Online Student-t Processes with an Overall-local Scale Structure for Modelling Non-stationary Data

Taole Sha
University of Hong Kong
u3577089@connect.hku.hk

Michael Minyi Zhang
University of Hong Kong
mzhang18@hku.hk

Abstract

Mixture-of-expert (MOE) models are popular methods in machine learning, since they can model heterogeneous behaviour across the space of the data using an ensemble collection of learners. These models are especially useful for modelling dynamic data as time-dependent data often exhibit non-stationarity and heavy-tailed errors, which may be inappropriate to model with a typical single expert model. We propose a mixture of Student-t processes with an adaptive structure for the covariance and noise behaviour for each mixture. Moreover, we use a sequential Monte Carlo (SMC) sampler to perform online inference as data arrive in real time. We demonstrate the superiority of our proposed approach over other models on synthetic and real-world datasets to prove the necessity of the novel method.

1 INTRODUCTION

In modelling dynamical systems, it is common that the data will exhibit non-stationarity, where the functional behavior changes across the input space. Kernel methods like the Gaussian process (GP) are a popular choice of prior distribution over real-valued functions in Bayesian models of time series data (Rasmussen and Williams, 2005). However, updating the model in real-time is not trivial, as typical inference algorithms are a function of the entire data set. Moreover, common choices for the covariance kernel assumes stationarity, and non-stationary kernels are often too computationally demanding to use in practical scenarios.

The Student-t process (TP) is indeed such a prior distribution over real-valued functions (Shah et al., 2014). One attractive feature of the TP is that it can model heavy-tailed errors through its degrees of freedom parameter, allowing more modelling flexibility compared to the GP. Similar to the GP, the TP has consistent marginals and closed-form conditionals which make it as convenient as the GP to use in statistical modelling. However, TPs are still liable to suffer from the aforementioned issues that GPs face when modelling data. *In this paper, we introduce a mixture of TPs with an SMC sampler so that we may take advantage of the additional flexibility of a mixture-of-experts model with a convenient online inference algorithm.*

Our paper proceeds as follows: In Section 2, we discuss previous work in online GP regression. Then, we introduce the TP-mixture model and the online inference algorithm in Section 3, whose application on bandit optimization problems can be found in Section 8 of the supplementary materials. We use the experiment results on synthetic and real-world datasets to compare it with GP-based online models in Section 4. Finally, we conclude the paper in Section 5 with a discussion of future work.

2 RELATED WORK

A GP distributed function, $f \sim \mathcal{GP}(\mu(\cdot), \Sigma(\cdot,\cdot))$, is defined by a mean function, $\mu(\cdot)$, and a covariance function, $\Sigma(\cdot,\cdot)$, with a property that GPs are multivariate normally distributed conditioned on a finite set of points: $f(x) \sim \mathcal{N}_N(\mu(x), \Sigma(x,x'))$ (Rasmussen and Williams, 2005). This multivariate normal property leads to tractable posterior inference in many classes of models, however it suffers from the an $O(N^3)$ computational cost, for N being the number of observations.

Numerous scalable methods have been developed for the computational issue of GPs, where sparse inducing point methods are a popular technique (Snelson and Ghahramani, 2006; Titsias, 2009; Bauer et al., 2016). The sparse GP methods form a low-rank approximation of the kernel using M "pseudo-inputs", reducing the

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

computational complexity to $O(NM^2)$. Product-of-expert models employ a block diagonal approximation of the full covariance matrix to reduce the complexity of the full matrix inversion to inverting each smaller block (Deisenroth and Ng, 2015; Cohen et al., 2020). While not necessarily faster, mixture-of-expert models use a mixture of GPs to model functions with greater flexibility compared to a single GP (Rasmussen and Ghahramani, 2001; Meeds and Osindero, 2005).

For fast online GP methods, Csató and Opper (2002) used variational inference to approximate the posterior in a sparse online GP model but the hyperparameters are assumed to be fixed. Nguyen-tuong et al. (2008) proposed a product-of-experts local online GP method, though it ignores the correlation between experts, which can lead to poor uncertainty quantification. Bui et al. (2017) developed a sparse variational GP regression approach called OSVGP that online updates the hyperparameters.

However, OSVGP tends to be numerically unstable and empirically is liable to underfit the data. As analyzed in Bauer et al. (2016), sparse variational GPs will inherently overestimate the noise variance term leading to an underfit model. This is because the covariance of the variational objective depends only on the low-rank approximation to the kernel and a homoscedastistic variance term.

Thus, any misfit of the data to the model or any additional complexity of the model is penalized through the noise variance term, σ^2. To address the numerical instability and underfitting issues in OSVGP, Stanton et al. (2021) developed an exact sparse online model called WISKI, where a structured and sparse covariance matrix approximation developed by Wilson and Nickisch (2015) is used, leading to constant computational complexity concerning the number of observations.

Regarding SMC methods in GPs, Svensson et al. (2015) proposed an SMC sampler to marginalize the kernel hyperparameters and Gramacy and Polson (2011) proposed an SMC sampler for sequential design in GPs. While these SMC methods allow for online updating, they cannot account for non-stationary data, nor can they limit the computational cost which still scales $O(N^3)$.

Later research in using Monte Carlo methods can directly model non-stationarity using mixtures of GPs. For example, Zhang and Williamson (2019) proposed an importance sampling method for scaling up a mixture-of-experts GP model to an average complexity of $O(N^3/K^2)$ for non-stationary data, where K stands for the number of mixtures. Later, Zhang et al. (2023) extended the importance sampler in Zhang and Williamson (2019) by using an SMC sampler and modeled the mixture of GPs using a Dirichlet process mixture rather than a finite Dirichlet mixture. In parallel, Härkönen et al. (2022) also extended Zhang and Williamson (2019) using SMC^2 (Chopin et al., 2013) to infer the model parameters for a mixture of GPs.

Despite the advances in online mixtures of GPs, little attention has been paid to online mixtures of the Student-t process. Student-t priors have long been used in Bayesian linear regression for modelling sparse regression coefficients or heavy-tailed errors (Fernández and Steel, 1999; Tipping, 2001; West, 1984; Geweke, 1993). Vanhatalo et al. (2009) introduced a robust method of GP regression where the latent function was GP distributed but the observation likelihood was assumed to be a Student-t distribution.

However, they estimated parameters using a Laplace approximation to the posterior distribution instead of performing exact Bayesian inference. Later, Jylänki et al. (2011) used an expectation propagation algorithm for posterior inference in the same model. Again, expectation propagation is only an approximate method that cannot exactly capture the underlying uncertainty.

Student-t processes (TPs) are an attractive alternative choice of latent function prior to the Gaussian processes. Zhang and Yeung (2010) derived a TP model for multi-task learning but they wrongly assumed the noise to be independent in this model. Shah et al. (2014) obtained a TP as the marginalization of an inverse Wishart process prior on the covariance of a Gaussian process distributed function.

They further showed TP's robustness to model misspecfication and an improved covariance matrix for modeling tail-dependence. But under their derivation, the prior-likelihood relationship is unclear when modelling noisy data. Tang et al. (2017) combined both a Student-t process model with Student-t noise, but, again, used only a Laplace approximation for the posterior instead of performing exact inference.

3 ONLINE STUDENT-t PROCESSES FOR NON-STATIONARY DATA

In our online setting, data arrives sequentially across time, the $\mathbf{x}_i \in \mathbb{R}^D$ is the input covariate that includes the variable "time", and $y_i \in \mathbb{R}$ is the output of interest. We now present the data-generating process for our proposed Student-t process mixture-of-experts (TP-MOE) model, the notations and intuitions of which will be explained in the order of the inputs to the output, and how we update their parameters when a new observation arrives. The data generating process is as follows:

$$
\begin{aligned}
&\mathbf{x}_i \sim \mathcal{T}_D(\nu_{z_i}, \boldsymbol{\mu}_{z_i}, \boldsymbol{\Psi}_{z_i}),\\
&\alpha \sim \text{Gamma}(a_0, b_0), \quad z_i|\alpha \sim \text{CRP}(\alpha),\\
&\boldsymbol{\theta}_k \sim \log\mathcal{N}(m_0, s_0^2\mathbf{I}), \quad \nu_k \sim \text{Gamma}(2, 0.1),\\
&h_k \sim \mathcal{N}(0, k_0^2), \quad k_0^2 \sim \text{Inv-Gamma}\left(\frac{1}{2}, \frac{1}{2}\right),\\
&\mathbf{y}_k|\mathbf{X}_k, \boldsymbol{\theta}_k, \sim \mathcal{T}_{N_k}(\nu_k, 0, \mathbf{K}_{\boldsymbol{\theta}_k} + |h_k|\mathbf{I}).
\end{aligned} \tag{1}
$$

The i-th D-dimensional input $\mathbf{x}_i$ is modelled as a Dirichlet process Gaussian-inverse Wishart mixture model with the scale-location parameters marginalized out (Antoniak, 1974), which allows efficient online inference. To explain, $\mathbf{x}_i \sim \mathcal{N}_D(\mathbf{M}_{z_i}, \mathbf{C}_{z_i})$, the latent parameters $(\mathbf{M}_{z_i}, \mathbf{C}_{z_i})$ are integrated out over a normal-inverse Wishart prior, $\mathcal{NIW}(\boldsymbol{\mu}_{z_i}, \lambda_{z_i}, \boldsymbol{\Psi}_{z_i}, \nu_{z_i})$. This marginalization results in a likelihood of $\mathbf{x}_i$ that is a multivariate t distribution $\mathcal{T}_D(\nu_{z_i}, \boldsymbol{\mu}_{z_i}, \boldsymbol{\Psi}_{z_i})$[1]. We assign the data to a cluster using the latent variable z_i according to the predictive distribution of the Dirichlet process–the Chinese restaurant process (Aldous, 1985), the procedure of which will be detailed later this section.

Given the N_k number of inputs $\mathbf{X}_k$ and the outputs $\mathbf{y}_k$ from cluster $z_i = k$, we obtain the Student-t process by marginalizing out the GP-distributed mapping function, $f(\mathbf{X}_k)$, and the overall noise term of each expert, σ_k^2:

$$
\begin{aligned}
&\sigma_k^2|\nu_k \sim \text{Inv-Gamma}\left(\frac{\nu_k}{2}, \frac{\nu_k}{2}\right),\\
&f(\mathbf{X}_k)|\mathbf{X}_k, \sigma_k^2, \boldsymbol{\theta}_k \sim \mathcal{N}_{N_k}(0, \sigma_k^2\mathbf{K}_{\boldsymbol{\theta}_k}),\\
&\mathbf{y}_k|f(\mathbf{X}_k), \sigma_k^2, h_k \sim \mathcal{N}_{N_k}(f(\mathbf{X}_k), \sigma_k^2|h_k|\mathbf{I}),
\end{aligned} \tag{2}
$$

$$
\begin{aligned}
\Rightarrow &\mathbf{y}_k|\mathbf{X}_k, \nu_k, h_k, \boldsymbol{\theta}_k\\
&= \int\int P(\mathbf{y}_k|f(\mathbf{X}_k))P(f(\mathbf{X}_k)|\mathbf{X}_k, \sigma_k^2, \boldsymbol{\theta}_k)P(\sigma_k^2)\\
&\qquad \mathrm{d}f(\mathbf{X}_k)\mathrm{d}\sigma_k^2\\
&\sim \mathcal{T}_{N_k}(\nu_k, 0, \mathbf{K}_{\boldsymbol{\theta}_k} + |h_k|\mathbf{I}).
\end{aligned} \tag{3}
$$

The above equations explain the model assumption and how the marginalization is obtained in closed form by integrating out the latent parameters. After marginalizing, we have a local scale parameter $|h_k|$ for the noise to control the heteroscedasticity of the expert. We model $|h_k|$ as a folded normal distribution $h_k \sim N(0, k_0^2)$ according to the suggestion of Gelman (2006), which proposes to use folded-t distributions (of which, the folded normal is a special case) as a prior on the observation noise parameters in hierarchical Bayesian models due to the fact that these priors do not exhibit problematic behavior in posterior inference compared to other common choices of priors.

[1] $\mathbf{x}_i$ is a fixed observation and not a random parameter. We only treat their scale and location parameters (which are marginalized out) as random.

Under our construction of the TP, the relationship between the prior over σ_k^2, $f(\mathbf{X}_k)$ and the likelihood contributed from $\mathbf{y}_k$ is clear. TP is used as a prior over the regression function f, yielding the same modelling flexibility as GP for the noisy data. The details of the TP derivation, and its marginal likelihood together with the predictive distribution can be found in Section 6 of the supplementary materials.

When the i-th streaming observation $(\mathbf{x}_i, y_i)$ arrives in the online setting, we assign it to mixture k (in other words, we set $z_i = k$) according to the Chinese restaurant process (CRP). We model the mixtures using a CRP due to its flexibility–*a priori* there is assumed to be an infinite number of mixtures but *a posteriori* there are only finitely many instantiated mixtures thereby allowing us to bypass the difficult task of tuning the number of experts in the model. Moreover, the CRP naturally allows us to update the mixture assignments when new data arrive because it is defined sequentially as a predictive process:

$$
P(z_i = k|\alpha, \mathbf{X}_k) \propto \begin{cases} N_k' \cdot \mathcal{T}_D(\nu_k', \boldsymbol{\mu}_k', \boldsymbol{\Psi}_k') & k \in K^+.\\ \alpha \cdot \mathcal{T}_D(\nu_0, \boldsymbol{\mu}_0, \boldsymbol{\Psi}_0) & \text{o.w.} \end{cases} \tag{4}
$$

K^+ refers to the existing clusters, and all $(\cdot)'$ represent summary statistics calculated with first $i-1$ observations[2]. The inputs' Student-t likelihood's parameters $(\nu_k', \boldsymbol{\mu}_k', \boldsymbol{\Psi}_k')$ are easily obtained by a standard conjugate result:

$$
\begin{aligned}
\boldsymbol{\mu}_k' =& \frac{\lambda_0\boldsymbol{\mu}_0 + N_k'\bar{\mathbf{x}}_k'}{\lambda_k'}, \quad \nu_k' = \nu_0 + N_k' - D + 1,\\
\boldsymbol{\Psi}_k' =& \frac{\lambda_k' + 1}{\lambda_k'\nu_k'}\left(\boldsymbol{\Psi}_0 + \mathbf{S}_k' + \mathbf{S}_{\bar{\mathbf{x}}_k}'\right),\\
\bar{\mathbf{x}}_k' =& \frac{\sum_{i':(z_{i'}=k, i'<i)}\mathbf{x}_{i'}}{N_k'}, \quad N_k' = \sum_{i'=1}^{i-1} I(z_{i'} = k),\\
\lambda_k' =& \lambda_0 + N_k',\\
\mathbf{S}_k' =& \sum_{i':(z_{i'}=k, i'<i)} (\mathbf{x}_{i'} - \bar{\mathbf{x}}_k')(\mathbf{x}_{i'} - \bar{\mathbf{x}}_k')^T,\\
\mathbf{S}_{\bar{\mathbf{x}}_k}' =& \frac{\lambda_0 N_k'}{\lambda_k'}(\bar{\mathbf{x}}_k' - \boldsymbol{\mu}_0)(\bar{\mathbf{x}}_k' - \boldsymbol{\mu}_0)^T.
\end{aligned} \tag{5}
$$

The DP concentration parameter α influences how likely a new cluster starts as shown in Equation (3), which we expect to be data-driven. A convenient choice is using a variable augmentation scheme to simplify the sampling procedure (Escobar and West, 1995), where a Gamma prior is placed on α, and the full conditional

[2] For example, N_k' stands for the number of existing observations in cluster k

posterior up to observation i can be sampled as below:

$$\begin{aligned}\rho|\alpha \sim&\text{Beta}(\alpha+1, i),\ K=|\{k: N_k>0\}|,\\ \frac{\pi_\alpha}{1-\pi_\alpha} =&\frac{a_0+K-1}{N(b_0-\log\rho)},\\ \alpha|\mathbf{z}_{1:i}, \pi_\alpha, \rho =&\pi_\alpha\cdot\text{Gamma}(\alpha_0+K, b_0-\log\rho)\\ &+(1-\pi_\alpha)\cdot\text{Gamma}(\alpha_0+K-1, b_0-\log\rho).\end{aligned} \tag{6}$$

The degree of freedom parameter ν_k has a prior distribution Gamma$(2, 0.1)$, since it puts mass on a large range of reasonable values for the degree of freedom as suggested by Juárez and Steel (2010). We sample the degrees of freedom parameter also through an efficient variable augmentation scheme. First, we instantiate the overall variance parameter, σ_k^2, since given it ν_k is independent of all other terms.

Due to the conjugacy between the Gaussian likelihood and the inverse Gamma prior, we can directly Gibbs sample the σ_k^2 from its full conditional. Then, conditioned on σ_k^2, we sample ν_k using the slice sampler from $P(\nu_k|\sigma_k^2)$ (Neal, 2003; Damien et al., 1999)[3].

$$\begin{aligned}&\sigma_k^2|\mathbf{X}_k, \mathbf{y}_k, \nu_k \sim \text{Inv-Gamma}(\alpha'_{\sigma^2},\ \beta'_{\sigma^2}),\\ &\alpha'_{\sigma^2}=\frac{\nu_k+N'_k}{2}, \beta'_{\sigma^2}=\frac{\nu_k+\mathbf{y}_k^T(\mathbf{K}_{\boldsymbol{\theta}_k}+|h_k|\mathbf{I})^{-1}\mathbf{y}_k}{2}\end{aligned} \tag{7}$$

We assume a hierarchical structure on the local heteroscedasticity parameter, $|h_k|$, where global scale k_0^2 is shared over all mixtures. Here, we will share scale data from other clusters to inform the posterior sampling of h_k. Because h_k has a normal prior, we can again sample the full conditional of k_0^2 in closed form:

$$k_0^2|h_1,\ldots,h_K \sim \text{Inv-Gamma}\left(\frac{K+1}{2}, \frac{1+\sum_{i=1}^K h_i^2}{2K}\right). \tag{8}$$

Then we sample h_k and the TP kernel hyperparameters $\boldsymbol{\theta}_k$ jointly from their posterior using the elliptical slice sampler (ESS), which is a sampling algorithm for non-conjugate models with Gaussian priors (Murray et al., 2010). Since h_k has a Gaussian prior and $\boldsymbol{\theta}_k$ has a log-normal prior, we use the ESS as an effective sampling algorithm in this setting.

3.1 SMC for online TP-MOE

For inference, we use a sequential Monte Carlo (SMC) sampler in order to update the model as new data arrive (Del Moral et al., 2006). SMC follows from importance sampling (IS) and sequential importance sampling (SIS) algorithms, where samples $\theta^{(j)}, j=1,\ldots,J$ (also known as "particles") are taken from a proposal distribution $Q(\theta)$ to approximate an intractable integral of a function $f(\theta)$ over the target distribution $P(\theta)$ by a weighted sum, where the weights are given by the ratios $w^{(j)}=\frac{P(\theta^{(j)})}{Q(\theta^{(j)})}$:

$$\int f(\theta)P(\theta)\mathrm{d}\theta \approx \sum_{j=1}^J w^{(j)} f(\theta^{(j)}), \tag{9}$$

IS and SIS suffer from the particle degeneracy problem where one proposal weight, $w^{(j)}$ dominates other proposals. Then they lose the advantage of Monte Carlo method and their approximation has a large variance. But SMC method we adopt in this paper can avoid the problem by a resampling step, which will be detailed later this section.

When adopting SMC for online inferring our model, each of the $j=1,\ldots,J$ particles $(\mathbf{z}^{(j)}, \boldsymbol{\theta}^{(j)}, \mathbf{h}^{(j)}, \nu^{(j)}, \alpha^{(j)})$ corresponds to an expert, each of which has its own instantiation of the mixture model from Eq. 1. Each expert is updated as described in last section when a new observation arrives. Then, we calculate the particle weights, resulting in a posterior weighted sample of TP product-of-experts models. The initial weight for particle j at time i is:

$$w_1^{(j)} \propto P(y_1|z_1^{(j)}, \mathbf{x}_1, \boldsymbol{\theta}^{(j)}, h^{(j)}, \nu^{(j)})P(\mathbf{x}_1|z_1^{(j)}, \alpha^{(j)}). \tag{10}$$

The updating procedure for time $i>1$ is shown in Algorithm 1.

The computational complexity is dominated by the inversion of a $N_k\times N_k$ matrix when calculating the TPs' marginal likelihood, the formula of which can be found in Section 6 of supplementary materials. Assuming that the average size of a cluster is $N_k=N/K$, the number of data divided by the number of clusters, then the computational complexity will be $\mathcal{O}(JN^3/K^2)$. Under the basic setting of our sampler, the complexity of the sampler still grows as new data arrive so the method cannot truly be considered "online".

To this end, we adopt the "minibatched" stochastic approximation that is widely used for reducing the computational complexity of posterior inference (Zhang et al., 2023; Zhang and Williamson, 2019; Minsker et al., 2014; Srivastava et al., 2015). We collect subsample of size B from the mixture with N_k observations is drawn uniformly without replacement by sampling the indices, $\mathbf{u}_k^{(j)}$, from a hypergeometric distribution. Then, their likelihood is calculated and upweighted by N_k/B power to approximate the full likelihood, which is then integrated over the prior to approximate the marginal

[3] α'_{σ^2} and β'_{σ^2} in the equations are auxiliary quantities to facilitate computation, and are not parameters of interest.

Algorithm 1: SMC Sampler for TP-MOE

Input: New observation $(\mathbf{x}_i, y_i)$

for $j = 1, \cdots, J$ *in parallel* **do**

Sample $z_i^{(j)} = k$ from $P(z_i^{(j)}|\alpha^{(j)}, \mathbf{X}_{1:i-1})$

Sample $\alpha^{(j)}$ from the full conditional $P(\alpha^{(j)}|\mathbf{z}_{1:i})$

Sample $\theta_k^{(j)}$ and $h_k^{(j)}$ jointly by using the ESS

Sample $(k_0^2)^{(j)}$ from $P((k_0^2)^{(j)}|h_1^{(j)}, \ldots, h_K^{(j)})$

Sample $(\sigma_k^2)^{(j)}$ from $P(\sigma_k^2|\mathbf{X}_k, \mathbf{y}_k, \nu_k^{(j)})$

Sample $\nu_k^{(j)}$ by using the slice sampler

Update particle weight:

$$w_i^{(j)} = w_{i-1}^{(j)} P(\mathbf{x}_i|\alpha^{(j)}, z_i^{(j)}) \times \frac{P\left(\mathbf{y}_{1:i}|\mathbf{X}_{1:i}, \theta_{k,i}^{(j)}, h_k^{(j)}, \nu_k^{(j)}\right)}{P\left(\mathbf{y}_{1:i-1}|\mathbf{X}_{1:i-1}, \theta_k'^{(j)}, h_k'^{(j)}, \nu_k'^{(j)}\right)} \quad (11)$$

end

Normalize weights:

$$w_i^{(j)} := \frac{w_i^{(j)}}{\sum_{j=1}^J w_i^{(j)}}$$

if $N_{eff} < \frac{J}{2}$ **then**

Resample particles $(\mathbf{z}_{1:i}^{(\mathbf{j}^*)}, \theta_k^{(\mathbf{j}^*)}, h_k^{(\mathbf{j}^*)}, \nu_k^{(\mathbf{j}^*)}, \alpha^{(\mathbf{j}^*)})$, where $\mathbf{j}^* \sim \text{Multinomial}(J, w_i^{(1)}, \cdots, w_i^{(j)})$

Set $w_i^{(j)} := \frac{1}{J}$ for $j = 1, \cdots, J$

end

Output: Particle weights $(w_i^{(1)}, \cdots, w_i^{(j)})$ and particles $(\mathbf{z}_{1:i}^{(1:J)}, \theta^{(1:J)}, \mathbf{h}^{(1:J)}, \nu^{(1:J)}, \alpha^{(1:J)})$

likelihood. The stochastic approximation leads us to:

$$\begin{aligned}
\mathbf{u}_k^{(j)} &= (u_1, \ldots, u_B) \\
&\sim \text{HyperGeometric}\big(B, \{i : z_i^{(j)} = k\}\big), \\
(\mathbf{y}_{\mathbf{u}_k^{(j)}}, \mathbf{X}_{\mathbf{u}_k^{(j)}}) &= \big(y_u, \mathbf{x}_u : u \in \mathbf{u}_k^{(j)}\big). \\
P\big(\mathbf{y}_{\mathbf{u}_k^{(j)}}|\mathbf{X}_{\mathbf{u}_k^{(j)}}, \boldsymbol{\theta}_k^{(j)}, h_k^{(j)}, \nu_k^{(j)}\big) &\\
&\sim \mathcal{T}_B\left(\nu_k^{(j)}, 0, \mathbf{K}_{\boldsymbol{\theta}_k^{(j)}} + \frac{N_k|h_k^{(j)}|}{B}\mathbf{I}\right). \quad (12)
\end{aligned}$$

Such a stochastic marginal likelihood approximation for TP works only under our overall-local scale structure model assumption since a clear prior-likelihood relationship leads to a well-defined marginal likelihood. With minibatching, the complexity is reduced to $\mathcal{O}(J\min\{N_k, B\}^3/K^2)$. As each particle can be updated independently, the parallel computation can be adopted to further reduce the complexity to $\mathcal{O}(\min\{N_k, B\}^3/K^2)$. After updating the particles' weights, we calculate the effective sample size, $N_{eff} = 1/\sum_{j=1}^J (w_i^{(j)})^2$. A small effective sample size indicates the particle degeneracy problem, and if it is lower than a certain threshold, typically $J/2$, the particles are resampled to only preserve the high-weighted ones. This helps avoid the aforementioned particle degeneracy problem, and reduces the variance of SMC inference results. We make predictions using a weighted average as in Algorithm 2.

Algorithm 2: TP-MOE Prediction

for $j = 1, \cdots, J$ **do**

Predict new observations on particle j with

$$\begin{aligned}
p_k &\propto N_k \cdot P(\mathbf{x}_*|z_* = k, \mathbf{X}_k, -) \\
P(y_*^{(j)}|\mathbf{y}, \mathbf{X}, \mathbf{x}_*, -) &\\
&= \sum_{k \in \mathbf{K}^+} p_k \cdot P(y_{k*}^{(j)}|\mathbf{y}_k, \mathbf{X}_k, \mathbf{x}_*, -) \quad (13)
\end{aligned}$$

end

Average predictions:

$P(\bar{y}_*|\mathbf{y}, \mathbf{X}, \mathbf{x}_*) = \sum_{j=1}^J w_i^{(j)} P(y_*^{(j)}|\mathbf{y}, \mathbf{X}, -)$

4 EXPERIMENTS

The choice of hyperparameters could significantly influence the MOE models' performance. According to Zhang and Williamson (2019), larger J and B will lead to better performance but more computation time. In this section, we proceed to study the advantages of the heavy tails by implementing the TP-MOE and other Gaussian-based online models on different non-stationary datasets and analysing their performances in terms of one-step-ahead predictions. The GP models include a Gaussian mixture-of-experts model (GP-MOE) (Zhang et al., 2023), a sparse online GP method using the Woodbury identity and structured kernel interpolation (WISKI) (Stanton et al., 2021), and an online sparse variational GP method (OSVGP) (Bui et al., 2017)[4].

For the experiments, we sequentially predict the next observation and update the models with the real data point. The one-step predictive mean squared error (MSE), log-likelihood and continuous rank probability

[4]The implementation for GP-MOE is available at *https://github.com/michaelzhang01/GPMOE*. The code of OSVGP and WISKI are available at: *https://github.com/wjmaddox/online_gp*. Our code with documentation can be found at *https://github.com/stllllll/TP-MOE*.

score (CRPS) (larger log-likelihood and lower CRPS represent better uncertainty quantification) are adopted to evaluate the results. The datasets used include[5]:

1. Synthetic data generated from a single GP with Gaussian noise, where both the noise's variance and the length scale of its radial basis function kernel are 0.5 (N=100).
2. Synthetic data generated from a mixture of TPs with Student-t noise, where for the first half, both the noise's variance and the length scale of its radial basis function kernel are 0.5, with the degree of freedom being 5. For the second half, both the noise's variance and the length scale of its radial basis function kernel are 0.2, with the degree of freedom being 2 (N=100).
3. An accelerometer measurement of a motorcycle crash (N=94).
4. The price of Brent crude oil (N=1025).
5. The daily features of Dow Jones Industrial Average (DJI dataset) (N = 112).
6. The annual carbon dioxide output in Canada (N=215).
7. The MIMIC-III data set where we model a patient's heart rate (Johnson et al., 2016) (N = 10000).
8. The annual water level of the Nile River data (N=100).
9. The exchange rate between the Euro and the US Dollar (N=3139)[6].
10. The Dow Jones Index dataset containing 30 stocks, the first stock's return is chosen to be the output, while other stocks' and time are used as inputs (N=25).
11. The returns of Istanbul Stock Exchange, the time and 7 international indexes are used as inputs (N = 536).
12. 22 series of exchange rates data, a non-stationary one is chosen to be the output, similarly others and time are inputs (N = 101).
13. A wind power generation dataset, 9 features including time are inputs (N = 10950) [7].
14. An electric power consumption dataset, 6 features including time are inputs (N = 10484) [8].

[5]The motorcycle dataset can be found in the R package `VarReg`; the Brent, Canada CO_2 and Nile River datasets at: *https://github.com/alan-turing-institute/TCPD*; the EUR-USD dataset in the R package `priceR`.

[6]*https://www.kaggle.com/datasets/brunotly/foreign-exchange-rates-per-dollar-20002019*

[7]*https://www.kaggle.com/datasets/mubashirrahim/wind-power-generation-data-forecasting*

[8]*https://www.kaggle.com/datasets/fedesoriano/electric-power-consumption*

The first two are synthetic datasets that illustrate our motivations. The subsequent datasets exhibit non-stationarity in both the length-scale and the noise, while the Nile River dataset shows only time-varying mean values and the exchange rate dataset is a series of non-stationary noise. The last five datasets have multi-dimensional inputs. All the datasets were then pre-processed to have zero mean and unit variance.

To make the results comparable, The TP-MOE and the GP-MOE share the same particle number $J = 100$ and the same 16 cores used on a shared memory process based on OpenMP, and the number of inducing points for all sparse models is set to be 50. The OSVGP's number of optimization iterations is set to the default value of 1. The radial basis function kernel for all models is: $\Sigma(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\theta}{2}\sum_{d=1}^{D}(x_d - x'_d)^2\right\}$. For WISKI, the memory complexity makes it impossible for high-dimensional inputs (Stanton et al., 2021). Thus, we use PCA to project the inputs of multi-dimensional datasets into two dimensions for WISKI.

The sample runs on a subset of the datasets analyzed are plotted in Figures 1 to 6, and the plots for the remaining datasets can be found in the supplementary materials Section 7. The plots contain the data points, one-step predictive mean (plotted with solid red lines) and 95% predictive interval (plotted with dashed black lines). The data points in MOE models' plots are coloured according to the cluster assignment given by the particle with the highest weight. The results in terms of the three chosen metrics are shown in Tables 1, 2, and 3 respectively. The CPU wall time in seconds is reported in Table 4.

When facing the synthetic data from a single stationary GP, according to Table 2, 3, WISKI achieves the best overall result since it's basically a stationary online GP model. TP-MOE and GP-MOE still perform comparatively in such a situation. While for the datasets generated from a mixture of two TPs, the trend becomes hard for WISKI and OSVGP to capture, where MOE models are a necessity and TP-MOE performs the best in terms of three metrics.

For the datasets with non-stationary length-scale and noise (Motorcycle - Heart Rate datasets), as seen in Table 1, TP-MOE performs better than the GP-based models and always achieves lower predictive MSE. Also see the sample runs plots (some are in Section 7 of the supplementary materials), showing that MOE models better capture the heterogeneity of the underlying function than the stationary models. Moreover, TP-MOE provides better uncertainty quantification in terms of the predictive log-likelihood and the CRPS as in Tables 2 and 3.

The OSVGP tends to underfit as expected, while the

Table 1: One-step predictive MSE. One standard error reported in parentheses

	TP-MOE	GP-MOE	WISKI	OSVGP
Single GP	**0.574 (0.009)**	0.587 (0.010)	0.588 (0.000)	0.975 (0.008)
TP Mixture	**0.476 (0.008)**	0.514 (0.037)	1.120 (0.000)	1.006 (0.001)
Motorcycle	**0.363 (0.028)**	0.381 (0.038)	0.631 (0.000)	0.998 (0.002)
Brent	**0.055 (0.019)**	0.057 (0.016)	0.177 (0.000)	0.862 (0.007)
DJI	**0.034 (0.002)**	0.040 (0.005)	0.083 (0.000)	0.715 (0.030)
Canada	**0.015 (0.003)**	0.016 (0.004)	0.048 (0.000)	0.711 (0.030)
Heart Rate	**0.354 (0.005)**	0.438 (0.009)	0.412 (0.000)	0.744 (0.036)
Nile	**0.738 (0.017)**	0.752 (0.025)	0.767 (0.000)	0.908 (0.008)
EUR-USD	1.010 (0.004)	**1.004 (0.003)**	1.007 (0.000)	1.010 (0.001)
Dow	**0.943 (0.009)**	0.945 (0.013)	1.011 (0.000)	0.948 (0.000)
Istanbul	**0.649 (0.010)**	0.654 (0.011)	0.684 (0.000)	0.991 (0.000)
Exchange	**0.439 (0.009)**	0.449 (0.011)	0.574 (0.000)	1.000 (0.000)
Wind	**0.369 (0.009)**	0.401 (0.006)	0.427 (0.000)	0.998 (0.000)
Power	**0.740 (0.043)**	0.810 (0.029)	0.760 (0.000)	0.989 (0.003)

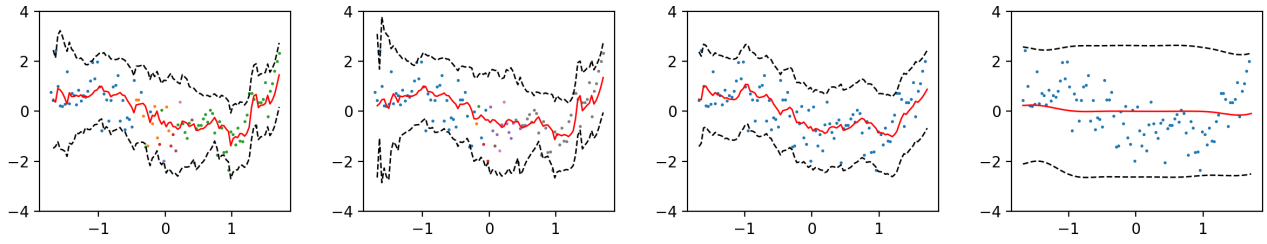
Table 2: One-step predictive log likelihood. One standard error reported in parentheses

	TP-MOE	GP-MOE	WISKI	OSVGP
Single GP	-119.708 (2.102)	-124.583	**-114.478 (0.000)**	-145.713 (0.284)
TP Mixture	**-106.678 (2.623)**	-109.747 (5.424)	-155.905 (0.000)	-144.818 (0.228)
Motorcycle	**-82.468 (13.027)**	-88.730 (12.529)	-112.468 (0.000)	-136.327 (0.175)
Brent	**113.067 (137.688)**	74.662 (109.869)	-800.856 (0.000)	-1412.110 (6.325)
DJI	**15.468 (5.717)**	11.449 (4.777)	-102.841 (0.000)	-156.046 (1.190)
Canada	208.464 (201.6)	**273.987 (87.699)**	-152.240 (0.000)	-295.574 (2.467)
Heart Rate	**-8399.773 (44.300)**	-8770.250 (75.635)	-10345.243 (0.000)	-13403.970 (219.706)
Nile	**-127.519 (2.026)**	-129.214 (2.673)	-127.997 (0.000)	-143.802 (0.274)
EUR-USD	**-4401.800 (33.108)**	-4463.015 (50.041)	-4482.187 (0.000)	-4653.289 (9.179)
Dow	-35.086 (0.421)	-36.544 (0.873)	**-34.225 (0.000)**	-35.523 (0.000)
Istanbul	**-628.442 (6.167)**	-632.165 (8.605)	-656.385 (0.000)	-798.133 (0.090)
Exchange	-153.770 (4.692)	-165.240 (6.031)	**-117.334 (0.000)**	-149.331 (0.000)
Wind	**-10167.675 (125.013)**	-10828.041 (79.021)	-10926.053 (0.000)	-16682.249 (1.000)
Power	**-12735.206 (215.951)**	-13391.495 (493.806)	-13508.126 (0.000)	-15882.683 (17.673)

Table 3: One-step predictive CRPS. One standard error reported in parentheses

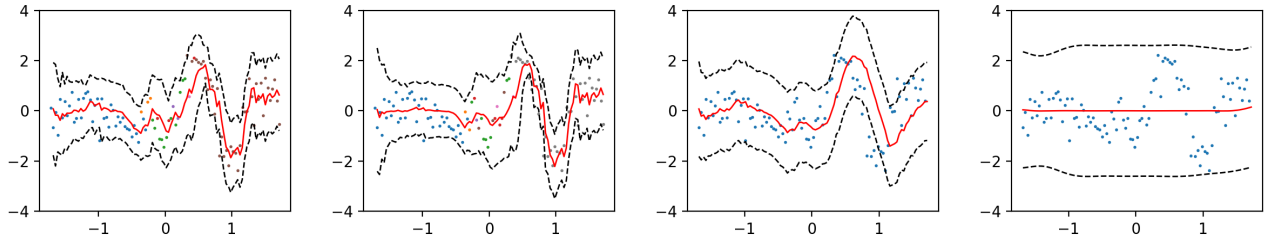
	TP-MOE	GP-MOE	WISKI	OSVGP
Single GP	0.428 (0.003)	0.435 (0.003)	**0.426 (0.000)**	0.573 (0.000)
TP Mixture	**0.382 (0.003)**	0.389 (0.013)	0.576 (0.000)	0.576 (0.001)
Motorcycle	**0.317 (0.014)**	0.324 (0.018)	0.431 (0.000)	0.562 (0.001)
Brent	**0.120 (0.015)**	0.125 (0.013)	0.253 (0.000)	0.530 (0.004)
DJI	**0.106 (0.003)**	0.110 (0.003)	0.249 (0.000)	0.517 (0.008)
Canada	**0.014 (0.002)**	**0.014 (0.002)**	0.208 (0.000)	0.519 (0.009)
Heart Rate	**0.319 (0.002)**	0.340 (0.003)	0.351 (0.000)	0.486 (0.009)
Nile	0.504 (0.006)	0.521 (0.011)	**0.490 (0.000)**	0.562 (0.002)
EUR-USD	0.548 (0.001)	**0.509 (0.008)**	0.547 (0.000)	0.569 (0.001)
Dow	0.538 (0.004)	0.555 (0.011)	**0.535 (0.000)**	0.549 (0.000)
Istanbul	**0.442 (0.003)**	0.444 (0.003)	0.457 (0.000)	0.569 (0.000)
Exchange	**0.366 (0.005)**	0.372 (0.005)	0.416 (0.000)	0.588 (0.000)
Wind	**0.340 (0.004)**	0.356 (0.003)	0.365 (0.000)	0.607 (0.000)
Power	**0.475 (0.009)**	0.496 (0.013)	0.501 (0.000)	0.594 (0.001)

WISKI cannot quantify the uncertainty as well as the MOE models due to its assumption of stationarity. For the Nile River data which exhibits only non-stationary mean values, the zero-mean assumption makes it hard for the online models to model the trend. Despite the model misspecification, the TP-MOE still achieves the best performance among the four models according to Tables 1, 2 and 3. Figure 10, in the supplementary materials, shows the TP-MOE's 95% predictive intervals are the most consistent with the trend. We posit that the heavy-tailed property helps the model more robust to the model misspecification.



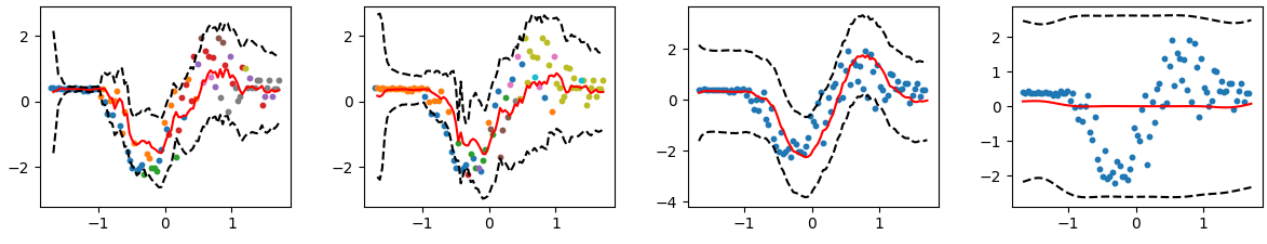
(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

Figure 1: Single GP Dataset. N = 100. Sample Runs.



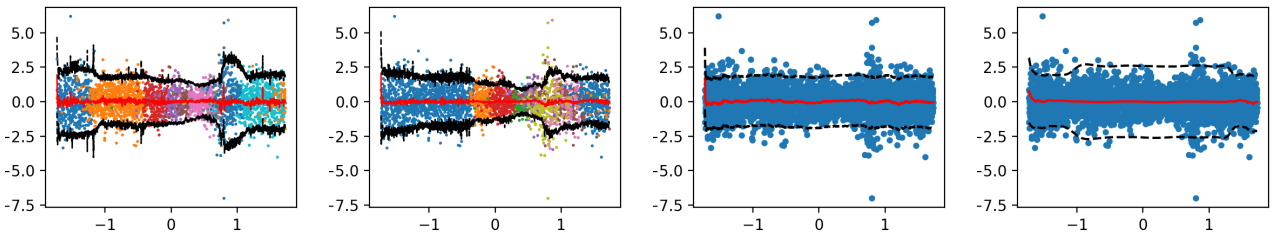
(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

Figure 2: Mixture of TPs. N = 100. Sample Runs



(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

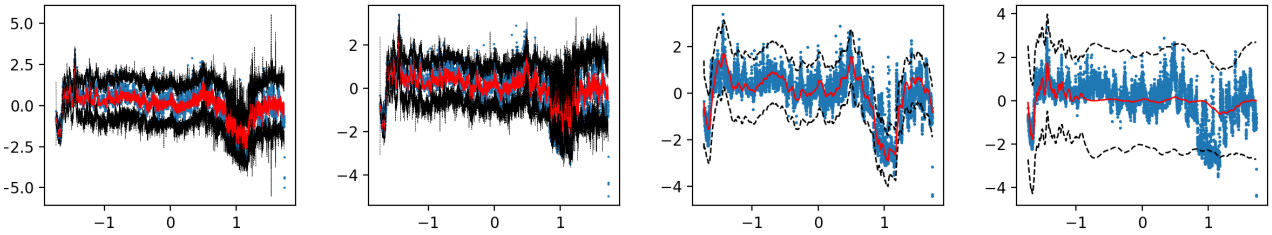
Figure 3: Motorcycle Dataset. N = 94. Sample Runs



(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

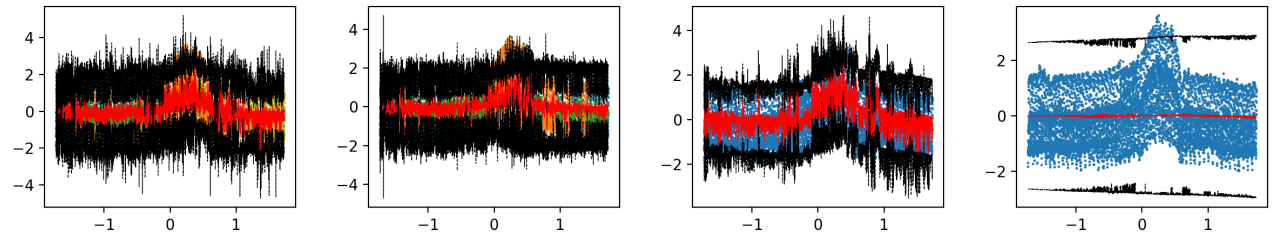
Figure 4: EUR-USD Dataset. N = 3139. Sample Runs.

However, when modelling the time-varying noise in the EUR-USD dataset, the GP-MOE handles this task the best. The OSVGP achieves good results this time because since the mean prediction should generally be zero. Although our method achieves a high predictive log-likelihood, it still does not outperform the GP-based



(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

Figure 5: Heart Rate Dataset. N = 10000. Sample Runs.



(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

Figure 6: Power Dataset. N = 10484. Sample Runs.

models. On the multi-dimensional input datasets (Dow - Power datasets), our TP-MOE still outperforms other GP-based models concerning all three metrics as in Tables 1, 2 and 3. In the sample runs plots (in the supplementary materials), we can also see that TP-MOE produces prediction intervals most consistent with the trend.

According to Table 4, the MOE models' CPU wall time is generally longer, where TP-MOE takes the longest time of around 1.5 times the GP-MOE takes, serving as a cost that comes along with better predictions. However, concerning one update, TP-MOE on average takes around 1 second for inference, which is still acceptable in real-life applications. If faster inference is a necessity, distributing the inference over more CPU cores can be an efficient way of reducing the time. We also discuss using TP-MOE for the Bayesian optimization problem in Appendix 8, serving as an example real-life application of it.

Table 4: One-step predictive CPU wall time. One standard error reported in parentheses

	TP-MOE	GP-MOE	WISKI	OSVGP
Single GP	103.020 (2.641)	68.087 (6.519)	**1.873 (0.401)**	3.471 (1.077)
TP Mixture	106.166 (16.063)	87.990 (16.782)	**1.825 (0.502)**	3.406 (1.053)
Motorcycle	109.515 (17.425)	56.984 (3.876)	**1.284 (0.125)**	2.272 (0.109)
Brent	1330.193 (68.456)	877.146 (140.541)	**13.294 (1.188)**	33.594 (15.869)
DJI	146.311 (15.179)	102.293 (12.125)	**1.314 (0.197)**	2.181 (0.116)
Canada	268.633 (8.341)	189.899 (15.737)	**2.582 (0.327)**	6.315 (2.062)
Heart Rate	14912.057 (1268.507)	9434.020 (1872.652)	254.921 (14.681)	**213.097 (1.424)**
Nile	109.489 (3.405)	81.590 (6.613)	**1.412 (0.006)**	2.605 (0.043)
EUR-USD	3824.995 (192.507)	3013.590 (363.891)	**36.838 (3.508)**	64.979 (3.009)
Dow	94.879 (7.113)	57.755 (3.014)	0.574 (0.030)	**0.516 (0.027)**
Istanbul	1040.887 (19.807)	874.755 (27.088)	14.788 (1.819)	**11.465 (1.240)**
Exchange	197.789 (1.429)	155.116 (13.344)	2.429 (0.057)	**2.247 (0.286)**
Wind	22359.280 (1754.292)	15241.125 (2568.025)	272.491 (11.710)	**239.733 (16.204)**
Power	23462.262 (1671.426)	16928.608 (2012.807)	253.122 (5.559)	**220.494 (2.925)**

5 CONCLUSION

Heavy-tailed data sets appear in a wide variety of applied settings. However, devising models that can adequately handle their noise structure is not trivial. In this paper, we build a Bayesian mixture of Student-t processes model with an overall-local scale structure for noisy data, which can be inferred by an SMC online algorithm. We have shown that TP-MOE has advantages over the Gaussian-based models when facing commonly encountered non-stationary datasets.

In future work, we are interested in applying the TP-MOE in financial prediction tasks. For such tasks, the learning, prediction, and decision-making aspects of the model occur in sparse, noisy environments that require heavy-tailed models in order for a learning agent to properly handle the problem at hand. Modelling data with a mixture of Student-t processes is a natural method for dealing with non-stationarity and heavy-tailed errors yet their popularity has still eluded the machine learning community. We seek to fill that gap with the method proposed in this paper.

Acknowledgements

We thank the reviewers for their constructive feedback comments. The contribution of Taole Sha was funded by the HKU Summer Research Fellowship. The work of Michael Minyi Zhang was supported by the HKU-URC Seed Fund for Basic Research for New Staff.

Bibliography

Aldous, D. (1985). Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII—1983*, pages 1–198.

Antoniak, C. E. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Annals of Statistics*, pages 1152–1174.

Bauer, M., van der Wilk, M., and Rasmussen, C. E. (2016). Understanding probabilistic sparse Gaussian process approximations. *Advances in Neural Information Processing Systems*, 29.

Bergstra, J., Bardenet, R., Bengio, Y., and Kégl, B. (2011). Algorithms for hyper-parameter optimization. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Bui, T. D., Nguyen, C., and Turner, R. E. (2017). Streaming sparse gaussian process approximations. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. (2013). SMC^2: An efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 75(3):397–426.

Cohen, S., Mbuvha, R., Marwala, T., and Deisenroth, M. (2020). Healing products of Gaussian process experts. In *International Conference on Machine Learning*, pages 2068–2077. PMLR.

Csató, L. and Opper, M. (2002). Sparse on-line Gaussian processes. *Neural Computation*, 14(3):641–668.

Damien, P., Wakefield, J., and Walker, S. (1999). Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(2):331–344.

Deisenroth, M. and Ng, J. W. (2015). Distributed Gaussian processes. In *International Conference on Machine Learning*, pages 1481–1490. PMLR.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436.

Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588.

Fernández, C. and Steel, M. F. (1999). Multivariate student-t regression models: Pitfalls and inference. *Biometrika*, 86(1):153–167.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 1(3):515–533.

Geweke, J. (1993). Bayesian treatment of the independent student-t linear model. *Journal of Applied Econometrics*, 8(S1):S19–S40.

Gramacy, R. B. and Polson, N. G. (2011). Particle learning of Gaussian process models for sequential design and optimization. *Journal of Computational and Graphical Statistics*, 20(1):102–118.

Härkönen, T., Wade, S., Law, K., and Roininen, L. (2022). Mixtures of Gaussian process experts with SMC^2. *arXiv preprint arXiv:2208.12830*.

Hutter, F., Hoos, H. H., and Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. In Coello, C. A. C., editor, *Learning and Intelligent Optimization*, pages 507–523, Berlin, Heidelberg. Springer Berlin Heidelberg.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L.-w. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Anthony Celi, L., and Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Juárez, M. A. and Steel, M. F. J. (2010). Model-based clustering of non-Gaussian panel data based on skew-t distributions. *Journal of Business & Economic Statistics*, 28(1):52–66.

Jylänki, P., Vanhatalo, J., and Vehtari, A. (2011). Robust Gaussian process regression with a student-t likelihood. *Journal of Machine Learning Research*, 12(11).

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., and Talwalkar, A. (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52.

Meeds, E. and Osindero, S. (2005). An alternative infinite mixture of Gaussian process experts. *Advances in Neural Information Processing Systems*, 18.

Minsker, S., Srivastava, S., Lin, L., and Dunson, D. (2014). Scalable and robust Bayesian inference via the median posterior. In Xing, E. P. and Jebara, T.,

editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1656–1664, Bejing, China. PMLR.

Murray, I., Adams, R., and MacKay, D. (2010). Elliptical slice sampling. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 541–548, Chia Laguna Resort, Sardinia, Italy. PMLR.

Neal, R. M. (2003). Slice sampling. *Annals of Statistics*, 31(3):705–767.

Nguyen-tuong, D., Peters, J., and Seeger, M. (2008). Local Gaussian process regression for real time online model learning. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc.

Rasmussen, C. and Ghahramani, Z. (2001). Infinite mixtures of Gaussian process experts. *Advances in Neural Information Processing Systems*, 14.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning*. The MIT Press.

Shah, A., Wilson, A., and Ghahramani, Z. (2014). Student-t Processes as Alternatives to Gaussian Processes. In Kaski, S. and Corander, J., editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 877–885, Reykjavik, Iceland. PMLR.

Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. *Advances in Neural Information Processing Systems*, 18:1259–1266.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Srinivas, N., Krause, A., Kakade, S., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 1015–1022.

Srivastava, S., Cevher, V., Dinh, Q., and Dunson, D. (2015). WASP: Scalable Bayes via barycenters of subset posteriors. In Lebanon, G. and Vishwanathan, S. V. N., editors, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 912–920, San Diego, California, USA. PMLR.

Stanton, S., Maddox, W., Delbridge, I., and Gordon Wilson, A. (2021). Kernel interpolation for scalable online Gaussian processes. In Banerjee, A. and Fukumizu, K., editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 3133–3141. PMLR.

Svensson, A., Dahlin, J., and Schön, T. B. (2015). Marginalizing Gaussian process hyperparameters using sequential Monte Carlo. In *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 477–480. IEEE.

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 Conferences on Artificial Intelligence*, pages 2822–2828.

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1(Jun):211–244.

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

Vanhatalo, J., Jylänki, P., and Vehtari, A. (2009). Gaussian process regression with student-t likelihood. *Advances in Neural Information Processing Systems*, 22.

West, M. (1984). Outlier models and prior distributions in Bayesian linear regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 46(3):431–439.

Wilson, A. and Nickisch, H. (2015). Kernel interpolation for scalable structured gaussian processes (kiss-gp). In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1775–1784, Lille, France. PMLR.

Zhang, M. M., Dumitrascu, B., Williamson, S. A., and Engelhardt, B. E. (2023). Sequential gaussian processes for online learning of nonstationary functions. *IEEE Transactions on Signal Processing*, 71:1539–1550.

Zhang, M. M. and Williamson, S. A. (2019). Embarrassingly parallel inference for Gaussian processes. *Journal of Machine Learning Research*, 20(169):1–26.

Zhang, Y. and Yeung, D. (2010). Multi-task learning using generalized t process. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence*

and Statistics, volume 9 of *Proceedings of Machine Learning Research*, pages 964–971, Chia Laguna Resort, Sardinia, Italy. PMLR.

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/No/Not Applicable]
   Yes, all the details about our model and inference algorithm are provided in the paper.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes/No/Not Applicable]
   Yes, see Section 3.1 which details our algorithm.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable]
   Yes, codes are provided in the supplementary material.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/Not Applicable]
   Yes, all the derivations are clearly shown.
   (b) Complete proofs of all theoretical results. [Yes/No/Not Applicable]
   Yes, for the derivation of formulas, details can be found in the main paper and the supplementary material.
   (c) Clear explanations of any assumptions. [Yes/No/Not Applicable]
   Yes, our model assumption is well explained in Section 3.

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/No/Not Applicable]
   Yes, codes and data are submitted in the supplementary material.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes/No/Not Applicable]
   Yes, in Section 4 of the paper we include all experiments details.
   (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/No/Not Applicable]
   Yes, in Section 4 about experiments we introduce the way we do our experiments (one-step ahead prediction) and the metrics (MSE, log-likelihood, CRPS) we choose.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/No/Not Applicable]
   Yes, we introduce how we perform parallel computing in Section 4.

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/No/Not Applicable]
   Yes, we have included proper citations when using datasets and models for comparison.
   (b) The license information of the assets, if applicable. [Yes/No/Not Applicable]
   Not Applicable, no license information.
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/Not Applicable]
   Not Applicable, we have no new assets.
   (d) Information about consent from data providers/curators. [Yes/No/Not Applicable]
   Not Applicable, the data are open source datasets.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/Not Applicable]
   Not Applicable, no sensible 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. [Yes/No/Not Applicable]
   Not Applicable, no human subjects are included.
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/Not Applicable]
   Not Applicable, no participation of humans is included
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/Not Applicable]
   Not Applicable, no participants are included.

Online Student-t Processes with an Overall-local Scale Structure for Modelling Non-stationary Data Supplementary Materials

6 STUDENT-t PROCESS FOR NOISY DATA

In our model as introduced in Section 3 of the main paper, we handle noisy data using an additional heteroscedastic parameter. Consider the observations $(\mathbf{X}_k, \mathbf{y}_k)$ belonging to cluster k, it is assumed that $\mathbf{y}_k = f(\mathbf{X}_k) + \sigma_k \boldsymbol{\epsilon}_k$. The output is generated by a latent zero-mean Gaussian Process $f(\mathbf{X}_k)$ and a Gaussian noise term $\boldsymbol{\epsilon}_k \sim \mathcal{N}_{N_k}(0, |h_k|\mathbf{I})$. σ_k^2 is an overall scale parameter for both the covariance function $\sigma^2 \mathbf{K}_{\boldsymbol{\theta}_k}$ and the noise term $\sigma_k \boldsymbol{\epsilon}_k$, where $\boldsymbol{\theta}_k$ is the kernel parameter for the kernel $\mathbf{K}$. Additionally $h_k \in \mathbb{R}$ and $|h_k|$ is a scale parameter for the noise term to control the heteroscedasticity, and $\mathbf{I}$ is an identity matrix. The GP and the noise are not independent here since they share the same overall scale. The latent function $f(\mathbf{X}_k)$ is from the a GP evaluated at locations $\mathbf{X}_k$, which follows a multivariate normal distribution:

$$f(\mathbf{X}_k)|\mathbf{X}_k, \sigma_k^2, \boldsymbol{\theta}_k \sim \mathcal{N}_{N_k}(0, \sigma_k^2 \mathbf{K}_{\boldsymbol{\theta}_k}). \tag{14}$$

For noisy observations $\mathbf{y}_k$, the data is generated by:

$$\mathbf{y}_k|f(\mathbf{X}_k), \sigma_k^2, h_k \sim \mathcal{N}_{N_k}(f(\mathbf{X}_k), \sigma_k^2|h_k|\mathbf{I}). \tag{15}$$

Then marginally:

$$\begin{aligned}\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}_k, \sigma_k^2, h_k &= \int P(\mathbf{y}_k|f(\mathbf{X}_k), \sigma_k^2, h_k) P(f(\mathbf{X}_k)|\mathbf{X}_k, \sigma_k^2, \boldsymbol{\theta}_k) df(\mathbf{X}_k) \\ &\sim \mathcal{N}_{N_k}(0, \sigma_k^2(\mathbf{K}_{\boldsymbol{\theta}_k} + |h_k|\mathbf{I})).\end{aligned} \tag{16}$$

By marginalizing an inverse Gamma prior on σ_k^2 out, we can arrive at the target multivariate Student-t distribution with degree of freedom ν_k:

$$\begin{aligned}\sigma_k^2|\nu_k &\sim \text{Inv-Gamma}\left(\frac{\nu_k}{2}, \frac{\nu_k}{2}\right), \\ \mathbf{y}_k|\mathbf{X}_k, \boldsymbol{\theta}_k, h_k, \nu_k &= \int P(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}_k, \sigma_k^2, h_k) P(\sigma_k^2|\nu_k) d\sigma_k^2 \\ &\sim \mathcal{T}_{N_k}(\nu_k, 0, \mathbf{K}_{\boldsymbol{\theta}_k} + |h_k|\mathbf{I}).\end{aligned} \tag{17}$$

The TP's log marginal likelihood of cluster k is:

$$\begin{aligned}\log P(\mathbf{y}_k|\mathbf{X}_k, \boldsymbol{\theta}_k, h_k, \nu_k) = &-\frac{N_k}{2}\log(\nu_k \pi) - \frac{1}{2}\log(|\mathbf{K}_{\boldsymbol{\theta}_k} + |h_k|\mathbf{I}|) + \log\left(\frac{\Gamma(\frac{\nu_k + N_k}{2})}{\Gamma(\frac{\nu_k}{2})}\right) \\ &-\frac{\nu_k + N_k}{2}\log\left(1 + \frac{\mathbf{y}_k^T(\mathbf{K}_{\boldsymbol{\theta}_k} + |h_k|\mathbf{I})^{-1}\mathbf{y}_k}{\nu_k}\right).\end{aligned} \tag{18}$$

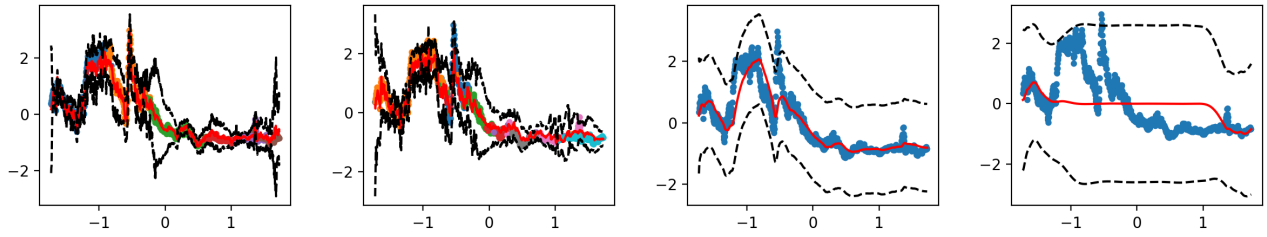
When making predictions of N^* target outputs $\mathbf{y}^*$ given new inputs $\mathbf{X}^*$, denote the kernel matrix evaluated between $\mathbf{X}_k$ and $\mathbf{X}^*$ as $\mathbf{K}'_{\boldsymbol{\theta}_k} \in \mathbb{R}^{N^* \times N_k}$, and the kernel matrix evaluated at $\mathbf{X}^*$ as $\mathbf{K}^*_{\boldsymbol{\theta}_k} \in \mathbb{R}^{N^* \times N^*}$, the posterior predictive distribution is:

$$\begin{aligned}&\tilde{\phi}_2 = \mathbf{K}'_{\boldsymbol{\theta}_k}(\mathbf{K}_{\boldsymbol{\theta}_k} + |h_k|\mathbf{I})^{-1}\mathbf{y}, \quad \beta_1 = \mathbf{y}^T(\mathbf{K}_{\boldsymbol{\theta}_k} + |h_k|\mathbf{I})^{-1}\mathbf{y}, \\ &\tilde{\mathbf{K}}_{22} = (\mathbf{K}^*_{\boldsymbol{\theta}_k} + |h_k|\mathbf{I}) - \mathbf{K}'_{\boldsymbol{\theta}_k}(\mathbf{K}_{\boldsymbol{\theta}_k} + |h_k|\mathbf{I})^{-1}\left(\mathbf{K}'_{\boldsymbol{\theta}_k}\right)^T, \\ &\mathbf{y}^*|\mathbf{y}, \mathbf{X}_k, \mathbf{X}^* \sim \mathcal{T}_{N^*}\left(\nu_k + N_k, \tilde{\phi}_2, \frac{\nu_k + \beta_1}{\nu_k + N_k}\tilde{\mathbf{K}}_{22}\right),\end{aligned} \tag{19}$$

where the quantities $\tilde{\phi}_2, \beta_1, \tilde{\mathbf{K}}_{22}$ are not of interest, but intermediate compuatation steps for clearer expression.

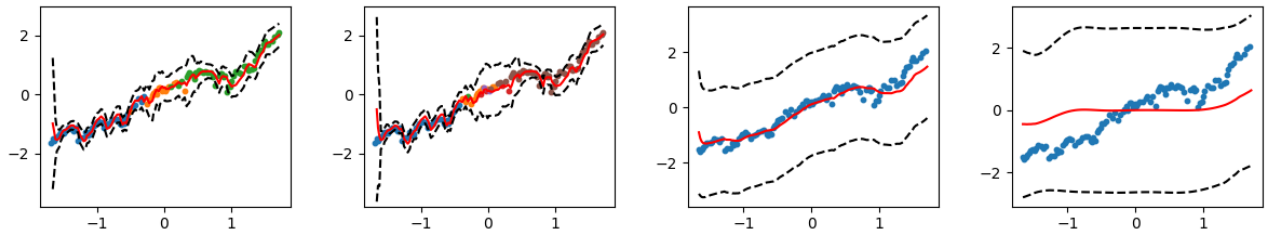
7 PLOTS OF SAMPLE RUNS

The plots of other sample runs (Figures 7-14) on the datasets mentioned in Section 4 of the main paper are listed here, which are additional evidences of our statements. **The plots contain the data points, one-step predictive mean (plotted with solid red lines) and 95% predictive interval (plotted with dashed black lines). The data points in MOE models' plots are coloured according to the cluster assignment given by the particle with the highest weight**



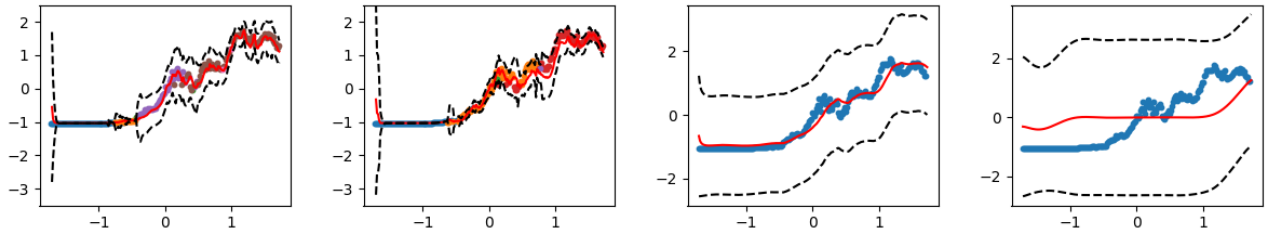
(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

Figure 7: Brent Dataset. N = 1025. Sample Runs



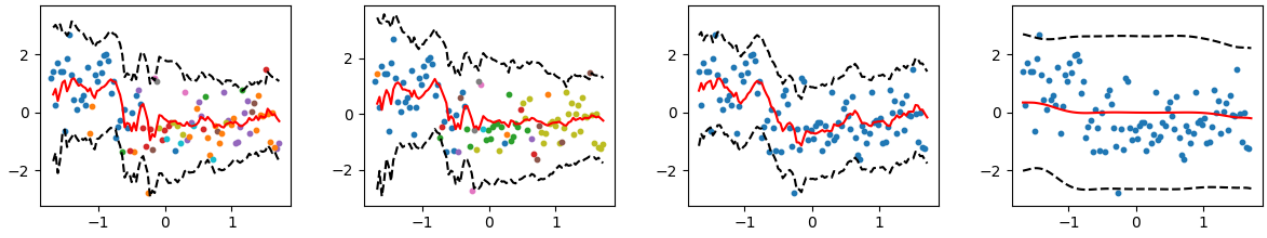
(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

Figure 8: DJI Dataset. N = 112. Sample Runs.



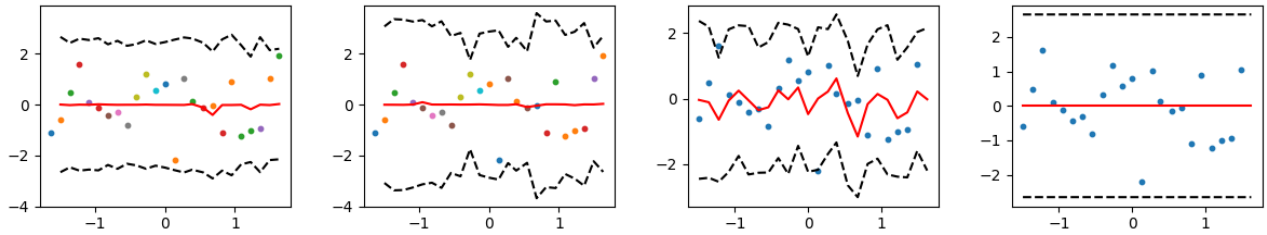
(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

Figure 9: Canada Dataset. N = 215. Sample Runs



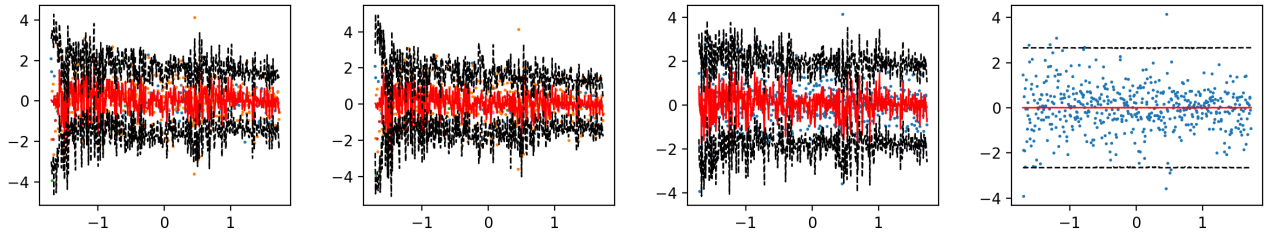
(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

Figure 10: Nile River Dataset. N = 100. Sample Runs



(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

Figure 11: Dow Dataset. N = 25. Sample Runs

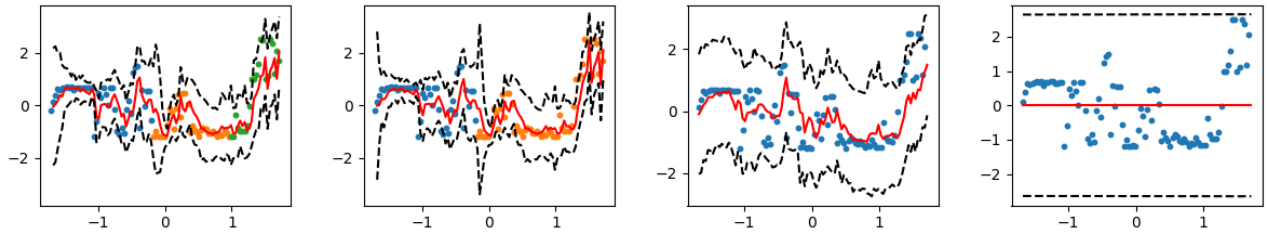


(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

Figure 12: Istanbul Dataset. N = 536. Sample Runs

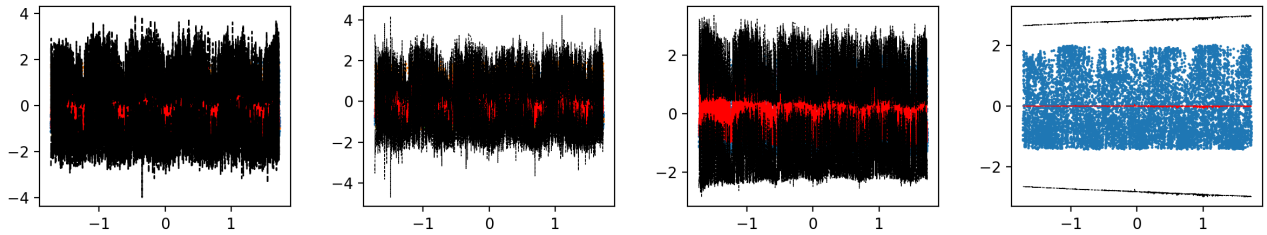
8 ONLINE TP-MOE FOR OPTIMIZATION

Optimization techniques are quite useful to find the parameters that best model the data. Popular approaches include (Hutter et al., 2011; Bergstra et al., 2011; Snoek et al., 2012), which adaptively do the optimization. In GP/TP contexts, bandit formulations are celebrated (Srinivas et al., 2010; Li et al., 2018), where the objective is to optimize a function f. Treating f to be sampled from a GP/TP, a bandit algorithm will sequentially pick up arms (input $\mathbf{x}_i$) that maximize the rewards, where noisy output y_i will be observed. When selecting the arms, in the exploration step the rewards of each arm are approximated, according to which in the exploitation step we pick up arms that reach the maximum. A sampling-based algorithm for it is named Thompson sampling (TS), where samples from the posterior predictive distribution at $\mathbf{x}_1^*, \mathbf{x}_2^*, \cdots, \mathbf{x}_{N_{sample}}^*$ are taken and the test input at



(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

Figure 13: Exchange Rates Dataset. N = 101. Sample Runs



(a) TP-MOE (b) GP-MOE (c) WISKI (d) OSVGP

Figure 14: Wind Dataset. N = 10950. Sample Runs

which the highest y^* is evaluated is selected:

$$\begin{aligned} &y_n^* \sim P(y_n^*|\mathbf{X}, \mathbf{y}, \mathbf{x}_n^*), \ n = 1, 2, \cdots, N_{sample}, \\ &\mathbf{x}_i := \mathbf{x}_{\text{argmax}_n y_n^*}. \end{aligned} \tag{20}$$

The selection is based on the maximum sample values achieved by each arm, high predictive mean and high uncertainty are naturally preferred, which balances exploration and exploitation.

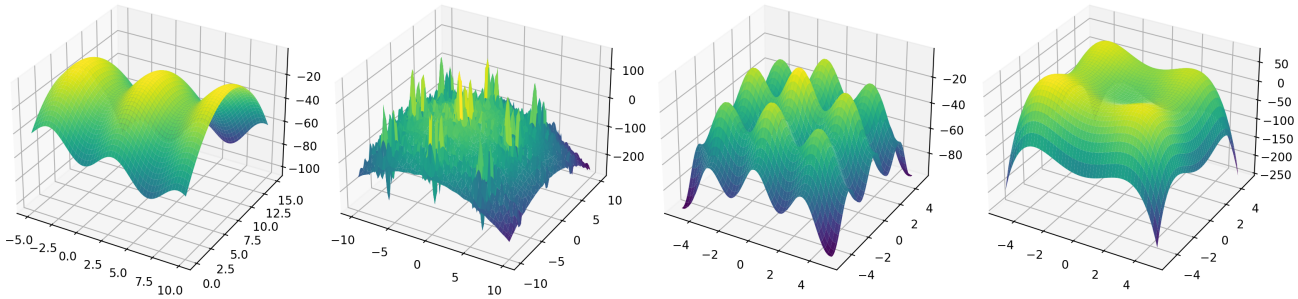
Since the function f is typically hard to evaluate and therefore it's not possible to observe large enough data. The hyperparameters are not known as a priori and are unlikely to be accurately estimated right after fitting the model. Hence updating the models when new function queries arrive is an attractive and necessary choice for the above Bayesian optimization procedure, where our TP-MOE is appealing. To optimize a function f using the online TP-MOE, firstly a new test input $\mathbf{x}_i$ is selected following the strategy in (21), then the model is updated with $(\mathbf{x}_i, f(\mathbf{x}_i))$ by Algorithm 1 in the main paper. Repeating it for N evaluations leads to Algorithm 3

Algorithm 3: TP-MOE Bandit Optimization with Thompson Sampling

for $i = 1, \cdots, N$ **do**
  Sample points from: $y_n^* \sim P(y_n^*|\mathbf{X}, \mathbf{y}, \mathbf{x}_n^*), \ n = 1, 2, \cdots, N_{sample}$ by Algorithm 2 in main paper.
  Select the point $\mathbf{x}_i := \mathbf{x}_{\text{argmax}_n y_n^*}$.
  Update model with $(\mathbf{x}_i, f(\mathbf{x}_i))$ by Algorithm 1 in main paper.
end

8.1 Bandit Optimization

We also implement our TP-MOE and other GP-based models in a Bandit optimization application using Thompson sampling as shown in Algorithm 3. Four test functions are tried, whose plots can be found in Figure 15.



(a) Branin-Hoo (b) Levy 5 (c) Egg Crate (d) Styblinski-Tang

Figure 15: Test functions for the bandit optimization Experiments

They are commonly used to evaluate optimization algorithms, namely the Branin-Hoo function, the Levy 5 function, the Rastrigin function and the Styblinski-Tang function. As suggested by Zhang et al. (2023), the particle numbers of TP-MOE and GP-MOE are reset to be 10 to achieve a balance between performance and speed. The number of inducing points is still 50, and we run 500 iterations for each model on each test function. Two metrics are adopted to evaluate the results, where less mean absolute regret (MAR) and larger maximum function value obtained $\max_{\mathbf{x}} f(\mathbf{x})$ is preferred.

For the Branin-Hoo function and the Styblinski-Tang where the trend is smooth and simple, a stationary GP model is enough and the WISKI achieves good results regarding $\max_{\mathbf{x}} f(\mathbf{x})$ and MAR as in Table 5. However, when facing more complex scenarios like the Levy 5 function and the Egg Crate function, using MOE models becomes a necessity. From Table 5 we can see that MOE models achieve larger $\max_{\mathbf{x}} f(\mathbf{x})$, showing the modelling flexibility. Although WISKI still achieves low MAR, since it is unable to find a comparable $\max_{\mathbf{x}} f(\mathbf{x})$, we conclude that the MAR results are due to its poor exploration of the function. Focusing on the MOE models, our TP-MOE can generally achieve lower MAR than GP-MOE, implying that it could capture the function trend faster, and fewer iterations are needed for it to converge.

Table 5: Maximum function value. One standard error reported in parentheses

	Maximum function value				Mean absolute regret			
	TP-MOE	GP-MOE	WISKI	OSVGP	TP-MOE	GP-MOE	WISKI	OSVGP
Branin-Hoo	-0.412 (0.012)	-0.424 (0.029)	**-0.408 (0.009)**	-0.524 (0.120)	16.651 (6.840)	20.551 (6.392)	**5.098 (0.215)**	29.758 (0.940)
Levy 5	134.636 (27.243)	**138.336 (6.001)**	124.761 (33.247)	106.498 (29.498)	**28.625 (1.874)**	55.628 (6.001)	28.903 (2.467)	71.863 (2.190)
Egg Crate	**-0.248 (0.255)**	-0.864 (1.077)	-3.201 (4.321)	-1.492 (1.555)	33.289 (7.236)	33.365 (8.137)	**12.428 (0.639)**	43.291 (0.811)
Styblinski-Tang	78.068 (0.414)	78.048 (0.362)	**78.099 (0.601)**	77.217 (1.211)	**6.706 (6.793)**	7.339 (1.722)	6.755 (0.475)	37.197 (1.203)