Fixing the Pitfalls of Probabilistic Time-Series Forecasting Evaluation by Kernel Quadrature

Masaki Adachi* *Lattice Lab, Toyota Motor Corporation / Machine Learning Research Group, University of Oxford*
Masahiro Fujisawa* *The University of Osaka / Lattice Lab, Toyota Motor Corporation / RIKEN AIP*
Michael A. Osborne *Machine Learning Research Group, University of Oxford*

Abstract

Despite the significance of probabilistic time-series forecasting models, their evaluation metrics often involve intractable integrations. The most widely used metric, the continuous ranked probability score (CRPS), is a strictly proper scoring function; however, its computation requires approximation. We found that popular CRPS estimators—specifically, the quantile-based estimator implemented in the widely used GluonTS library and the probability-weighted moment approximation—both exhibit inherent estimation biases. These biases lead to crude approximations, resulting in improper rankings of forecasting model performance when CRPS values are close. To address this issue, we introduce a kernel quadrature approach that leverages an unbiased CRPS estimator and employs cubature construction for scalable computation. Empirically, our approach consistently outperforms the two widely used CRPS estimators.

Proceedings of the 1st International Conference on Probabilistic Numerics (ProbNum) 2025,
Sophia Antipolis, France. PMLR Volume 271. Copyright 2025 with the author(s)

1 Introduction

Time-series forecasting plays a central role in various applications, including finance (Kim, 2003; Sezer et al., 2020), healthcare (Bui et al., 2018; Morid et al., 2023), and renewable energy (Wang et al., 2019; Dumas et al., 2022; Adachi et al., 2023). Perfect prediction is inherently unattainable due to the difficulty of forecasting the future. Consequently, probabilistic modeling is often employed—not only to improve point prediction accuracy but also to capture the predictive distribution.

A wide range of probabilistic models has been proposed, including ARIMA (Box and Jenkins, 1970), Gaussian processes (GPs; Roberts et al. (2013)), and deep learning-based models (Dumas et al., 2022; Oskarsson et al., 2024). The key question is how to properly evaluate the accuracy of predictive distribution inference, rather than relying solely on point prediction metrics such as mean squared error. Since the observed value at time t is an instantiation of an underlying random variable, the goal is to match the true generative distribution rather than overfitting to the observed test point.

Thus, researchers have long sought better metrics for probabilistic forecasting (Matheson and Winkler, 1976; Hersbach, 2000), and there is now a consensus that the desired scoring rule should be *strictly proper* (Gneiting and Raftery, 2007). This condition ensures that the expected score is minimized when the predicted distribution matches the true generative distribution. Several strictly proper scoring rules exist, such as the Brier score (Brier, 1950), but the Continuous Ranked Probability Score (CRPS; Matheson and Winkler (1976)) has gained particular popularity in the modern machine learning community (Alexandrov et al., 2020; Kollovieh et al., 2024; Tóth et al., 2024). CRPS has a closed-form solution for commonly used parametric distributions, such as Gaussian and logistic distributions, making it especially suitable for evaluating GP models.

However, deep learning models typically do not rely on classical parametric distributions, necessitating the approximation of CRPS via sampling from the predictive distribution. The current sampling-based approach relies on a grid search over the quantile space, but we identify issues in its estimation bias. In particular, while we expect sample-based estimators to exhibit asymptotic convergence behavior, we demonstrate that a persistent bias exists between the true CRPS and its approximation—one that remains even with an infinite number of samples when the grid size is fixed.

To address this, we propose kernel quadrature as a principled approach to improving finite-sample estimators. We show that our method converges to the true CRPS faster than popular CRPS estimators while remaining free from estimation bias. Although any unbiased estimator can correct this, our kernel quadrature method further reduces the quadratic complexity to linear.

2 Problem setting

Let $\mathbf{x}_{0:L} = (\mathbf{x}_0, \cdots, \mathbf{x}_L) \in \mathrm{Seq}(\mathbb{R}^d)$ be an input time-series and $\mathbf{y}_{0:L} = (\mathbf{y}_0, \cdots, \mathbf{y}_L) \in \mathrm{Seq}(\mathbb{R})$ be a univariate output time series[1]. We assume a latent variable models $p_\theta(\mathbf{y}_{0:L}) = \int p_\theta(\mathbf{y}_{0:L}, \mathbf{x}_{0:L})\mathrm{d}\mathbf{x}$, where $\mathbf{y}_{0:L} \sim p(\mathbf{y}_{0:L})$ represents the true underlying distribution. We define the training dataset as $\mathbf{D}_{0:L} = (\mathbf{x}_{0:L}, \mathbf{y}_{0:L})$ and the test dataset as $\mathbf{D}_{L+1:L+T} = (\mathbf{x}_{L+1:L+T}, \mathbf{y}_{L+1:L+T})$, where L and T denote the training and test sizes, respectively.

We consider a set of candidate autoregressive models[2], $f^{(i)}(x) = p_\theta(y \mid x, \mathbf{D}_{0:L})$, where different models $f^{(i)}$

[1] We can extend the multivariate output, e.g., via multi-output GPs, but we describe only the univariate case.

[2] The model is conditioned on the previous state; see Tóth et al. (2024) for details.

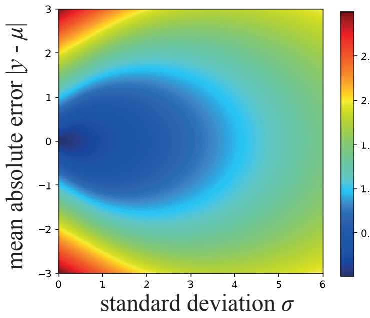

Figure 1: The exact CRPS for given Gaussian parameters.

and $f^{(j)}$ are not merely distinguished by parameterization θ but belong to entirely different function classes (e.g., Gaussian processes and diffusion models). Our goal is to identify the most plausible model given a collection of datasets $\{\mathbf{D}^{(k)}_{0:L_k+T_k}\}_{k=1}^{K}$.

At test time, for each dataset $\mathbf{D}^{(k)}$, we have access only to the test input sequence $\mathbf{x}_{L_k+1:L_k+T_k}$, where the dataset size $|\mathbf{D}^{(k)}| = L_k + T_k$ varies across datasets. Given a probabilistic prediction $f^{(i)}(x_l)$ and the hidden ground truth $\mathbf{y}_l$ at the l-th timestep, we evaluate the performance of the i-th model using the CRPS metric:

$$\mathrm{CRPS}(F, y_l) := \int_{-\infty}^{\infty} \left(F(y') - \mathbb{1}_{y'<y}\right)^2 \mathrm{d}y', \qquad (1)$$

where $F(y)$ is the cumulative density function (CDF) of the probability distribution of y, and $\mathbb{1}_{y'<y}$ is an indicator function that returns 1 if the condition $y' < y$ is satisfied and 0 otherwise.

It is common to use the sample average of CRPS as the scoring rule: $S(f^{(i)} \mid \mathbf{D}^{(k)}, r_j) = {}^{1}\!/_{T_k - L_k} \sum_{l=L_k}^{T_k} \mathrm{CRPS}(F^{(k)}, y_l)$, where $r_j \in \mathcal{R}$ is the j-th random seed for computation, sampled uniformly from $\mathcal{U}(\mathcal{R})$. To assess the effect of random seeds, we further iterate this scoring procedure by re-training the model $f^{(i)}$ with different random seeds r_j. The final evaluation metrics reported in the paper are the sample mean: $\overline{S} = \mathbb{E}_{r_j \sim \mathcal{U}(\mathcal{R})}[S(f^{(i)} \mid \mathbf{D}^{(k)}, r_i)]$ and the variance: $\mathbb{V}_{r_j \sim \mathcal{U}(\mathcal{R})}[S(f^{(i)} \mid \mathbf{D}^{(k)}, r_i)]$. The performance ranking of the proposed model $f^{(i)}$ is primarily based on the expectation $\overline{S}$. As such, the goal of this task is to minimize the following integral approximation error:

$$\left|\mathbb{E}_{r_j \sim \mathcal{U}(\mathcal{R})}[S(f^{(i)} \mid \mathbf{D}^{(k)}, r_i)] - \mathbb{E}_{r_i \sim \mathcal{U}(\mathcal{R})}[\hat{S}(f^{(i)} \mid \mathbf{D}^{(k)}, r_i)]\right|,$$

where $\hat{S}$ denotes the approximated sample-mean CRPS[3].

2.1 Known results and approximation

Exact CRPS with Gaussian CRPS has a closed-form solution for a univariate Gaussian predictive distribution, where $y_l \sim \mathcal{N}(m_l, \sigma_l^2)$ at the l-th timestep (see Eq. (5) in Gneiting et al. (2005)). Let $z_l := {}^{y_l - m_l}\!/_{\sigma_l}$ be the standardized output, then we have:

$$\mathrm{CRPS}(F, y_l) = \sigma_l \left[z_l(2\Phi(z_l) - 1) + 2\phi(z_l) - \frac{1}{\sqrt{\pi}} \right], \qquad (2)$$

where $\Phi(\cdot)$ and $\phi(\cdot)$ denote the CDF and probability density function (PDF) of the standard normal distribution $\mathcal{N}(0, 1)$, respectively. This closed-form expression offers us to directly compute the exact CRPS for GP models. Similarly, closed-form solutions are available for popular parametric distributions (see Appendix B in Taillardat et al. (2016) for a complete list)[4].

Eq.(2) provides an intuitive understanding of CRPS. Since the variance σ_l is a multiplicative factor in all terms, a smaller predictive variance leads to a lower CRPS. Additionally, the expression inside the brackets is convex with respect to the standardized output z_l, attaining its minimum at $z_l = 0$. This implies that a lower mean absolute error, $|y_l - m_l|$, results in a better score. Fig.1 illustrates this intuition. Thus, CRPS serves as a reasonable scoring rule for evaluating both predictive accuracy and the tightness of predictive variance.

Approximating CRPS with quantile loss. Exact computation of CRPS is not always feasible for all predictive models. In particular, deep learning-based probabilistic forecasting models, such as diffusion models, do not have closed-form predictive distributions. Thus, CRPS must be estimated from i.i.d. function samples.

There are two common approaches for sample-based CRPS approximation. The most widely used method is the quantile loss reformulation (Kollovieh et al., 2024):

$$\mathrm{CRPS}(F, y_l) = \int_0^1 2\Lambda_\kappa(F^{-1}(\kappa), y_l)\mathrm{d}\kappa, \qquad (3)$$

where F^{-1} is the quantile function (also known as the inverse CDF), and $\Lambda_\kappa(q, y) = (\kappa - \mathbb{1}_{y<q})(y - q)$ represents the pinball loss for a given quantile level κ. To approximate the quantile function, we typically use the empirical CDF (Dekking et al., 2006).

The estimation procedure consists of two steps: First, we draw M i.i.d. function samples at the test input time series, $\mathbf{f}_l = \{f_m^{(i)}(x_l)\}_{m=1}^{M}$, and then estimate the empirical CDF, $\hat{F}(y) = {}^{1}\!/_{M} \sum_{m=1}^{M} \mathbb{1}_{f_m^{(i)}(x_l) \leq y}$. Next, we discretize the quantile levels $\kappa_\ell \in \mathcal{K}$ using a finite set, $\mathcal{K} = (\kappa_1, ..., \kappa_Q) = ({}^{1}\!/_{2Q}, ..., {}^{2Q-1}\!/_{2Q})$. Using this discretization, we approximate the CRPS in Eq. (1) as:

$$\widehat{\mathrm{CRPS}}(\hat{F}, y_l) = \frac{1}{Q} \sum_{\kappa_\ell \in \mathcal{K}} 2\Lambda_{\kappa_\ell}(\hat{F}^{-1}(\kappa_\ell), y_l). \qquad (4)$$

Due to the high computational cost of deep learning models, the sample sizes for all approximation steps are limited (Kollovieh et al., 2024; Tóth et al., 2024). For the empirical CDF, the number of function samples is typically set to $M = 100$, and the quantile levels are uniformly discretized into nine points, $\mathcal{K} =$

[3] GP has a closed-form CRPS, making this error zero. However, for a fair comparison with deep learning models, it is common to evaluate GP using an approximated CRPS obtained via sampling.

[4] The closed-form CDF is limited to univariate distributions. Thus, multivariate time series require approximation.

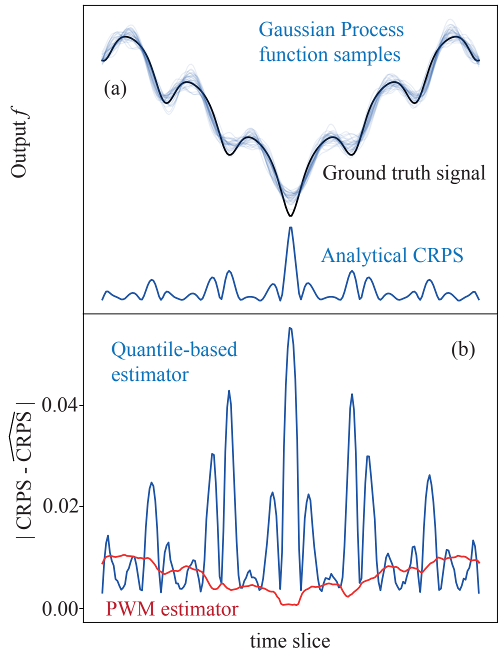

Figure 2: Illustrative example: (a) Ackley function fitted by a GP with its function samples and the analytical CRPS computed using Eq.(2), (b) slicewise CRPS estimation error for the quantile-based estimator (Eq.(3)) and the probability-weighted moment (PWM) estimator (Eq. (5)).

$(0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9)$. Also, due to the computational burden of retraining deep learning models, the number of random seeds is limited to $|\mathcal{R}| = 3$[5].

Approximating the CRPS with the PWM. Another widely used approximation is the probability-weighted moment (PWM; Taillardat et al. (2016)):

$$\begin{aligned}&\mathrm{CRPS}(F, y_l)\\ &= \underbrace{\mathbb{E}_{y\sim\mathbb{P}(f|x_l)}[|y-y_l|]}_{\text{error term}} + \underbrace{\mathbb{E}_{y\sim\mathbb{P}(f|x_l)}[y]}_{\text{mean term}} - \underbrace{2\mathbb{E}_{y\sim\mathbb{P}(f|x_l)}[yF(y)]}_{\text{CDF term}}.\end{aligned} \tag{5}$$

The advantage of this approach is that it simplifies CRPS estimation into a straightforward Monte Carlo (MC) integration. For a Gaussian predictive distribution, each term has a closed-form expression:

$$\begin{aligned}\mathbb{E}_{y\sim\mathbb{P}(f|x_l)}[|y-y_l|] &= \sigma_l[z_l(2\Phi(z_l)-1)+2\phi(z_l)],\\ \mathbb{E}_{y\sim\mathbb{P}(f|x_l)}[y] &= \mu_l,\\ \mathbb{E}_{y\sim\mathbb{P}(f|x_l)}[yF(y)] &= \frac{1}{2}\left(\mu_l + \frac{\sigma_l}{\sqrt{\pi}}\right).\end{aligned} \tag{6}$$

As such, this approximation equals to Eq. (2).

3 Pitfalls of CRPS approximation

3.1 Quantile or PWM?

Given the two approximation methods, a natural question arises: which one should we use for evaluating time-series models? Due to the popularity of the GluonTS library (Alexandrov et al., 2020), the quantile-based estimator has dominated recent publications. However, we argue that this approach falls into hidden pitfalls. To illustrate these pitfalls, we first analyze a toy example to understand the typical behavior of CRPS approximation and identify the sources of evaluation bias.

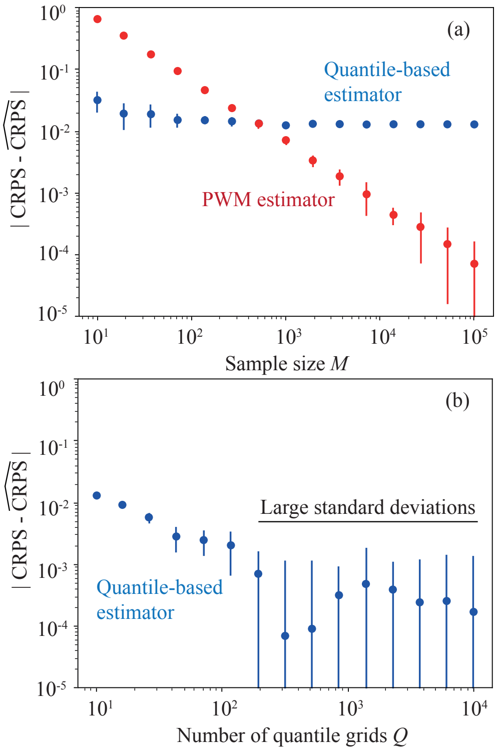

Figure 3: Convergence rate analysis (mean ± 1 standard deviation over 10 random seeds). (a) While the PWM estimator converges with respect to the sample size M, the quantile-based estimator does not for a fixed $Q = 9$. (b) The quantile-based estimator exhibits convergence with respect to the number of quantile grids Q, but it plateaus around $Q = 100$ for $M = 10^2$.

Fig. 2 explains the set up: we use the Ackley function as the test function (Ackley, 1987) and a GP time-series model as the forecasting model. For simplicity, we randomly sample nine points from the domain and fit a GP model to these data points. We then draw M function samples over $T = 200$ test points. Since the GP predictive distribution is Gaussian, we compute the analytical CRPS using Eq. (2) (see Fig. 2(a)). Next, we compare the two approximation methods—quantile-based and PWM estimators—based on M samples. Fig. 2(b) shows the estimation error across the domain. Notably, the error of the quantile-based estimator closely follows the shape of the analytical CRPS, whereas the error of the PWM estimator roughly follows the predictive mean of the GP model. Ideally, an unbiased estimator should exhibit no systematic pattern over time. This result suggests that both the quantile and PWM estimators introduce estimation bias.

The bias issue becomes more evident when we examine the convergence rate with respect to the sample size M in Fig. 3(a). While the PWM estimator exhibits asymptotic convergence, the quantile-based estimator plateaus, indicating clear estimation bias. Interestingly, in the small sample size regime ($M < 10^3$), the quantile-based estimator shows lower errors, but this advantage disappears at larger sample sizes.

[5] We note that the quantile prediction approach (Cai, 2002) can accelerate quantile-based CRPS evaluation.

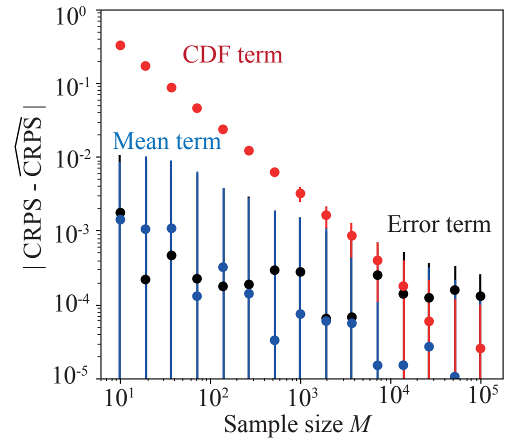

Figure 4: The decomposition of PWM estimation errors. CDF term dominates other two terms in the small sample regime.

This leads to the first pitfall: *spurious supremacy of quantile-based estimator*. This phenomenon misguides the community into adopting the current de facto standard setting (quantile estimator with $M = 100$ and $Q = 9$). Under this setting, the quantile method appears superior because its error is lower. Additionally, its stability under varying sample sizes reinforces the misconception that $M = 100$ is sufficiently large and that further sampling is unnecessary.

However, this plateau occurs because the default quantile grid size, $Q = 9$, is too coarse. Fig. 3 (b) confirms that increasing the grid size Q leads to asymptotic convergence. Hence, the quantile estimator requires a balance between grid size Q and sample size M.

From a computational perspective, the quantile estimator has complexity $\mathcal{O}(QM \log M)$, whereas the PWM estimator has a lower complexity of $\mathcal{O}(M \log M)$. Thus, achieving convergence with the quantile estimator is more computationally expensive. Ideally, we should prefer the simpler PWM estimator.

3.2 Why is PWM worse in small samples?

The key question is why the PWM estimator performs worse in the small sample size regime. The empirical convergence rate in Fig. 3(a) is approximately $\mathcal{O}({}^1\!/\!_M)$, which is faster than the well-known Monte Carlo (MC) integration rate of $\mathcal{O}({}^1\!/\!_{\sqrt{M}})$. This is unexpected because the second term of the PWM estimator in Eq. (5) is a pure MC integral, implying a convergence rate of $\mathcal{O}({}^1\!/\!_{\sqrt{M}})$. Consequently, the overall convergence rate of PWM should be limited by this slowest component.

A possible explanation is that a certain bottleneck term in Eq.(5), which has a better convergence rate but a large constant, slows down the overall convergence. Fig.4 decomposes the convergence rate of each error term using the closed-form expressions in Eq. (6). The analysis reveals that the CDF term dominates the error—it follows a faster $\mathcal{O}(1/M)$ rate but with a large constant—while the other two terms approximately follow the slower $\mathcal{O}({}^1\!/\!_{\sqrt{M}})$ rate but with a smaller constant.

This leads to the second pitfall: *the hidden bottleneck of the PWM estimator*. Unlike the first pitfall, this issue is more subtle and requires a step-by-step examination. First, consider the CDF term in Eq. (5). It is a nonlinear functional of the estimated CDF $\hat{F}$, making it a type of plug-in estimator. As previously discussed, we typically approximate the CDF using the empirical CDF, which has a well-known asymptotic convergence rate for i.i.d. samples:

$$\hat{F}(y) - \epsilon \le F(y) \le \hat{F}(y) + \epsilon, \quad \text{where } \epsilon = \sqrt{\frac{\ln {}^2\!/\!_\alpha}{2M}}, \tag{7}$$

with at least probability $1 - \alpha$. This bound is well known as the Dvoretzky–Kiefer–Wolfowitz (DKW) inequality (Dvoretzky et al., 1956), which states that the convergence rate of the empirical CDF is also $\mathcal{O}({}^1\!/\!_{\sqrt{M}})$. However, this contradicts our observation in Fig. 4.

This discrepancy can be explained by plug-in bias, a common issue where the convergence rate of a plug-in estimator for a nonlinear functional introduces an asymptotic bias term that is independent of the finite-sample estimation error. Even though the empirical CDF weakly converges to the true CDF, the nonlinearity in the CRPS functional causes the expectation of the plug-in estimator to deviate from the true CRPS by a constant bias term. Therefore, to fully leverage the PWM estimator, we need to correct for this bias.

3.3 The source of plug-in bias

Recall that the empirical CDF is defined as: $\hat{F}(y_i) = \frac{1}{M}\sum_{j=1}^{M} \mathbb{1}_{y_j \le y_i}$. For simplicity, we denote the CDF term in Eq.(5) as $\mathbb{E}[C(\hat{F})]$ and the corresponding analytical solution as $C(F)$ from Eq.(6). We have:

$$\begin{aligned}
\mathbb{E}[C(\hat{F})] &= \frac{1}{M}\sum_{i=1}^{M} y_i \hat{F}(y_i), \\
&= \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} y_i \mathbb{1}_{y_j \le y_i}, \\
&= \frac{1}{M^2}\sum_{i,j=[M]} h(y_i, y_j), \\
&= \frac{1}{M^2}\left(\underbrace{M\mathbb{E}[h(y_i, y_i)]}_{\text{diagonal}} + \underbrace{M(M-1)\mathbb{E}[h(y_i, y_i)]}_{\text{off-diagonal}} \right),
\end{aligned}$$

where $h(y_i, y_j) = y_i \mathbb{1}_{y_j \le y_i}$. We then have:

$$\begin{aligned}
\mathbb{E}[h(y_i, y_i)] &= \mathbb{E}[y \mathbb{1}_{y \le y}] = \mathbb{E}[y] = \mu_l, \\
\mathbb{E}[h(y_i, y_i)] &= \mathbb{E}[y_i \mathbb{1}_{y_i \ge y_j}] = C(F).
\end{aligned}$$

Thus, we observe that the off-diagonal term corresponds to the true value $C(F)$, yet the estimator $C(\hat{F})$ unnecessarily includes an additional diagonal term. Using these identities, we obtain:

$$\text{plug-in bias} := \mathbb{E}[C(\hat{F})] - C(F) = \frac{1}{M}\left(\mu_l - C(F)\right).$$

This explains our observations. In Fig. 2(b), we see a μ_l-dependent bias term, while in Fig. 4, the CDF term exhibits an $\mathcal{O}(1/M)$ convergence rate. These artifacts arise solely due to the inclusion of the diagonal term.

Table 1: (Copied from Tóth et al. (2024) Table 1): Forecasting results on eight benchmark datasets ranked by CRPS. The best and second best models have been shown as **bold** and underlined, respectively.

method	Solar	Electricity	Traffic	Exchange	M4	UberTLC	KDDCup	Wikipedia
Seasonal Naïve	0.512 ± 0.000	0.069 ± 0.000	0.221 ± 0.000	0.011 ± 0.000	0.048 ± 0.000	0.299 ± 0.000	0.561 ± 0.000	0.410 ± 0.000
ARIMA	0.545 ± 0.006	-	-	**0.008 ± 0.000**	0.044 ± 0.001	0.284 ± 0.001	0.547 ± 0.004	-
ETS	0.611 ± 0.040	0.072 ± 0.004	0.433 ± 0.050	**0.008 ± 0.000**	0.042 ± 0.001	0.422 ± 0.001	0.753 ± 0.008	0.715 ± 0.002
Linear	0.569 ± 0.021	0.088 ± 0.008	0.179 ± 0.003	0.011 ± 0.001	0.039 ± 0.001	0.360 ± 0.023	0.513 ± 0.011	1.624 ± 1.114
DeepAR	0.389 ± 0.001	0.054 ± 0.000	0.099 ± 0.001	0.011 ± 0.003	0.052 ± 0.006	**0.161 ± 0.002**	0.414 ± 0.027	0.231 ± 0.008
MQ-CNN	0.790 ± 0.063	0.067 ± 0.001	-	0.019 ± 0.006	0.046 ± 0.003	0.436 ± 0.020	0.516 ± 0.012	0.220 ± 0.001
DeepState	0.379 ± 0.002	0.075 ± 0.004	0.146 ± 0.018	0.011 ± 0.001	0.041 ± 0.002	0.288 ± 0.087	-	0.318 ± 0.019
Transformer	0.419 ± 0.008	0.076 ± 0.018	0.102 ± 0.002	0.010 ± 0.000	0.040 ± 0.014	0.192 ± 0.004	0.411 ± 0.021	**0.214 ± 0.001**
TSDiff	0.358 ± 0.020	**0.050 ± 0.002**	**0.094 ± 0.003**	0.013 ± 0.002	0.039 ± 0.006	0.172 ± 0.008	0.754 ± 0.007	0.218 ± 0.010
SVGP	**0.341 ± 0.001**	0.104 ± 0.037	-	0.011 ± 0.001	0.048 ± 0.001	0.326 ± 0.043	0.323 ± 0.007	-
DKLGP	0.780 ± 0.269	0.207 ± 0.128	-	0.014 ± 0.004	0.047 ± 0.004	0.279 ± 0.068	0.318 ± 0.010	-
RS^{3}GP	0.377 ± 0.004	0.057 ± 0.001	0.165 ± 0.001	0.012 ± 0.001	0.038 ± 0.003	0.354 ± 0.016	0.297 ± 0.007	0.310 ± 0.012
VRS^{3}GP	0.366 ± 0.003	0.056 ± 0.001	0.160 ± 0.002	0.011 ± 0.001	**0.035 ± 0.001**	0.347 ± 0.009	**0.291 ± 0.015**	0.295 ± 0.005

3.4 Why does this error matter?

These errors are significant because they are on the same order of magnitude as the differences between forecasting models. As an example, we reference the experimental results from Table 1 in Tóth et al. (2024), which uses CRPS as implemented in the GluonTS library (quantile-based estimator with $M = 100$ and $Q = 9$). The reported differences between models are roughly in the range of 10^{-1} to 10^{-3}, while the CRPS approximation errors at the default setting are on the order of 10^{-1} to 10^{-2}. In other words, crude CRPS approximations can lead to incorrect rankings of forecasting model performance [6].

Summary. The pitfalls are summarised as follows:

> **Pitfall 1: Spurious supremacy of quantile-based estimator.** Under the default settings, the quantile-based approach appears superior. However, its convergence behavior is a complex function of both the sample size M and the number of quantile grids Q. It is also computationally more expensive than the PWM estimator.
>
> **Pitfall 2: Plug-in bias of PWM estimator.** A naïve MC integration introduces plug-in bias in the CDF term estimation due to the nonlinear nature of the functional.

Due to these errors, the current evaluation methods for time-series forecasting models may not accurately reflect their true performance rankings. Therefore, it is necessary to correct the plug-in bias in the PWM estimator to achieve faster and more reliable convergence.

4 Method: kernel quadrature

Now, we introduce our approach, an unbiased estimator for the PWM-based CRPS. Any method that can unbias the PWM estimator could be used to address this issue—for example, multi-level Monte Carlo (Hong and Juneja, 2009; Rainforth et al., 2018), but we opted for kernel quadrature based approximation.

4.1 Unbiased PWM estimator

We introduce the following unbiased estimator:

$$\tilde{h}(y_i, y_j) := \frac{1}{2}\left(y_i \mathbb{1}_{y_i > y_j} + y_j \mathbb{1}_{y_j > y_i}\right). \tag{8}$$

Note that we use $y_j > y_i$ rather than $y_j \geq y_i$, which naturally ensures zero diagonal elements. As a result, we obtain the unbiased estimator $\mathbb{E}[\tilde{h}(y_i, y_j)] = C(F)$.

Recall that y_l is the observed value. The simplest way to utilize this unbiased estimator is through Monte Carlo (MC) integration:

$$\begin{aligned}\widehat{\text{CRPS}} =& \mathbb{E}_{y \sim \mathbb{P}(f \mid x_i)}[|y - y_l|] + \mathbb{E}_{y \sim \mathbb{P}(f \mid x_i)}[y] \\ &- 2\mathbb{E}_{y, y' \sim \mathbb{P}(f \mid x_i)}[\tilde{h}(y, y')]\end{aligned} \tag{9}$$

4.2 Scalable estimator via quantization

The estimation bias has already been eliminated by the simple solution described above; the remaining challenge is scalability. Although an $\mathcal{O}(M \log M)$ algorithm is asymptotically efficient, it can become impractical for very large M due to memory constraints and increased runtime. To address this, we adopt a kernel quadrature approach that replaces the original set of M equally weighted points with a much smaller collection of m weighted samples such that:

$$\frac{1}{M}\sum_{i=1}^{M} z(y_i) \approx \sum_{i=1}^{m} w_i z(y_i), \tag{10}$$

where $m \ll M$, while ensuring that the approximation error remains minimal. Here, we set $z(y)$ as the symmetrized integrand function of Eq. (9)[7], then the above

[6] To be clear, this issue is not specific to Tóth et al. (2024), but is rather a persistent problem within the time-series forecasting community. Since the default setting in GluonTS has become the de facto standard for benchmarking time-series forecasting models, it has been widely adopted in various studies.

[7] We set $z(y)$ as the following positive and symmetric function (ξ is a constant that ensures the positivity.):

$$\begin{aligned} & k(y_i, y_j \mid y_l) := \tilde{k}(y_i, y_j \mid y_l) + \xi \delta_{y_i, y_j}, \\ \text{where} \quad & \tilde{k}(y_i, y_j \mid y_l) := G(y_i, y_j \mid y_l) - 2\tilde{h}(y_i, y_j), \\ & G(y_i, y_j \mid y_l) := \tilde{g}(y_i \mid y_l) + \tilde{g}(y_j \mid y_l), \\ & \tilde{g}(y_i \mid y_l) := \frac{1}{2}\left(|y_i - y_l| + y_i\right), \end{aligned}$$

Eq. (10) can be understood as compressing the MC integration points into smaller, weighted points (also known as *quantization* (Graf and Luschgy, 2007)). Such a smaller weighted set can exist, given by Tchakaloff's theorem (Tchakaloff, 1957):

Theorem 4.1 (Tchakaloff's theorem). *Let $x_1, \cdots, x_m$ be m samples, $w_1, \cdots, w_m \geq 0$ be (positive) weights such that $\sum_{i=1}^m w_i = 1$, $\{x_j\}_{j=1}^M = \mu(x)$ be a discrete measure with $M > m$, $\boldsymbol{\varphi} := (\varphi_1, \cdots, \varphi_n)^\top$ be a n-dimensional, integrable, and vector-valued function with $n \leq M+1$, then there exists a cubature rule*

$$\int_{\mathcal{X}} \boldsymbol{\varphi}(x) d\mu(x) = \sum_{i=1}^{m} w_i \boldsymbol{\varphi}(x_i). \tag{11}$$

such that Eq. (4.1) holds.

Notably, this is equality, implying that "compression" into smaller weighted points can be performed without introducing approximation errors (lossless compression). The only distinction in our case is that the function $z(y)$ is not vector-valued.

To address this, Hayakawa et al. (2022) introduced the Nyström method, which approximates the symmetric function as a vector-valued function via eigendecomposition of the Gram matrix, followed by a cubature construction algorithm using recombination. It is based on Carathéodory's theorem and formulates the problem as one of subset selection: finding the convex hull of random points $(x_j)_{j=1}^m \subset (x_i)_{i=1}^M$. At a high level, the algorithm is conceptually closer to k-means clustering than to classical optimization. The method iteratively:

1. Finds a linear dependency among the current support points,
2. Removes a point while preserving the weighted sum (via pivoting),
3. Updates the weights accordingly,
4. Repeats until only nonzero weights remain.

The computational complexity of this algorithm is $\mathcal{O}(C_\varphi M + m^3 \log(M/m))$, where C_φ is the cost of evaluating the functions $(\varphi)_i^n$ at a given point. Thus, it scales linearly with the sample size M and remains computationally efficient. For further methodological details, see Hayakawa et al. (2022); Adachi et al. (2022).

Since this approach relies on cubature, the only source of error comes from the Nyström approximation. The Nyström method exploits the spectral decay of the Gram matrix, meaning that if the kernel is smooth, the decay is rapid and the error bound remains tight. Empirically, we found that our kernel decays sufficiently fast for large M. The Nyström method was originally introduced for kernel quadrature but is more general beyond well-defined Mercer kernels used in typical kernel quadrature, thus we can apply this to the symmetrised matrix by $z(y)$. Yet, this scalable approach remains optional, as using all samples does not introduce additional error. Therefore, it is only necessary when M is too large to handle computationally.

5 Related work

Probabilistic time-series forecasting There is a vast array of probabilistic time-series forecasting models. In classical statistical approaches, commonly used models include seasonal naïve, ARIMA, ETS, and linear (ridge) regression (Hyndman, 2018). For deep learning models, key representatives for each architecture include: DeepAR (Salinas et al., 2020), based on RNNs, MQ-CNN (Wen et al., 2017), based on CNNs, DeepState (Rangapuram et al., 2018), based on state-space models, Transformer-based models (Vaswani et al., 2017), leveraging self-attention, and TSDiff (Kollovieh et al., 2024), a diffusion-based model, which is considered the current state-of-the-art. For GP models, commonly tested methods include: Sparse Variational GP (SVGP) (Hensman et al., 2013), Deep Kernel Learning GPs (Wilson et al., 2016), and VRS^{3}GP (Tóth et al., 2024), a recent model using signature kernels, which has demonstrated state-of-the-art performance within the GP framework and achieves comparable accuracy to TSDiff, while requiring significantly shorter training times.

Metric for time-series Candille and Talagrand (2005); Ferro et al. (2008); Gneiting and Raftery (2007) demonstrated that the CRPS estimator is inherently sensitive to both bias and variance, as CRPS generalizes the absolute error. To overcome, various methods have been proposed to mitigate this bias. Müller et al. (2005) addressed bias in a CRPS-based skill measure within the specific context of ensemble prediction. Ferro (2014) introduced a bias correction factor to improve the fairness of CRPS for ensemble forecasts, accounting for finite ensemble sizes. Zamo and Naveau (2018) further reviewed CRPS estimators derived from limited sample information, providing practical guidelines for selecting the optimal estimator based on the type of random variable. Unlike these post-processing corrections aimed at reducing bias in CRPS estimators, our study constructed an unbiased CRPS estimator.

Kernel quadrature There are several kernel quadrature algorithms, including herding/optimization (Chen et al., 2010; Huszár and Duvenaud, 2012), random sampling (Bach, 2017), determinantal point processes (DPP; Belhadji et al. (2019), recombination (Hayakawa et al., 2022). While any of these methods can be applied to our problem, their primary focus is on selecting quadrature nodes, rather than debiasing.

6 Results

6.1 Experimental setup

We implemented our code using PyTorch (Paszke et al., 2019) and GPyTorch (Gardner et al., 2018) for modeling Gaussian processes (GPs). The implementation of scalable kernel quadrature is based on SOBER (Adachi et al., 2024a), but our method is not limited to this library. All experiments were averaged over 100 repeats

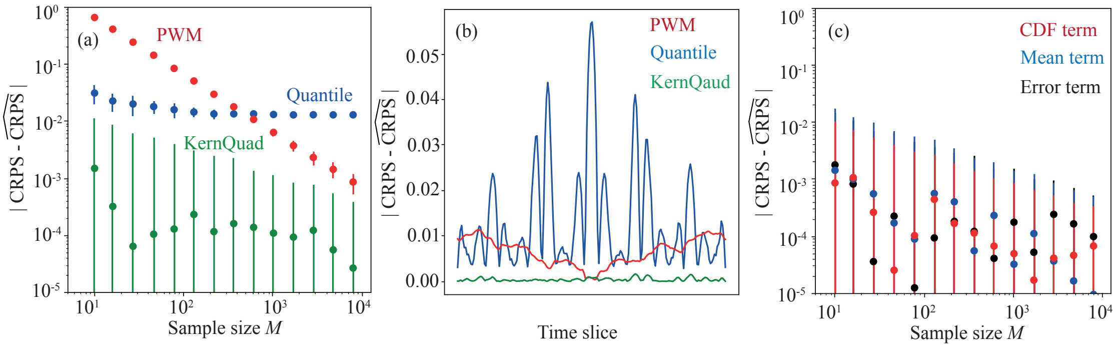

Figure 5: The kernel quadrature (KernQuad) method achieves unbiased estimation. (a) KernQuad outperforms the two widely used alternatives across all sample sizes M. (b) KernQuad eliminates bias across time slices. (c) The previously biased CDF term is now at the same level as the other two terms.

Table 2: Comparison of four CRPS estimators on the high-frequency multi-sinusoidal test dataset across three forecasting models. The quantile estimator incorrectly ranks SRV^3 as performing worse than RFF-GP on this dataset. The actual values should be scaled by $\times 10^{-3}$.

method	SVGP	RFFGP	VRS^{3}GP
closed-form	8.1401 ± 0.9767	2.424 ± 0.0081	2.419 ± 0.0094
quantile	9.0332 ± 1.1105	2.701 ± 0.0240	2.702 ± 0.0254
PWM	8.2144 ± 1.0066	2.409 ± 0.0326	2.397 ± 0.0468
KernQuad	8.1110 ± 0.9264	2.424 ± 0.0147	2.422 ± 0.0090

for the Ackley function and 3 repeats for the multi-sinusoidal wave datasets, each with different random seeds. The experiments were conducted on a MacBook Pro (2019), 2.4 GHz 8-Core Intel Core i9, 64 GB.

6.2 Unbiased estimator

We confirmed that the bias issues identified in Section 3 have been resolved using our kernel quadrature (KernQuad) approach. Fig. 5 clearly demonstrates that KernQuad achieves unbiased estimation. The previously observed bias over time slices has disappeared, leaving only location-independent noise. Additionally, the CDF term, which was previously the primary bottleneck, is no longer a limiting factor, as it now exhibits the same convergence rate as the other error terms. These results clearly indicate that KernQuad is unbiased and consistently outperforms the two existing baselines.

6.3 Time-series forecasting models

We further evaluate the performance of KernQuad across various time-series forecasting models. Among the available models, we select SVGP (Hensman et al., 2013), random Fourier feature GP (RFF-GP), and variational recurrent sparse spectrum signature GP (VRS^{3}GP; Tóth et al. (2024)) for comparison. The primary reason for choosing these models is that GP-based methods allow for the computation of true CRPS, enabling us to directly assess estimation errors. Notably, VRS^{3}GP has demonstrated state-of-the-art performance in time-series forecasting tasks, as shown in Table 1, making this a practically relevant setting.

To analyze performance dependencies, we test on synthetic time-series data generated from multi-sinusoidal waves with four weighted components of different frequencies but no phase shift. We prepare two synthetic test functions: (a) Low-frequency waves $[0.1, 1, 2, 5]$, and (b) High-frequency waves $[1, 5, 10, 20]$ for the $L = 800$ training and $T = 100$ test time steps. By definition, learning high-frequency components is easier than low-frequency ones, as the latter appear less frequently. Thus, the low-frequency dataset is more challenging, making it easier to distinguish model performance differences. Conversely, the high-frequency dataset is easier for most models, making it harder to differentiate model performance, and thus CRPS estimation accuracy becomes more critical. To ensure robustness, we repeat model training three times with different random seeds. Consequently, even the closed-form CRPS estimator is computed as a MC integration over three samples.

Fig. 6 presents the convergence rates across three forecasting models and two datasets. The trends remain consistent across all datasets and models: Our kernel quadrature method consistently outperforms the two other CRPS estimators. The quantile-based estimator, which is the current default approach, performs the worst. The estimation error from the quantile estimator leads to incorrect model rankings. Table 2 further highlights this issue, showing that the quantile estimator erroneously ranks RFF-GP as outperforming VRS^{3}GP. While the difference between the two models falls within the standard deviation, making this specific instance detectable by closely examining error bars, it serves as a clear counterexample where the quantile-based CRPS estimator can misrank models. This further motivates the use of our kernel quadrature estimator instead.

Although this issue is identifiable in GP-based forecasting models, where we can numerically verify CRPS approximation errors, the same verification is not possible for deep learning models. As seen in Table 1, the standard deviations of deep learning models are typically larger than those of GP-based methods, and their variability is unpredictable, as different methods exhibit the largest standard deviation on different datasets. Thus, we argue that fixing this issue with our kernel quadrature approach is essential for ensuring more reliable model performance comparisons.

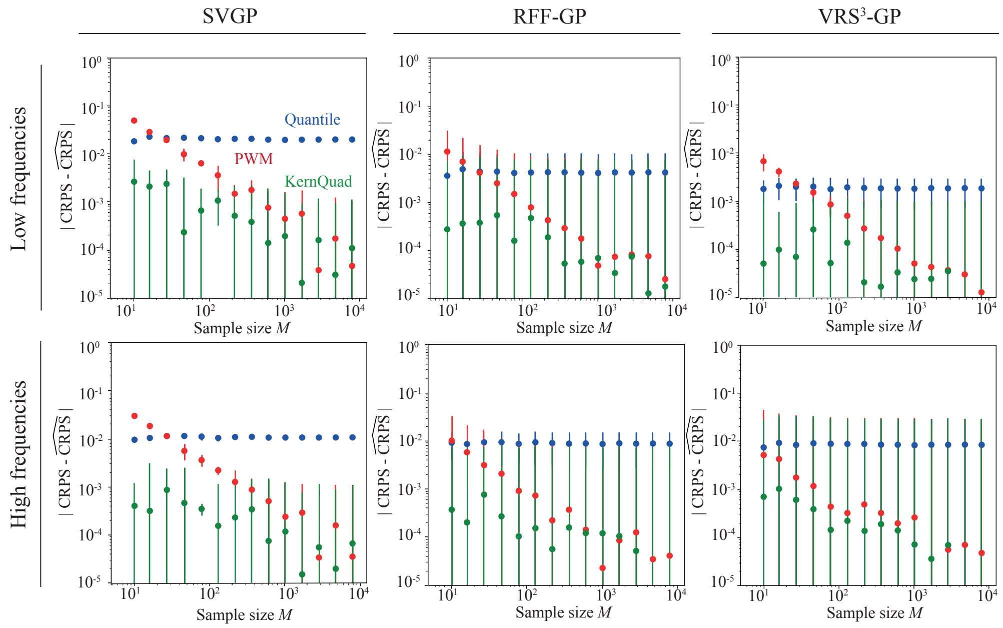

Figure 6: The kernel quadrature method consistently outperforms the quantile-based and PWM estimators across three forecasting models and two test datasets.

7 Conclusion and Limitations

We first identify the pitfalls of the two existing CRPS estimators; the quantile-based and PWM estimators. The quantile estimator, which is the current default in GluonTS (Alexandrov et al., 2020), exhibits a consistent bias that cannot be eliminated by increasing the sample size. The PWM estimator suffers from plug-in bias, which weakly converges to the true CRPS in the infinite sample limit. However, this bias remains the bottleneck in its estimation process. To address the pitfalls, we propose an unbiased estimator and its scalable approximation using kernel quadrature. Our proposed unbiased estimator consistently achieves lower estimation errors across all sample sizes, three datasets, and three forecasting models. Moreover, we demonstrated that the quantile estimator can lead to incorrect model rankings on certain datasets, whereas our kernel quadrature estimator preserves the correct rankings. This highlights the importance of minimizing approximation errors in time-series forecasting model evaluation.

Similar CDF-based estimators exist for other probabilistic metrics, such as the energy score (Fahy, 1994), calibration score (Futami and Fujisawa, 2024; Fujisawa and Futami, 2025), conformal prediction (Snell and Griffiths, 2025), and spectral risk measure (Pandey et al., 2021). We anticipate that our approach can be extended to these metrics, enabling more reliable and unbiased estimation in broader applications. Recent techniques from probabilistic numerics, such as those proposed by Wenger et al. (2020); Adachi et al. (2024b), also present promising directions for further extension.

Although kernel quadrature offers scalable computation, it still introduces additional approximation error. Also, the inherent convergence rate is limited by the standard MC rate $\mathcal{O}(1/\sqrt{M})$. Leveraging the faster convergence rate of standard Bayesian quadrature (BQ) is a promising direction, yet it produces the following challenge:

1. Applying BQ to the error and mean terms is straightforward, but the CDF term is challenging. The CDF is monotonic and bounded, whereas GPs do not inherently impose such constraints. Although prior work addresses these limitations, the methods are often computationally expensive. In time-series applications, where datasets can contain millions of time points, CRPS estimation must be repeated millions of times per random seed. Hence, computational efficiency is critical.
2. Constructing suitable (x, y) pairs is non-trivial. Here, x represents samples like $f(x_l)$, but obtaining the corresponding "ground-truth CDF" values for y is difficult. We tried using empirical CDF values as y for GP training and applied BQ, but the results were worse than MC integration. If one uses a standard kernel like RBF, the task ends up being more similar to kernel density estimation than BQ. Recent paper, Snell and Griffiths (2025), which takes a promising alternative approach by placing a Dirichlet prior on quantile spacing. While it still requires MC integration, it may be worth exploring in future work.

Acknowledgements

We thank anonymous reviewers for their valuable feedback and discussions. MA is supported by the Clarendon Fund, the Oxford Kobe Scholarship, the Watanabe Foundation, and Toyota Motor Corporation. MF is supported by KAKENHI (Grant Number: 25K21286) and Toyota Motor Corporation.

References

D. Ackley. *A connectionist machine for genetic hillclimbing*, volume 28. Springer science & business media, 1987. URL https://doi.org/10.1007/978-1-4613-1997-9.

M. Adachi, S. Hayakawa, M. Jørgensen, H. Oberhauser, and M. A. Osborne. Fast bayesian inference with batch bayesian quadrature via kernel recombination. *Advances in Neural Information Processing Systems*, 35:16533–16547, 2022.

M. Adachi, Y. Kuhn, B. Horstmann, A. Latz, M. A. Osborne, and D. A. Howey. Bayesian model selection of lithium-ion battery models via Bayesian quadrature. *IFAC-PapersOnLine*, 56(2):10521–10526, 2023.

M. Adachi, S. Hayakawa, M. Jørgensen, S. Hamid, H. Oberhauser, and M. A. Osborne. A quadrature approach for general-purpose batch bayesian optimization via probabilistic lifting. *arXiv preprint arXiv:2404.12219*, 2024a.

M. Adachi, S. Hayakawa, M. Jørgensen, X. Wan, V. Nguyen, H. Oberhauser, and M. A. Osborne. Adaptive batch sizes for active learning: A probabilistic numerics approach. In *International Conference on Artificial Intelligence and Statistics*, pages 496–504. PMLR, 2024b.

A. Alexandrov, K. Benidis, M. Bohlke-Schneider, V. Flunkert, J. Gasthaus, T. Januschowski, D. C. Maddix, S. Rangapuram, D. Salinas, J. Schulz, et al. GluonTS: Probabilistic and neural time series modeling in Python. *Journal of Machine Learning Research (JMLR)*, 21(116):1–6, 2020.

F. Bach. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18:714, 2017.

A. Belhadji, R. Bardenet, and P. Chainais. Kernel quadrature with DPPs. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

G. E. P. Box and G. M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day series in time series analysis and digital processing. Holden-Day, 1970. ISBN 9780816210947.

G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

C. Bui, N. Pham, A. Vo, A. Tran, A. Nguyen, and T. Le. Time series forecasting for healthcare diagnosis and prognostics with the focus on cardiovascular diseases. In *International Conference on the Development of Biomedical Engineering in Vietnam (BME)*, pages 809–818. Springer, 2018.

Z. Cai. Regression quantiles for time series. *Econometric theory*, 18(1):169–192, 2002.

G. Candille and O. Talagrand. Evaluation of probabilistic prediction systems for a scalar variable. *Quarterly Journal of the Royal Meteorological Society*, 131(609): 2131–2150, 2005.

Y. Chen, M. Welling, and A. Smola. Super-samples from kernel herding. In *International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.

F. M. Dekking, C. Kraaikamp, H. P. Lopuhaä, and L. E. Meester. *A Modern Introduction to Probability and Statistics: Understanding why and how*. Springer Science & Business Media, 2006.

J. Dumas, A. Wehenkel, D. Lanaspeze, B. Cornélusse, and A. Sutera. A deep generative model for probabilistic energy forecasting in power systems: normalizing flows. *Applied Energy*, 305:117871, 2022.

A. Dvoretzky, J. Kiefer, and J. Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, pages 642–669, 1956.

F. J. Fahy. Statistical energy analysis: a critical overview. *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences*, 346(1681):431–447, 1994.

C. A. T. Ferro. Fair scores for ensemble forecasts. *Quarterly Journal of the Royal Meteorological Society*, 140 (683):1917–1923, 2014.

C. A. T. Ferro, D. S. Richardson, and A. P. Weigel. On the effect of ensemble size on the discrete and continuous ranked probability scores. *Meteorological Applications*, 15(1):19–24, 2008.

M. Fujisawa and F. Futami. PAC-Bayes analysis for recalibration in classification. In *Forty-second International Conference on Machine Learning*, 2025.

F. Futami and M. Fujisawa. Information-theoretic generalization analysis for expected calibration error. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 84246–84297. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9961e42624a6c083279303767c73269d-Paper-Conference.pdf.

J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, pages 7576–7586, 2018.

T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

T. Gneiting, A. E. Raftery, A. H. Westveld, and T. Goldman. Calibrated probabilistic forecasting using ensemble model output statistics and minimum crps estimation. *Monthly Weather Review*, 133(5): 1098–1118, 2005.

S. Graf and H. Luschgy. *Foundations of quantization for probability distributions.* Springer, 2007.

S. Hayakawa, H. Oberhauser, and T. Lyons. Positively weighted kernel quadrature via subsampling. *Advances in Neural Information Processing Systems*, 35: 6886–6900, 2022.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence (UAI)*, 2013.

H. Hersbach. Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting*, 15(5):559–570, 2000.

L. J. Hong and S. Juneja. Estimating the mean of a nonlinear function of conditional expectation. In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, pages 1223–1236. IEEE, 2009.

F. Huszár and D. Duvenaud. Optimally-weighted herding is bayesian quadrature. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 377–386, 2012.

R. Hyndman. *Forecasting: principles and practice.* OTexts, 2018.

K. Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1):307–319, 2003.

M. Kollovieh, A. F. Ansari, M. Bohlke-Schneider, J. Zschiegner, H. Wang, and Y. B. Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. *Advances in Neural Information Processing Systems*, 36, 2024.

J. E. Matheson and R. L. Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.

M. A. Morid, O. R. L. Sheng, and J. Dunbar. Time series prediction using deep learning methods in healthcare. *ACM Transactions on Management Information Systems*, 14(1):1–29, 2023.

W. A. Müller, C. Appenzeller, F. J. Doblas-Reyes, and M. A. Liniger. A debiased ranked probability skill score to evaluate probabilistic ensemble forecasts with small ensemble sizes. *Journal of Climate*, 18(10): 1513–1523, 2005.

J. Oskarsson, T. Landelius, M. Deisenroth, and F. Lindsten. Probabilistic weather forecasting with hierarchical graph neural networks. In *Advances in Neural Information Processing Systems*, volume 37, pages 41577–41648, 2024.

A. K. Pandey, L. Prashanth, and S. P. Bhat. Estimation of spectral risk measures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12166–12173, 2021.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga. PyTorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

T. Rainforth, R. Cornish, H. Yang, A. Warrington, and F. Wood. On nesting monte carlo estimators. In *International Conference on Machine Learning*, pages 4267–4276. PMLR, 2018.

S. S. Rangapuram, M. W. Seeger, J. Gasthaus, L. Stella, Y. Wang, and T. Januschowski. Deep state space models for time series forecasting. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110550, 2013.

D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3):1181–1191, 2020.

O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90:106181, 2020.

J. C. Snell and T. L. Griffiths. Conformal prediction as bayesian quadrature. *arXiv preprint arXiv:2502.13228*, 2025.

M. Taillardat, O. Mestre, M. Zamo, and P. Naveau. Calibrated ensemble forecasts using quantile regression forests and ensemble model output statistics. *Monthly Weather Review*, 144(6):2375–2393, 2016.

V. Tchakaloff. Formules de cubatures mécaniques à coefficients non négatifs. *Bull. Sci. Math*, 81(2):123–134, 1957.

C. Tóth, M. Adachi, M. A. Osborne, and H. Oberhauser. Learning to forget: Bayesian time series forecasting using recurrent sparse spectrum signature gaussian processes. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024. URL https://doi.org/10.48550/arXiv.2412.19727.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

H. Wang, Z. Lei, X. Zhang, B. Zhou, and J. Peng. A review of deep learning for renewable energy forecasting. *Energy Conversion and Management*, 198: 111799, 2019.

R. Wen, K. Torkkola, B. Narayanaswamy, and D. Madeka. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017.

J. Wenger, H. Kjellström, and R. Triebel. Nonparametric calibration for classification. In *International Conference on Artificial Intelligence and Statistics*, pages 178–190. PMLR, 2020.

A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Artificial intelligence and statistics (AISTATS)*, pages 370–378. PMLR, 2016.

M. Zamo and P. Naveau. Estimation of the continuous ranked probability score with limited information and applications to ensemble weather forecasts. *Mathematical Geosciences*, 50:209–234, 2018.