{(1.872)}$	$2.604_{(1.230)}$	$\mathbf{2.297}_{(1.738)}$
	Yes	$\mathbf{0.978}_{(0.455)}$	$1.009_{(0.393)}$	$1.047_{(0.384)}$	$2.498_{(1.859)}$	$\mathbf{2.359}_{(1.466)}$	$3.349_{(2.800)}$
ELE	No	$0.961_{(0.159)}$	$\mathbf{0.874}_{(0.146)}$	$0.963_{(0.402)}$	$3.194_{(1.869)}$	$2.892_{(1.487)}$	$\mathbf{2.664}_{(1.879)}$
	Yes	$0.933_{(0.565)}$	$0.895_{(0.558)}$	$\mathbf{0.813}_{(0.575)}$	$5.355_{(5.201)}$	$3.871_{(4.210)}$	$\mathbf{2.609}_{(2.673)}$
MPG	No	$0.566_{(0.125)}$	$\mathbf{0.528}_{(0.139)}$	$0.531_{(0.127)}$	$0.924_{(0.240)}$	$\mathbf{0.904}_{(0.242)}$	$0.978_{(0.302)}$
	Yes	$\mathbf{0.616}_{(0.535)}$	$0.699_{(0.505)}$	$0.647_{(0.518)}$	$\mathbf{0.969}_{(0.963)}$	$1.248_{(1.323)}$	$1.595_{(2.129)}$
Machine CPU	No	$0.799_{(0.684)}$	$0.801_{(0.685)}$	$\mathbf{0.770}_{(0.698)}$	$1.382_{(0.696)}$	$1.277_{(0.757)}$	$\mathbf{1.069}_{(0.470)}$
	Yes	$0.825_{(1.144)}$	$0.809_{(1.146)}$	$\mathbf{0.803}_{(1.179)}$	$3.257_{(7.125)}$	$2.050_{(4.026)}$	$\mathbf{0.925}_{(1.332)}$
Neal	No	$0.853_{(0.120)}$	$0.744_{(0.196)}$	$\mathbf{0.488}_{(0.182)}$	$3.019_{(2.275)}$	$1.854_{(1.429)}$	$\mathbf{0.753}_{(1.100)}$
	Yes (input)	$0.937_{(0.067)}$	$0.865_{(0.120)}$	$\mathbf{0.470}_{(0.150)}$	$2.543_{(0.493)}$	$4.662_{(3.943)}$	$\mathbf{1.098}_{(0.604)}$
	Yes	$0.747_{(0.089)}$	$0.595_{(0.167)}$	$\mathbf{0.457}_{(0.277)}$	$2.527_{(1.397)}$	$2.364_{(1.055)}$	$\mathbf{0.643}_{(0.505)}$
Yacht	No	$0.750_{(0.217)}$	$0.388_{(0.189)}$	$\mathbf{0.219}_{(0.073)}$	$0.934_{(0.825)}$	$0.719_{(1.336)}$	$\mathbf{-0.418}_{(0.623)}$
	Yes	$0.722_{(0.528)}$	$0.735_{(0.478)}$	$\mathbf{0.509}_{(0.556)}$	$2.145_{(1.982)}$	$0.670_{(1.132)}$	$\mathbf{0.652}_{(1.391)}$

J ABLATION STUDY ON A VI-BASED TPR BASELINE

This section details the hyperparameter tuning for the XuTPR baseline (Xu and Zeng, 2024), another key comparison method in our main experiments. The XuTPR model relies on MC sampling to approximate the terms in its Evidence Lower Bound (ELBO). We conducted an ablation study to analyze the impact of the number of MC samples on model performance. We evaluated three configurations, using 10, 100, and 1000 MC samples for the ELBO estimation. All other experimental settings were kept identical to those in the main experiments, as described in Section 6.

The results, summarized in Table 5, clearly indicate a strong positive correlation between the number of MC samples and overall performance. Increasing the sample size consistently leads to a lower final RMSE, lower PNLL, and a better-optimized ELBO. While a larger number of samples incurs a higher computational cost, establishing a strong performance benchmark for a fair comparison requires a precise approximation of the objective function.

Therefore, we selected 1000 MC samples as the fixed setting for the XuTPR baseline in all main experiments. This choice is justified by the data, which shows this configuration consistently yields the most accurate predictions and the best model fit across nearly all tested scenarios.

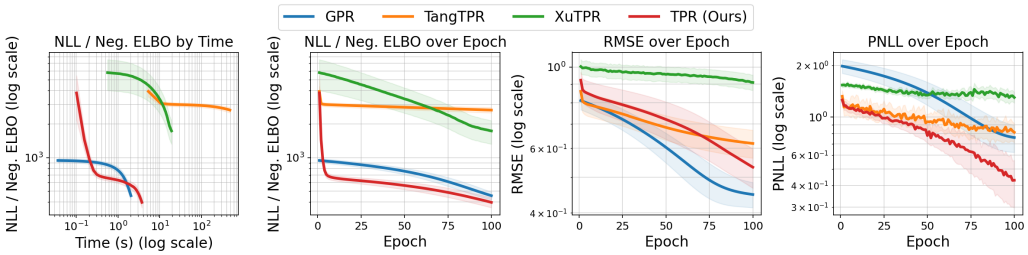
Table 5: Ablation study on the number of MC samples used for ELBO estimation in the XuTPR baseline. We report the mean and standard deviation over 10 folds for RMSE and Predictive Negative Log-Likelihood (PNLL). The **best**-performing configuration for each task and outlier setting is highlighted in **bold**.

		RMSE			**PNLL**		
Task	**Outlier**	10 Samples	100 Samples	1000 Samples	10 Samples	100 Samples	1000 Samples
Boston	No	$1.196_{(0.240)}$	$0.943_{(0.086)}$	$\mathbf{0.908}_{(0.087)}$	$4.479_{(1.724)}$	$1.612_{(0.352)}$	$\mathbf{1.390}_{(0.190)}$
	Yes	$1.149_{(0.417)}$	$0.956_{(0.393)}$	$\mathbf{0.928}_{(0.423)}$	$3.435_{(2.083)}$	$1.759_{(0.743)}$	$\mathbf{1.430}_{(0.859)}$
Diabetes	No	$1.122_{(0.410)}$	$\mathbf{0.952}_{(0.415)}$	$1.012_{(0.357)}$	$3.406_{(1.604)}$	$2.205_{(1.317)}$	$\mathbf{1.967}_{(0.907)}$
	Yes	$\mathbf{0.991}_{(0.415)}$	$1.028_{(0.359)}$	$1.030_{(0.375)}$	$3.599_{(2.580)}$	$4.440_{(6.382)}$	$\mathbf{1.906}_{(0.724)}$
ELE	No	$2.125_{(1.294)}$	$1.097_{(0.713)}$	$\mathbf{0.718}_{(0.217)}$	$2.766_{(1.852)}$	$\mathbf{1.279}_{(0.501)}$	$1.341_{(0.560)}$
	Yes	$3.897_{(4.399)}$	$0.902_{(0.673)}$	$\mathbf{0.889}_{(0.685)}$	$2.785_{(2.041)}$	$1.375_{(1.132)}$	$\mathbf{1.217}_{(0.924)}$
MPG	No	$1.128_{(0.201)}$	$0.780_{(0.143)}$	$\mathbf{0.673}_{(0.135)}$	$3.199_{(1.137)}$	$1.577_{(0.348)}$	$\mathbf{0.950}_{(0.209)}$
	Yes	$0.998_{(0.487)}$	$0.792_{(0.530)}$	$\mathbf{0.700}_{(0.504)}$	$2.968_{(1.271)}$	$1.792_{(1.636)}$	$\mathbf{1.241}_{(1.188)}$
Machine CPU	No	$3.546_{(1.912)}$	$0.800_{(0.628)}$	$\mathbf{0.659}_{(0.683)}$	$5.397_{(3.279)}$	$0.765_{(0.296)}$	$\mathbf{0.285}_{(0.351)}$
	Yes	$2.589_{(1.187)}$	$0.878_{(1.121)}$	$\mathbf{0.734}_{(1.158)}$	$7.208_{(5.381)}$	$2.240_{(2.542)}$	$\mathbf{0.655}_{(1.222)}$
Neal	No	$2.070_{(1.121)}$	$0.736_{(0.261)}$	$\mathbf{0.657}_{(0.225)}$	$2.679_{(1.479)}$	$0.828_{(0.627)}$	$\mathbf{0.231}_{(0.340)}$
	Yes (input)	$1.667_{(0.927)}$	$1.016_{(0.645)}$	$\mathbf{0.734}_{(0.304)}$	$4.020_{(2.335)}$	$0.558_{(0.390)}$	$\mathbf{0.223}_{(0.331)}$
	Yes	$1.746_{(1.209)}$	$0.851_{(0.425)}$	$\mathbf{0.628}_{(0.347)}$	$5.098_{(4.825)}$	$0.614_{(0.546)}$	$\mathbf{0.322}_{(0.601)}$
Yacht	No	$0.925_{(0.306)}$	$0.720_{(0.273)}$	$\mathbf{0.635}_{(0.233)}$	$3.655_{(3.442)}$	$0.694_{(0.719)}$	$\mathbf{0.355}_{(0.525)}$
	Yes	$1.080_{(0.457)}$	$0.738_{(0.466)}$	$\mathbf{0.571}_{(0.539)}$	$3.065_{(2.054)}$	$1.149_{(1.033)}$	$\mathbf{1.044}_{(1.603)}$

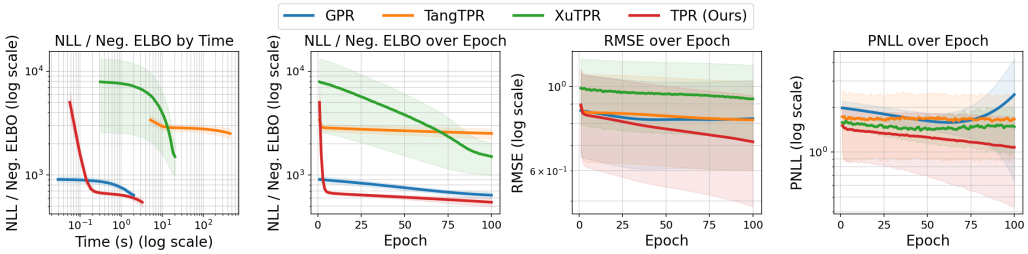
K DETAILED EXPERIMENTAL RESULTS

This section presents comprehensive learning curves and quantitative results for the experiments discussed in Section 6.

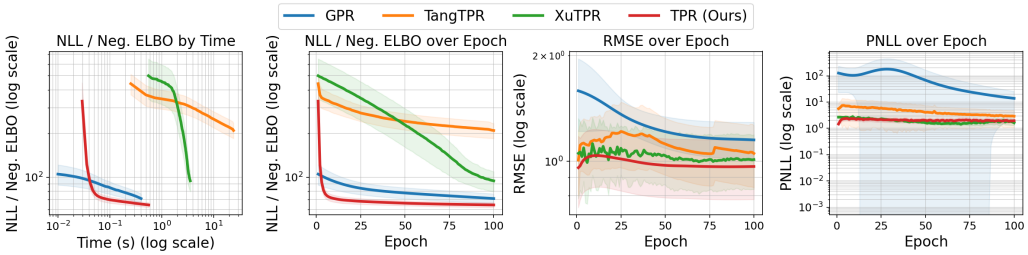
K.1 Full-Batch Model Results



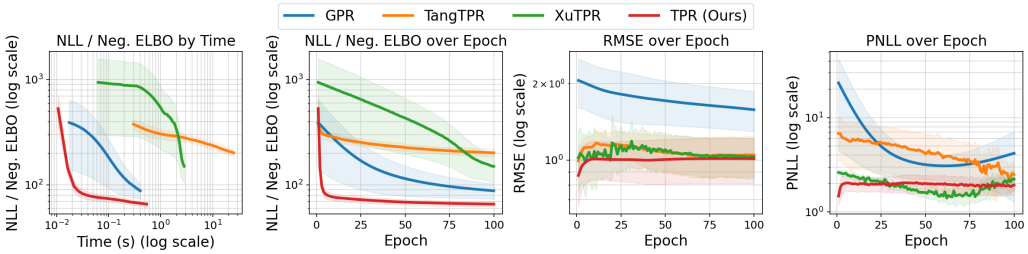
(a) Boston



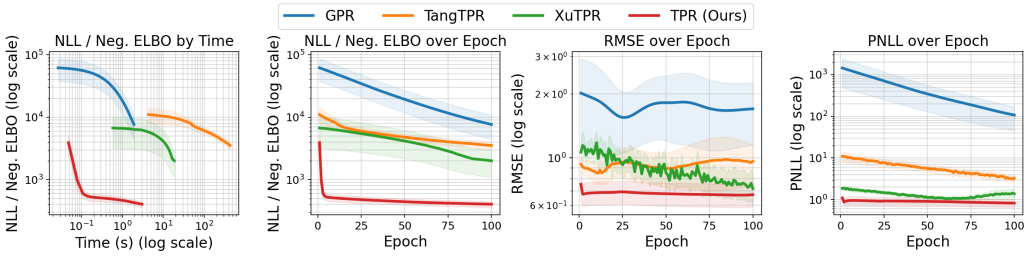
(b) Boston Outliers



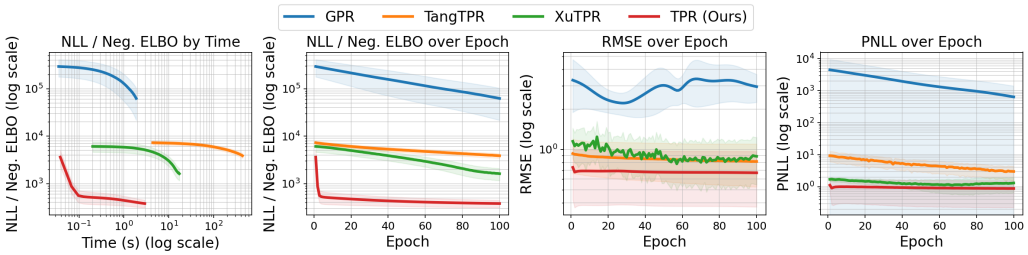
(c) Diabetes



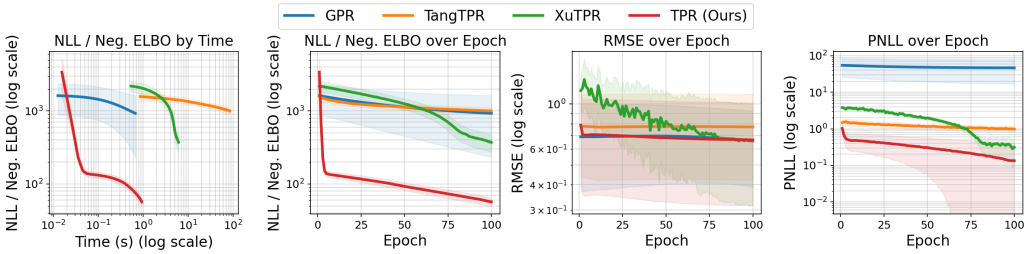
(d) Diabetes Outliers



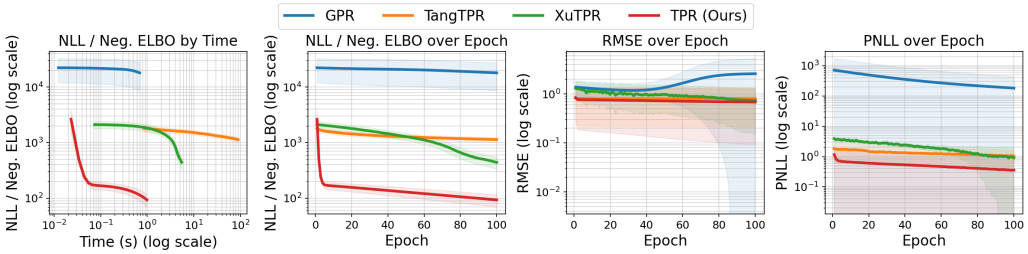
(e) ELE



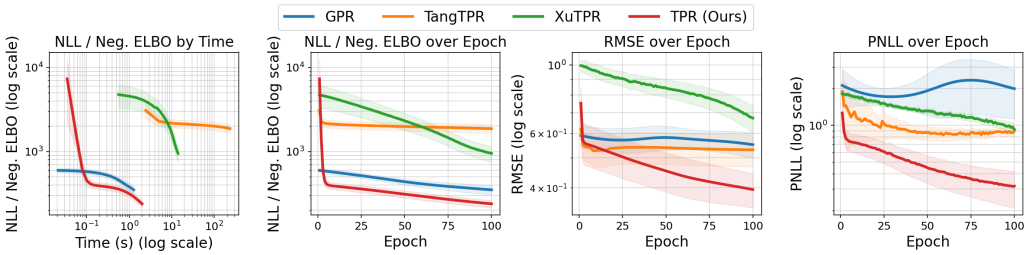
(f) ELE Outliers



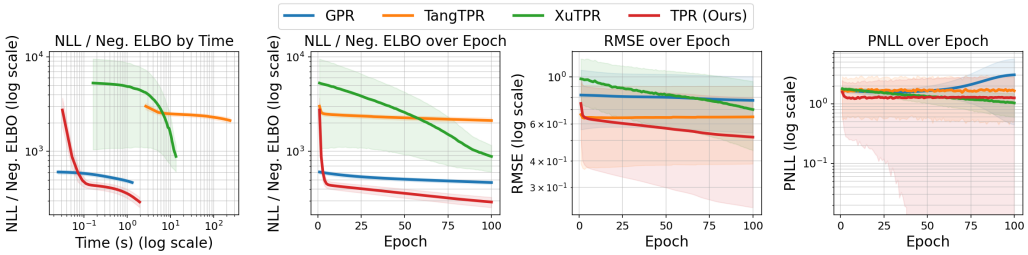
(g) Machine CPU



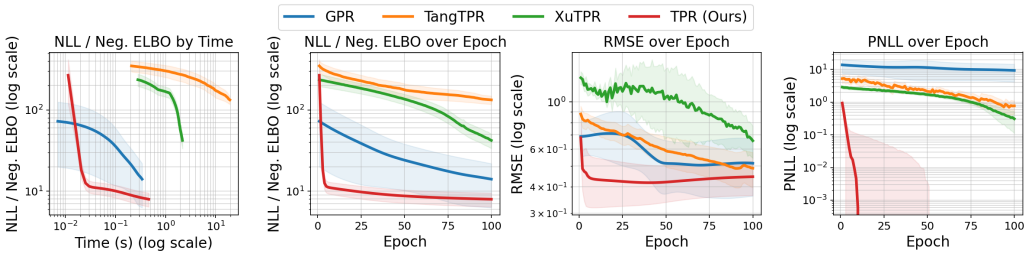
(h) Machine CPU Outliers



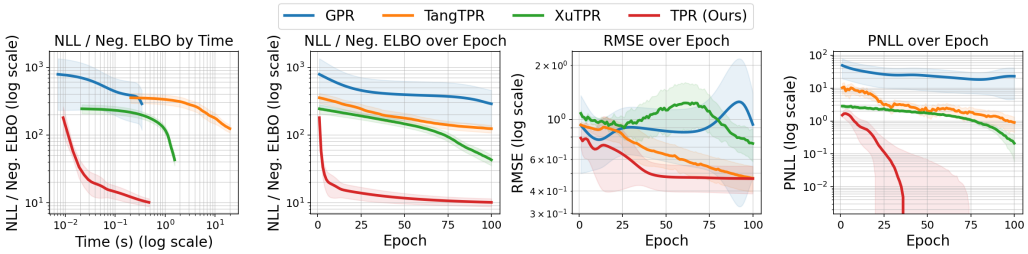
(i) MPG



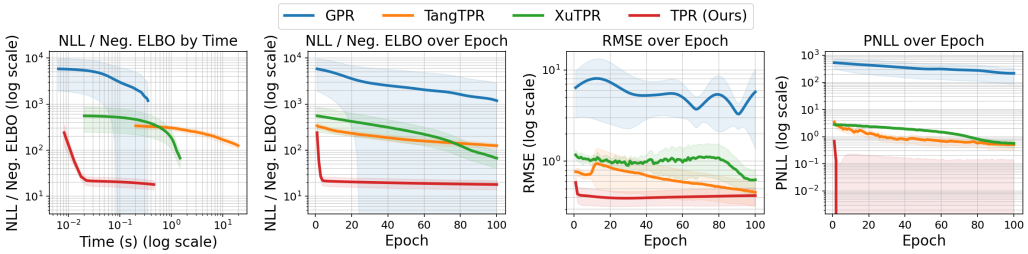
(j) MPG Outliers



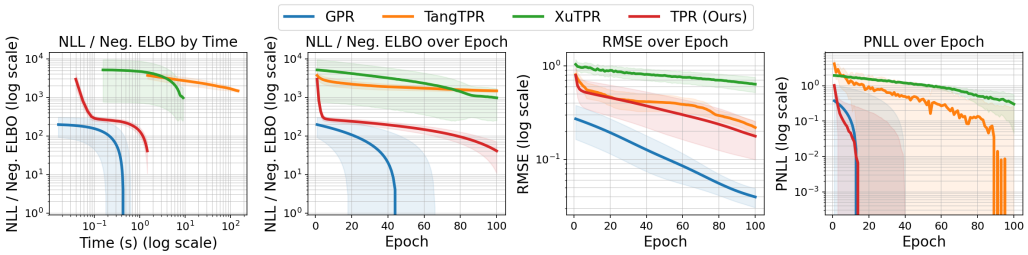
(k) Neal



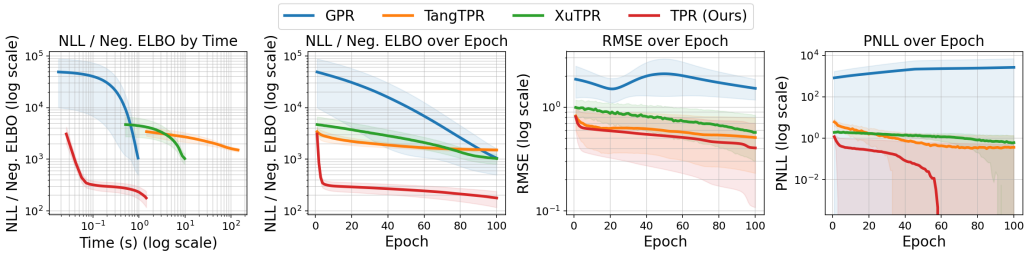
(l) Neal Outliers on inputs



(m) Neal Outliers



(n) Yacht



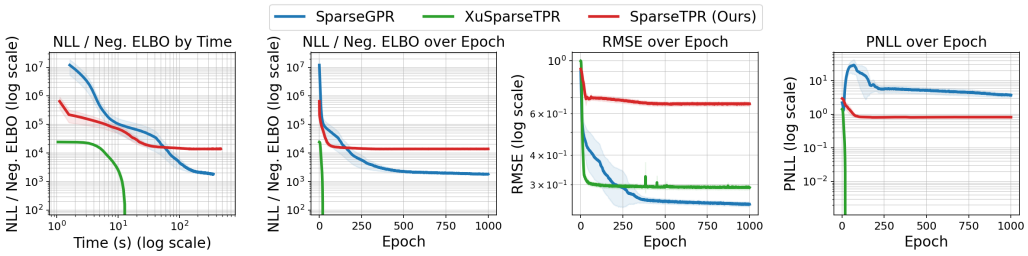
(o) Yacht Outliers

Figure 4: Learning curves for all full-batch models across a representative benchmark task (Boston dataset). Within the figure, the four panels show: (from left to right) the model's objective function (NLL/Neg. ELBO) vs. wall-clock time, the objective function vs. epochs, RMSE vs. epochs, and PNLL vs. epochs. The solid line represents the mean performance over 10 folds, and the shaded area indicates one standard deviation. Lower and faster-converging curves indicate better performance. The plots visually demonstrate the superior convergence speed and stability of our proposed TPR model (red) compared to the baselines.

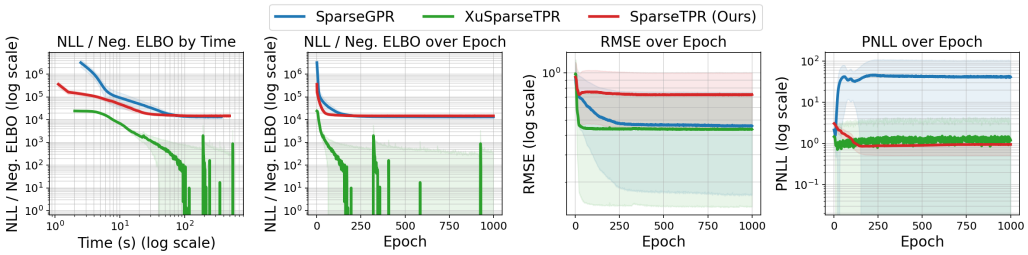
Table 6: Comprehensive performance comparison of full-batch models on all benchmark tasks. We report the mean and standard deviation over 10-fold cross-validation for Root Mean Squared Error (RMSE) and Predictive Negative Log-Likelihood (PNLL). Values ≥ 10 are rounded to integers and the **best**-performing model for each task and outlier setting is highlighted in **bold**.

		RMSE				**PNLL**			
Task	**Outlier**	GPR	TangTPR	XuTPR	TPR (Ours)	GPR	TangTPR	XuTPR	TPR (Ours)
Boston	No	**0.449**$_{(0.071)}$	0.619$_{(0.109)}$	0.908$_{(0.087)}$	0.533$_{(0.132)}$	0.760$_{(0.272)}$	0.815$_{(0.211)}$	1.299$_{(0.182)}$	**0.429**$_{(0.263)}$
	Yes	0.823$_{(0.431)}$	0.817$_{(0.434)}$	0.928$_{(0.423)}$	**0.715**$_{(0.465)}$	2.429$_{(3.594)}$	1.664$_{(1.507)}$	1.478$_{(0.973)}$	**1.077**$_{(1.291)}$
Diabetes	No	1.151$_{(0.281)}$	1.054$_{(0.423)}$	1.012$_{(0.357)}$	**0.967**$_{(0.374)}$	14$_{(23)}$	2.906$_{(3.670)}$	**1.627**$_{(0.638)}$	1.952$_{(0.802)}$
	Yes	1.577$_{(0.565)}$	1.047$_{(0.384)}$	1.030$_{(0.375)}$	**1.014**$_{(0.408)}$	4.188$_{(5.846)}$	2.473$_{(1.623)}$	2.225$_{(1.096)}$	**1.925**$_{(0.855)}$
ELE	No	1.701$_{(1.107)}$	0.963$_{(0.402)}$	0.718$_{(0.217)}$	**0.671**$_{(0.166)}$	108$_{(124)}$	3.272$_{(1.286)}$	1.381$_{(0.782)}$	**0.824**$_{(0.348)}$
	Yes	2.956$_{(1.394)}$	0.813$_{(0.575)}$	0.889$_{(0.685)}$	**0.669**$_{(0.595)}$	640$_{(1500)}$	3.000$_{(2.802)}$	1.308$_{(1.294)}$	**0.881**$_{(1.315)}$
MPG	No	0.552$_{(0.104)}$	0.531$_{(0.127)}$	0.673$_{(0.135)}$	**0.395**$_{(0.100)}$	1.997$_{(1.725)}$	0.897$_{(0.129)}$	0.928$_{(0.166)}$	**0.317**$_{(0.210)}$
	Yes	0.774$_{(0.461)}$	0.647$_{(0.518)}$	0.700$_{(0.504)}$	**0.519**$_{(0.555)}$	3.054$_{(5.220)}$	1.648$_{(2.274)}$	**1.031**$_{(0.875)}$	1.254$_{(3.103)}$
Machine CPU	No	**0.657**$_{(0.533)}$	0.770$_{(0.698)}$	0.659$_{(0.683)}$	0.665$_{(0.699)}$	46$_{(59)}$	0.982$_{(0.359)}$	0.313$_{(0.429)}$	**0.134**$_{(0.527)}$
	Yes	2.594$_{(5.365)}$	0.803$_{(1.179)}$	0.734$_{(1.158)}$	**0.684**$_{(1.180)}$	181$_{(514)}$	1.036$_{(1.208)}$	0.903$_{(2.027)}$	**0.357**$_{(1.630)}$
Neal	No	0.515$_{(0.307)}$	0.488$_{(0.182)}$	0.657$_{(0.225)}$	**0.445**$_{(0.163)}$	9.398$_{(10)}$	0.773$_{(0.656)}$	0.310$_{(0.399)}$	**-0.120**$_{(0.146)}$
	Yes (input)	0.936$_{(0.936)}$	0.470$_{(0.150)}$	0.734$_{(0.304)}$	**0.468**$_{(0.165)}$	23$_{(37)}$	0.900$_{(0.895)}$	0.210$_{(0.304)}$	**-0.117**$_{(0.153)}$
	Yes	5.749$_{(8.743)}$	0.457$_{(0.277)}$	0.628$_{(0.347)}$	**0.421**$_{(0.184)}$	218$_{(259)}$	0.504$_{(0.373)}$	0.564$_{(0.369)}$	**-0.072**$_{(0.404)}$
Yacht	No	**0.040**$_{(0.018)}$	0.219$_{(0.073)}$	0.635$_{(0.233)}$	0.177$_{(0.155)}$	**-2.201**$_{(0.726)}$	$-0.030_{(0.573)}$	0.302$_{(0.514)}$	$-0.652_{(0.326)}$
	Yes	1.517$_{(0.683)}$	0.509$_{(0.556)}$	0.571$_{(0.539)}$	**0.404**$_{(0.593)}$	2657$_{(8378)}$	0.366$_{(1.943)}$	0.615$_{(1.458)}$	**-0.143**$_{(1.265)}$

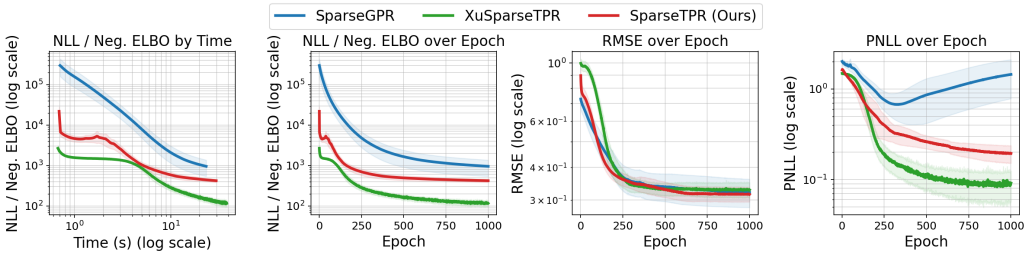
K.2 Sparse Model Results



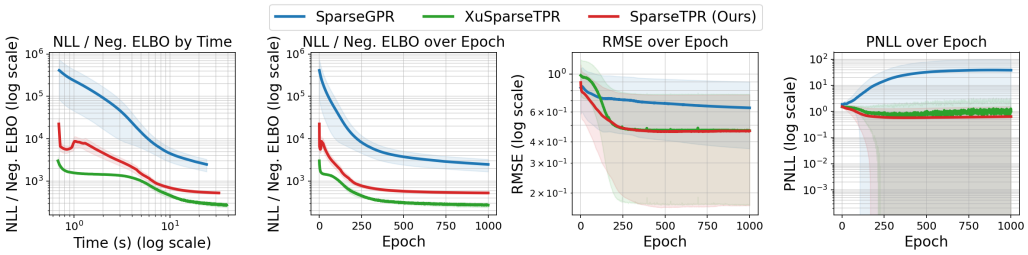
(a) Bike



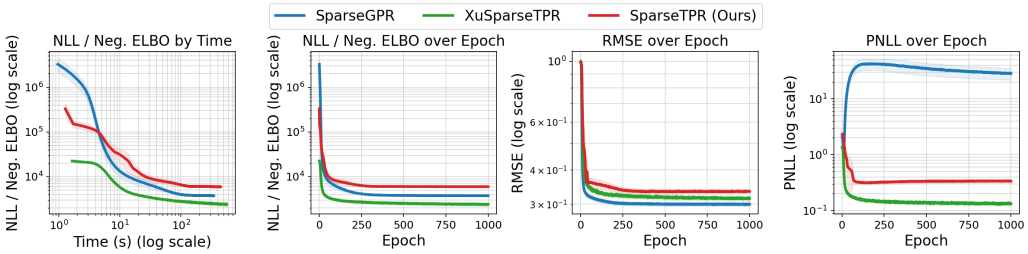
(b) Bike Outliers



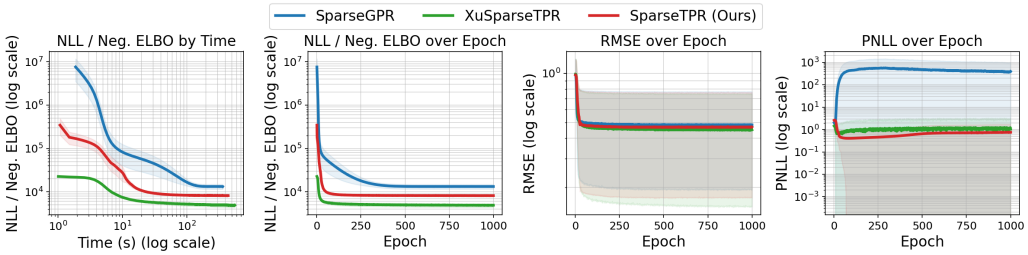
(c) Concrete



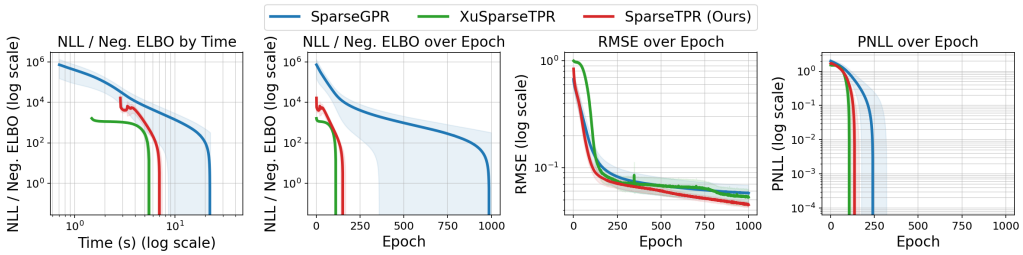
(d) Concrete Outliers



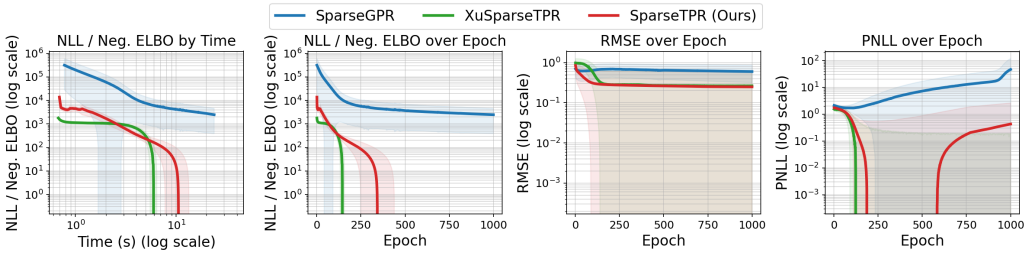
(e) Elevator



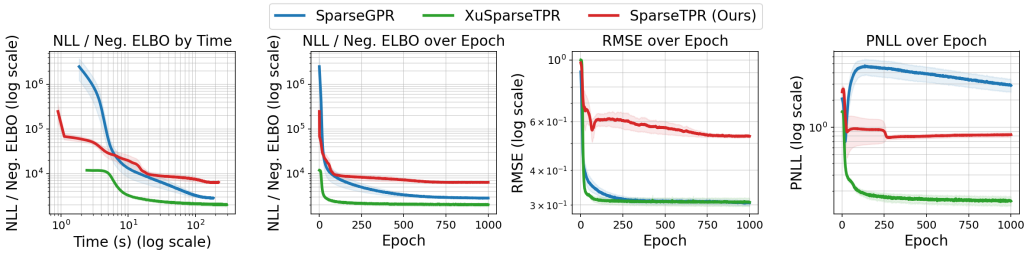
(f) Elevator Outliers



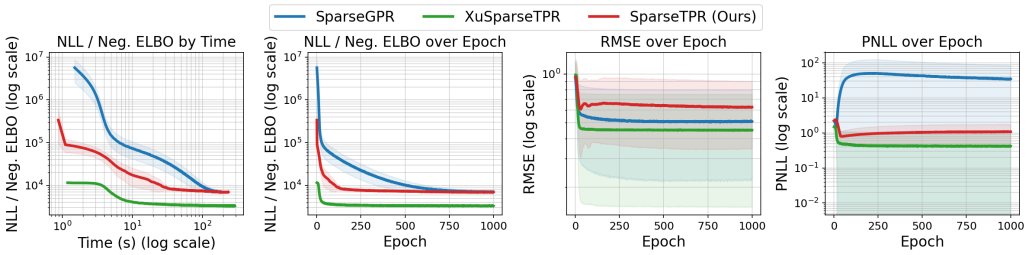
(g) Energy



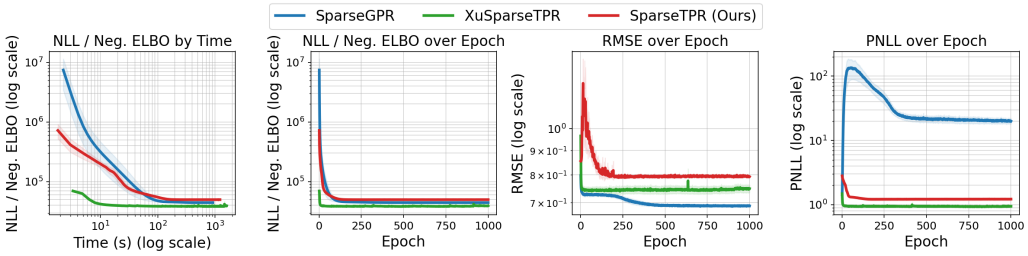
(h) Energy Outliers



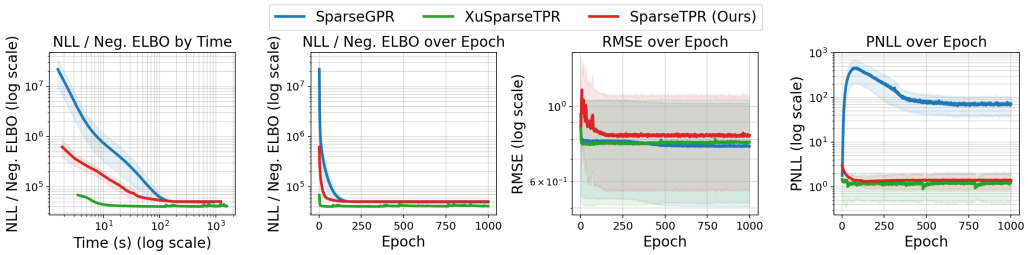
(i) Kin8nm



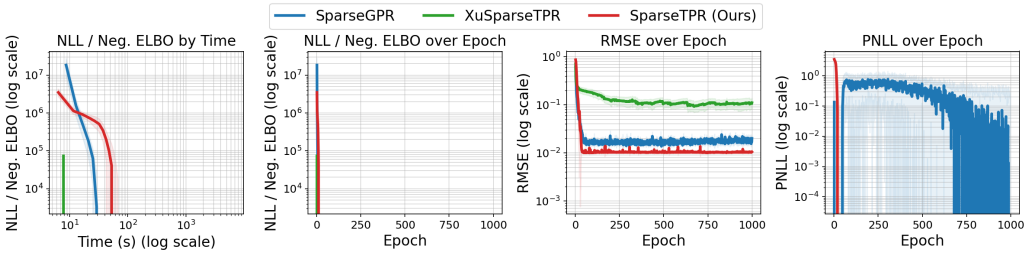
(j) Kin8nm Outliers



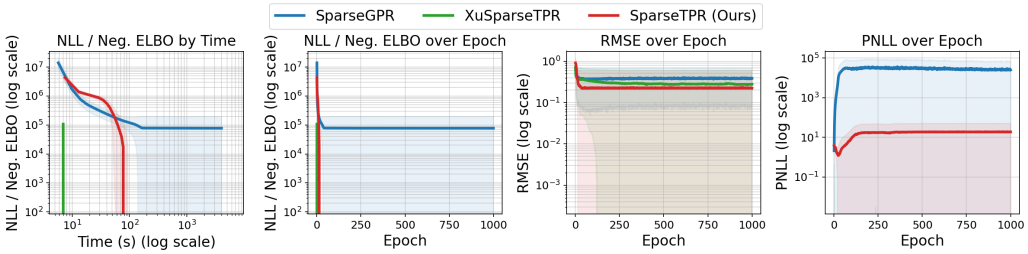
(k) Protein



(l) Protein Outliers



(m) Taxi



(n) Taxi Outliers

Figure 5: Learning curves for all sparse models across a representative large-scale benchmark task (Concrete dataset). The four panels are structured identically to Figure 4, showing the objective function vs. wall-clock time and epochs, RMSE vs. epochs, and PNLL vs. epochs. The plots highlight the competitive performance and robustness of our SparseTPR model (red).

Table 7: Comprehensive performance comparison of sparse models on all large-scale benchmark tasks. We report the mean and standard deviation over 10-fold cross-validation for RMSE and Predictive Negative Log-Likelihood (PNLL). Values ≥ 10 are rounded to integers and the **best**-performing model for each task and outlier setting is highlighted in **bold**.

		RMSE			PNLL		
Task	**Outlier**	SparseGPR	XuSparseTPR	SparseTPR (Ours)	SparseGPR	XuSparseTPR	SparseTPR (Ours)
Bike	No	**0.247**$_{(0.010)}$	0.292$_{(0.013)}$	0.658$_{(0.025)}$	3.728$_{(1.003)}$	**-0.211**$_{(0.036)}$	0.824$_{(0.035)}$
	Yes	0.457$_{(0.585)}$	**0.435**$_{(0.591)}$	0.726$_{(0.554)}$	41$_{(122)}$	1.201$_{(4.752)}$	**0.947**$_{(0.861)}$
Concrete	No	0.321$_{(0.079)}$	0.328$_{(0.043)}$	**0.315**$_{(0.041)}$	1.450$_{(1.333)}$	**0.092**$_{(0.066)}$	0.194$_{(0.085)}$
	Yes	0.633$_{(0.537)}$	0.465$_{(0.591)}$	**0.462**$_{(0.587)}$	38$_{(115)}$	1.036$_{(3.329)}$	**0.638**$_{(1.786)}$
Elevators	No	**0.299**$_{(0.010)}$	0.315$_{(0.012)}$	0.333$_{(0.014)}$	28$_{(14)}$	**0.132**$_{(0.020)}$	0.336$_{(0.050)}$
	Yes	0.482$_{(0.579)}$	**0.450**$_{(0.595)}$	0.468$_{(0.592)}$	394$_{(1217)}$	1.048$_{(3.464)}$	**0.753**$_{(1.869)}$
Energy	No	0.058$_{(0.021)}$	0.053$_{(0.019)}$	**0.045**$_{(0.005)}$	$-0.912_{(0.345)}$	$-0.992_{(0.104)}$	**-1.118**$_{(0.039)}$
	Yes	0.596$_{(0.549)}$	0.263$_{(0.653)}$	**0.248**$_{(0.660)}$	47$_{(144)}$	**-0.503**$_{(1.388)}$	0.430$_{(4.367)}$
Kin8nm	No	**0.305**$_{(0.013)}$	0.307$_{(0.021)}$	0.531$_{(0.013)}$	2.884$_{(0.890)}$	**0.154**$_{(0.034)}$	0.829$_{(0.096)}$
	Yes	0.510$_{(0.576)}$	**0.451**$_{(0.599)}$	0.624$_{(0.555)}$	35$_{(99)}$	**0.420**$_{(1.230)}$	1.075$_{(1.327)}$
Protein	No	**0.689**$_{(0.007)}$	0.749$_{(0.026)}$	0.793$_{(0.009)}$	20$_{(3.969)}$	**0.945**$_{(0.020)}$	1.221$_{(0.118)}$
	Yes	**0.764**$_{(0.517)}$	0.785$_{(0.534)}$	0.823$_{(0.519)}$	71$_{(64)}$	**1.287**$_{(1.752)}$	1.408$_{(1.072)}$
Taxi	No	0.020$_{(0.006)}$	0.111$_{(0.055)}$	**0.011**$_{(0.001)}$	$-0.031_{(0.875)}$	$-1.266_{(0.052)}$	**-1.351**$_{(0.002)}$
	Yes	0.388$_{(0.609)}$	0.279$_{(0.649)}$	**0.224**$_{(0.666)}$	25272$_{(79864)}$	**-0.779**$_{(1.444)}$	19$_{(64)}$

L ABLATION STUDY ON OPTIMIZATION ITERATIONS

In our main experiments, the number of optimization iterations for all full-batch models was set to 100. This configuration was chosen to maintain consistency with the experimental protocol established in prior work on TPR (Tang et al., 2017).

To address the concern regarding convergence, we conducted an ablation study to evaluate the impact of increasing the number of iterations to 500. Table 8 provides a comprehensive comparison of the predictive performance (RMSE and PNLL) for both the XuTPR baseline and our proposed TPR model at this extended iteration count.

The results confirm that increasing the iterations to 500 does not significantly alter the comparative landscape established at 100 iterations; our proposed TPR model continues to demonstrate superior or highly competitive performance across the vast majority of tasks. While the sampling-based XuTPR baseline shows numerical improvements with more iterations and occasionally surpasses our model in certain metrics (e.g., RMSE on the Boston task with outliers), these gains come at a substantial computational price.

As illustrated in Figure 4, XuTPR already consumes significantly more wall-clock time than our model at 100 iterations. Further increasing the iteration count for sampling-based methods exacerbates this inefficiency, whereas our proposed method reaches a high-quality local optimum almost immediately due to its closed-form coordinate ascent updates. This study reinforces that our method provides an optimal balance between predictive accuracy and practical computational efficiency.

Table 8: Predictive performance with 500 iterations. Values represent the mean ± standard deviation over 10 folds, reported to three decimal places. Results are shown for RMSE and PNLL.

		RMSE		**PNLL**	
Task	**Out.**	XuTPR	TPR (Ours)	XuTPR	TPR (Ours)
Boston	No	$0.438_{(0.108)}$	$\mathbf{0.347}_{(0.103)}$	$0.355_{(0.162)}$	$\mathbf{0.177}_{(0.212)}$
	Yes	$\mathbf{0.587}_{(0.500)}$	$0.616_{(0.524)}$	$\mathbf{0.899}_{(1.518)}$	$0.907_{(1.936)}$
Diabetes	No	$\mathbf{1.030}_{(0.365)}$	$1.049_{(0.369)}$	$\mathbf{1.787}_{(0.927)}$	$1.890_{(0.902)}$
	Yes	$\mathbf{1.021}_{(0.443)}$	$1.071_{(0.378)}$	$2.092_{(0.943)}$	$\mathbf{1.972}_{(1.078)}$
ELE	No	$0.643_{(0.180)}$	$\mathbf{0.588}_{(0.114)}$	$1.549_{(1.424)}$	$\mathbf{0.716}_{(0.294)}$
	Yes	$0.782_{(0.600)}$	$\mathbf{0.632}_{(0.592)}$	$0.961_{(1.133)}$	$\mathbf{0.724}_{(0.931)}$
MPG	No	$0.428_{(0.087)}$	$\mathbf{0.329}_{(0.085)}$	$0.510_{(0.135)}$	$\mathbf{0.218}_{(0.202)}$
	Yes	$0.571_{(0.530)}$	$\mathbf{0.464}_{(0.549)}$	$\mathbf{0.768}_{(0.825)}$	$1.013_{(2.902)}$
Mach. CPU	No	$\mathbf{0.568}_{(0.602)}$	$0.632_{(0.709)}$	$\mathbf{-0.130}_{(0.370)}$	$-0.039_{(0.561)}$
	Yes	$\mathbf{0.663}_{(1.150)}$	$0.729_{(1.150)}$	$0.378_{(2.185)}$	$\mathbf{0.283}_{(1.575)}$
Neal	No	$0.542_{(0.246)}$	$\mathbf{0.457}_{(0.175)}$	$-0.099_{(0.267)}$	$\mathbf{-0.190}_{(0.110)}$
	In	$0.576_{(0.330)}$	$\mathbf{0.455}_{(0.189)}$	$-0.039_{(0.436)}$	$\mathbf{-0.182}_{(0.117)}$
	Out	$0.513_{(0.235)}$	$\mathbf{0.433}_{(0.163)}$	$0.154_{(0.649)}$	$\mathbf{-0.010}_{(0.445)}$
Yacht	No	$0.069_{(0.039)}$	$\mathbf{0.050}_{(0.018)}$	$-1.053_{(0.590)}$	$\mathbf{-1.106}_{(0.053)}$
	Yes	$0.257_{(0.624)}$	$\mathbf{0.249}_{(0.626)}$	$\mathbf{-0.665}_{(1.693)}$	$-0.356_{(2.033)}$