MVG-CRPS: A Robust Loss Function for Multivariate Probabilistic Forecasting

Anonymous Author(s)
Affiliation
Address
email

Abstract

Multivariate probabilistic forecasting typically leverages neural network-based distributional regression, often employing Gaussian assumptions to simplify computation. While the standard negative log-likelihood provides analytical convenience, its sensitivity to outliers can severely degrade forecasting accuracy. Conversely, robust alternatives like the Energy Score, although less sensitive to extreme values, rely heavily on computationally expensive sampling approximations, limiting scalability in neural network training. To bridge this gap, we introduce the MVG-CRPS, a novel, strictly proper scoring rule for multivariate Gaussian distributions that maintains robustness to outliers while providing a closed-form expression, enabling efficient training and evaluation. Our approach leverages a whitening transformation, decorrelating multivariate outputs and reducing the multivariate scoring task to tractable univariate CRPS computations. Experiments on real-world datasets for both multivariate autoregressive and univariate sequence-to-sequence (Seq2Seq) forecasting tasks demonstrate that MVG-CRPS enhances robustness and predictive performance.

1 Introduction

Probabilistic forecasting is critical in applications ranging from financial risk management [1], to weather forecasting [2] and healthcare analytics [3], where accurate quantification of predictive uncertainty directly informs decision-making. Multivariate probabilistic forecasting models extend beyond point estimates, producing joint probability distributions across multiple correlated continuous variables. Neural network-based methods have become a dominant paradigm due to their flexibility and expressiveness [4–6]. Typically, these methods rely on parametric assumptions such as multivariate Gaussian distributions, allowing closed-form loss computations (e.g., log-likelihood) and efficient backpropagation.

Despite widespread adoption, standard metrics for model inference such as the negative log-likelihood (log-score) present substantial challenges. Most notably, under the Gaussian family, the log-score heavily penalizes unlikely events and outliers due to its exponential sensitivity in the tails of distributions, making it excessively sensitive to anomalies and model misspecification [7, 8]. As a result, neural network models trained using the log-score can generate overly conservative or inaccurate predictive distributions when exposed to real-world data characterized by occasional extreme events.

To address the limitations of the log-score, the Energy Score [ES, 9] emerged as a popular robust alternative. It generalizes the continuous ranked probability score [CRPS, 10, 11] for univariate distributions and effectively mitigates sensitivity to outliers by evaluating forecasts through expected pairwise distances between predictions and observations. However, the ES lacks a closed-form analytical expression in most cases, necessitating computationally intensive Monte Carlo sampling to

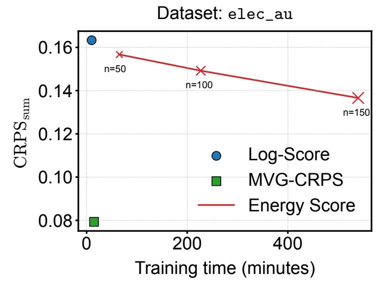

Figure 1: An example showing MVG-CRPS achieves better accuracy and faster training by avoiding sampling and reducing sensitivity to outliers. ES results are shown for different sample sizes.

approximate its value and gradients. Such approximations significantly slow down neural network training, limiting practical scalability [12, 13].

Motivated by the need for a robust yet computationally efficient scoring rule, this paper introduces MVG-CRPS (Multivariate Gaussian CRPS). We propose a strictly proper scoring rule specifically designed for multivariate Gaussian probabilistic forecasting tasks. Our approach circumvents the computational limitations of the ES by leveraging a PCA whitening transformation, decomposing the multivariate Gaussian distribution into independent, standard normal variables. Consequently, the multivariate scoring problem reduces to a set of analytically tractable univariate CRPS computations. MVG-CRPS provides explicit analytical gradients, enabling efficient integration into neural network training. The advantages of our approach are illustrated in Fig. 1, where the model trained with MVG-CRPS achieves higher accuracy while significantly reducing training time. The key contributions of our work are:

- We propose MVG-CRPS, a novel scoring rule for multivariate probabilistic forecasting that is less sensitive to outliers and extreme tails of the data distribution. Under the multivariate Gaussian family, we prove that MVG-CRPS is strictly proper.
- The proposed MVG-CRPS has a closed-form expression, allowing for the analytical computation of derivatives. This property facilitates efficient integration with backpropagation-based training in deep learning models and significantly reduces the computational cost compared to sampling-based alternatives.
- We perform extensive experiments with deep probabilistic forecasting models on real-world datasets. Our results demonstrate that MVG-CRPS balances accuracy and efficiency more effectively than standard scoring rules.

2 Related Work

2.1 Probabilistic Forecasting

Probabilistic forecasting focuses on modeling the complete probability distribution of target variables rather than producing single-point estimates. This comprehensive approach is essential for quantifying uncertainty inherent in time series data, thereby enabling more informed risk assessment and decision-making. Probabilistic forecasting methods typically fall into two main categories: parametric methods, which assume explicit probability density functions (PDFs), and non-parametric methods, which rely on quantile estimation [5].

Non-parametric methods generally forecast specific quantiles of the target distribution, thus avoiding restrictive parametric assumptions. A prominent example is the MQ-RNN [14], which leverages a Seq2Seq recurrent neural network (RNN) architecture to forecast multiple quantiles simultaneously. These quantile forecasts offer a robust approximation of the underlying distribution, making them particularly effective for capturing asymmetric and heavy-tailed behaviors.

Parametric methods assume a predefined probability distribution—such as Gaussian or Poisson—and estimate its parameters using neural networks. The DeepAR model [15], for instance, employs an RNN to capture hidden state transitions and predict Gaussian distribution parameters at each time step. GPVar [4], its multivariate extension, incorporates a Gaussian copula to transform observations into Gaussian variables, thus modeling joint dependencies among multiple time series effectively. This method efficiently captures temporal and cross-series correlations through generalized least squares (GLS) approaches [16, 17] or dynamic regression [18].

Neural networks also facilitate modeling more complex probabilistic structures, including state-space models (SSMs) [19, 20], normalizing flows (NFs) [6], and diffusion models [21]. Additionally, copula-based methods explicitly model dependencies between multiple time series. Recent studies by Drouin et al. [22] and Ashok et al. [23] employ copulas to combine individual marginal distributions and dependency structures, achieving flexible multivariate modeling capabilities. Most existing approaches predominantly use the log-score as their optimization criterion.

2.2 Scoring Rules

Scoring rules quantitatively assess probabilistic forecast quality by comparing predicted distributions with observed outcomes. A scoring rule is deemed proper if it incentivizes honest forecasting, achieving its minimal expected score when the predicted distribution when the predicted probability distribution p matches the true distribution q. Formally, a scoring rule $s(p, q)$ is proper if the divergence $d(p, q) = s(p, q) - s(q, q)$ is non-negative and it is strictly proper if $d(p, q) = 0$ implies $p = q$ [24].

The negative log-likelihood (log-score) is a prevalent strictly proper scoring rule, evaluating predictive densities directly at observed outcomes. Widely adopted due to its analytical tractability, the log-score is particularly beneficial when the predictive density has a known parametric form [25]. The log-score is a strictly proper scoring rule and has several desirable properties, such as consistency and sensitivity to the entire distribution. In addition, the analytical tractability (closed-form expression and gradients for many distributions) makes it a convenient default in deep probabilistic forecasting models. However, for certain distributions (e.g., Gaussian), the log-score severely penalizes unlikely events, rendering it sensitive to outliers and extreme observations [26]. To mitigate this sensitivity, the CRPS provides a robust alternative in univariate contexts [27]. The CRPS quantifies discrepancies between the predictive cumulative distribution function (CDF) and observations, integrating absolute error over all potential thresholds. Unlike the exponential penalty in log-score, CRPS linearly penalizes deviations, thus reducing vulnerability to extreme events [28]. CRPS-based optimization techniques have demonstrated superior calibration and robustness compared to likelihood-based approaches in various probabilistic forecasting applications [28–30]. Minimum CRPS estimation specifically targets improved calibration by optimizing parameters directly to minimize CRPS rather than maximizing likelihood.

Multivariate forecasting introduces additional complexity due to inter-dependencies and higher dimensionality. While the log-score remains applicable, its sensitivity to outliers persists in this setting. The ES [9] generalizes the CRPS for multivariate distributions by computing expected distances between predictive and observed distributions. While ES effectively detects errors in the forecast mean, it is less sensitive to variance errors and, more critically, to misspecifications in the correlation structure among variables [31, 32]. The absence of a closed form expression also necessitates the use of Monte Carlo simulations to approximate the ES by drawing samples from the predictive distribution, which can be computationally expensive [see e.g., 33, 25, 13, 12].

To overcome the limited sensitivity of ES to the dependence structure, the variogram score (VS) was proposed by Scheuerer and Hamill [34]. VS explicitly targets inter-variable dependencies by comparing pairwise differences between forecasted and observed components. Similar to the ES, VS is typically approximated using ensemble forecasts or Monte Carlo sampling. However, it introduces additional computational complexity and still lacks a fully closed-form expression, limiting its direct applicability in large-scale or real-time settings. For a broader discussion of multivariate scoring rules and their properties, we refer readers to the comprehensive reviews by Gneiting and Katzfuss [35], Ziel and Berk [36], Waghmare and Ziegel [37] and Pic et al. [38].

The most relevant recent work is by Olafsdottir et al. [39], who propose a parameter estimation framework for multivariate spatial models by maximizing the average leave-one-out score (LOOS). Their method leverages the tractable conditionals of multivariate Gaussians and robust scoring rules like the CRPS. It is especially efficient for models with sparse precision matrices (e.g., Gaussian Markov random fields), but incurs notable overhead for general multivariate Gaussians due to the cost of computing all conditionals.

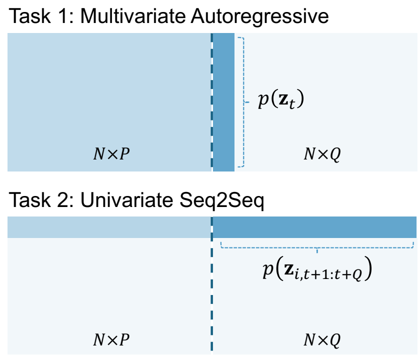

Figure 2: Illustration of the multivariate autoregressive and univariate Seq2Seq forecasting tasks.

3 Our Method

3.1 Multivariate Probabilistic Forecasting

Probabilistic forecasting aims to estimate the joint distribution over a collection of future quantities based on a given history of observations [35]. Denote the time series vector at a time point t as $\mathbf{z}_t = [z_{1,t}, \ldots, z_{N,t}]^\top \in \mathbb{R}^N$, where N is the number of series. The problem of probabilistic forecasting can be formulated as $p\left(\mathbf{z}_{T+1:T+Q} \mid \mathbf{z}_{T-P+1:T}; \mathbf{x}_{T-P+1:T+Q}\right)$, where $\mathbf{z}_{t_1:t_2} = [\mathbf{z}_{t_1}, \ldots, \mathbf{z}_{t_2}]$, P is the conditioning range, Q is the prediction range, and T is the time point that splits the conditioning range and prediction range. $\mathbf{x}_t$ are some known covariates for both past and future time steps.

Multivariate probabilistic forecasting can be formulated in different ways. One way is over the time series dimension, where multiple interrelated variables are forecasted simultaneously at each time point. Considering an autoregressive model, where the predicted output is used as input for the next time step, this formulation can be factorized as

$$\begin{aligned} & p\left(\mathbf{z}_{T+1:T+Q} \mid \mathbf{z}_{T-P+1:T}; \mathbf{x}_{T-P+1:T+Q}\right) \\ = & \prod_{t=T+1}^{T+Q} p\left(\mathbf{z}_t \mid \mathbf{z}_{t-P:t-1}; \mathbf{x}_{t-P:t}\right) = \prod_{t=T+1}^{T+Q} p\left(\mathbf{z}_t \mid \mathbf{h}_t\right), \end{aligned} \tag{1}$$

where $\mathbf{h}_t$ is a state vector that encodes all the conditioning information used to generate the distribution parameters, typically via a neural network.

Another option is over the prediction horizon, where forecasts are made across multiple future time steps for one or more variables, capturing temporal dependencies and uncertainties over time. Considering a shared model across different series:

$$p\left(\mathbf{z}_{i,T+1:T+Q} \mid \mathbf{z}_{i,T-P+1:T}; \mathbf{x}_{i,T-P+1:T+Q}\right), \tag{2}$$

where $i = 1, \ldots, N$ denotes the identifier of a particular time series. Since the model outputs forecasts for the entire prediction horizon directly, it is also called a Seq2Seq model. Without loss of generality, we use the first approach as an example to illustrate our method, since both approaches focus on estimating a multivariate distribution $p\left(\mathbf{z}_t\right)$ or $p\left(\mathbf{z}_{i,T+1:T+Q}\right)$ (Fig. 2).

A typical probabilistic forecasting model assumes Gaussian noise; for example, it models $\mathbf{z}_t$ as jointly following a multivariate Gaussian distribution:

$$\mathbf{z}_t \mid \mathbf{h}_t \sim \mathcal{N}\left(\boldsymbol{\mu}(\mathbf{h}_t), \boldsymbol{\Sigma}(\mathbf{h}_t)\right), \tag{3}$$

where $\boldsymbol{\mu}(\cdot)$ and $\boldsymbol{\Sigma}(\cdot)$ are the functions mapping $\mathbf{h}_t$ to the mean and covariance parameters. The log-likelihood of the distribution given observed time series data up to time point T can be used as the loss function for optimizing a DL model:

$$\mathcal{L} = \sum_{t=1}^{T} \log p\left(\mathbf{z}_t \mid \theta\left(\mathbf{h}_t\right)\right) \propto \sum_{t=1}^{T} -\frac{1}{2}[\ln|\boldsymbol{\Sigma}_t| + \boldsymbol{\eta}_t^\top \boldsymbol{\Sigma}_t^{-1} \boldsymbol{\eta}_t], \tag{4}$$

where $\boldsymbol{\eta}_t = \mathbf{z}_t - \boldsymbol{\mu}_t$. The above formulation simplifies to the univariate case when we set $N = 1$ for the model, with the same model being shared across all time series:

$$z_{i,t} \mid \mathbf{h}_{i,t} \sim \mathcal{N}\left(\mu(\mathbf{h}_{i,t}), \sigma^2(\mathbf{h}_{i,t})\right), \tag{5}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ map $\mathbf{h}_{i,t}$ to the mean and standard deviation of a Gaussian distribution. The corresponding log-likelihood becomes

$$\mathcal{L} = \sum_{t=1}^{T}\sum_{i=1}^{N} \log p\left(z_{i,t} \mid \theta\left(\mathbf{h}_{i,t}\right)\right) \propto \sum_{t=1}^{T}\sum_{i=1}^{N} -\frac{1}{2}\epsilon_{i,t}^2 - \ln\sigma_{i,t}, \tag{6}$$

where $\epsilon_{i,t} = \frac{z_{i,t}-\mu_{i,t}}{\sigma_{i,t}}$. Eq. (4) and Eq. (6), when used as scoring rules to optimize the model, are generally referred to as the log-score and are widely employed in probabilistic forecasting.

For univariate problems, the CRPS is also a strictly proper scoring rule, defined as

$$\mathrm{CRPS}(F, z) = \mathbb{E}_F|x - z| - \frac{1}{2}\mathbb{E}_F\left|x - x'\right|, \tag{7}$$

where F is the predictive CDF, z is the observation, and x and x' are independent random variables both associated with F. The CRPS has a closed-form expression when evaluating a Gaussian-distributed variable $z \sim \mathcal{N}\left(\mu, \sigma^2\right)$ [28]:

$$\mathrm{CRPS}\left(\Phi, z\right) = z\left(2\Phi\left(z\right) - 1\right) + 2\varphi\left(z\right) - \frac{1}{\sqrt{\pi}}, \tag{8}$$

$$\mathrm{CRPS}\left(F_{\mu,\sigma}, z\right) = \sigma\,\mathrm{CRPS}\left(\Phi, \frac{z-\mu}{\sigma}\right), \tag{9}$$

where $F_{\mu,\sigma}\left(z\right) = \Phi\left(\frac{z-\mu}{\sigma}\right)$, Φ and φ are the CDF and PDF of the standard Gaussian distribution.

The CRPS has been shown to be a more robust alternative to the log-score as a loss function in many problems [28, 27, 40]. We observe that the log-score can grow arbitrarily large in magnitude when a single outlier disproportionately influences the loss function, owing to the unbounded nature of the logarithmic function (Eq. (4) and Eq. (6)). Additionally, the quadratic form of the error terms in the Gaussian likelihood also makes it sensitive to outliers (e.g., $\epsilon_{i,t}^2$ in Eq. (6)). In contrast, the CRPS evaluates the entire predictive distribution rather than concentrating solely on the likelihood of individual data points (Eq. (8)). Moreover, the CRPS can directly replace the log-score, providing analytical gradients with respect to μ and σ for backpropagation. However, for a multivariate Gaussian distribution, the CRPS does not have a widely used closed-form expression.

3.2 MVG-CRPS as Loss Function for Multivariate Forecasting

In multivariate probabilistic forecasting, proper scoring rules such as the log-score (Eq. (4)) and the ES are used to evaluate predictive performance. The ES generalizes the CRPS to assess probabilistic forecasts of vector-valued random variables [9]:

$$\mathrm{ES}(F, \mathbf{z}) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim F}\|\boldsymbol{x} - \mathbf{z}\|^{\beta} - \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{x},\boldsymbol{x}'\sim F}\|\boldsymbol{x} - \boldsymbol{x}'\|^{\beta}, \tag{10}$$

where $\|\cdot\|$ denotes the Euclidean norm and $\beta = 1$ is commonly used in the literature [23]. With $\beta = 1$, the ES essentially becomes a multivariate extension of the CRPS and grows linearly with respect to the norm, making it less sensitive to outliers compared to the log-score. Since there is no simple closed-form expression for Eq. (10), it is often approximated using Monte Carlo methods, where multiple samples $\{\boldsymbol{x}_i\}_{i=1}^{n}$ are drawn from the forecast distribution to approximate the expected values:

$$\mathrm{ES}(F, \mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}_i - \mathbf{z}\|^{\beta} - \frac{1}{2n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^{\beta}. \tag{11}$$

However, a significant disadvantage of using Eq. (11) as the loss function is that it requires Monte Carlo sampling during the training process, which can substantially slow down training and create noisy gradients.

In this section, we propose the MVG-CRPS, a robust and efficient loss function designed as an alternative for multivariate forecasting. This loss function grows linearly with the prediction error, making it more robust than the log-score. Additionally, it does not require sampling during the training process, rendering it more efficient than the ES.

Our proposed method is based on the whitening transformation of a time series vector that follows a multivariate Gaussian distribution, $\mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$. The whitening process transforms a random vector with a known covariance matrix into a new random vector whose covariance matrix is the identity matrix. As a result, the elements of the transformed vector have unit variance and are uncorrelated. This transformation begins by performing the singular value decomposition (SVD) of the covariance matrix:

$$\boldsymbol{\Sigma}_t = \boldsymbol{U}_t \boldsymbol{S}_t \boldsymbol{U}_t^\top, \tag{12}$$

where $\boldsymbol{S}_t = \operatorname{diag}([\lambda_{1,t}, \ldots, \lambda_{N,t}]^\top)$ is a diagonal matrix containing the eigenvalues of $\boldsymbol{\Sigma}_t$, and $\boldsymbol{U}_t$ is the orthonormal matrix of corresponding eigenvectors. We then define

$$\mathbf{v}_t = \boldsymbol{U}_t^\top (\mathbf{z}_t - \boldsymbol{\mu}_t), \tag{13}$$

where $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{S}_t)$ is a random vector with a uncorrelated multivariate Gaussian distribution, having variances λ_i (i.e., the corresponding eigenvalue) along the diagonal of its covariance matrix. Next, we define

$$\mathbf{w}_t = \boldsymbol{S}_t^{-\frac{1}{2}} \mathbf{v}_t = \boldsymbol{S}_t^{-\frac{1}{2}} \boldsymbol{U}_t^\top (\mathbf{z}_t - \boldsymbol{\mu}_t), \tag{14}$$

where $\mathbf{w}_t$ is a random vector with each element following a standard Gaussian distribution, i.e., $w_{i,t} \sim \mathcal{N}(0, 1)$. We can then apply Eq. (8) individually to each element and formulate the MVG-CRPS mimicking Eq. (9) for multivariate problem:

$$\operatorname{MCRPS}(\Phi_N(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t), \mathbf{z}_t) = \sum_{i=1}^{N} \operatorname{CRPS}(\Phi(0, \lambda_{i,t}), v_{i,t}) = \sum_{i=1}^{N} \sqrt{\lambda_{i,t}} \operatorname{CRPS}(\Phi, w_{i,t}), \tag{15}$$

where $\Phi_N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is thd CDF of multivariate Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

The overall loss function for training the model is then formulated over an observation period T:

$$\mathcal{L} = \sum_{t=1}^{T} \operatorname{MCRPS}(\Phi_N(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t), \mathbf{z}_t). \tag{16}$$

By leveraging PCA whitening, the MVG-CRPS effectively discriminates between differences in both the mean and covariance within the multivariate Gaussian family—whereas the ES may overlook subtle covariance discrepancies and the log-score lacks robustness. The key advantage of MVG-CRPS lies in its ability to exploit the closed-form expression of the univariate CRPS by decorrelating multivariate time series variables via PCA whitening. This transformation enables the evaluation of marginal distributions in an orthogonalized space, where the whitening is derived from the original covariance matrix. As a result, the optimization process preserves and is sensitive to the dependence structure of the original multivariate distribution. Under the Gaussian assumption, MVG-CRPS constitutes a strictly proper scoring rule (see Appendix §A).

4 Experiments

4.1 Datasets and Models

We evaluate MVG-CRPS on two forecasting tasks: multivariate autoregressive forecasting using the RNN-based GPVar [4] and a decoder-only Transformer [41], and univariate Seq2Seq forecasting using the MLP-based N-HiTS model [42].

To generate the distribution parameters for probabilistic forecasting, we employ a Gaussian distribution head based on the hidden state $\mathbf{h}_{i,t}$ produced by the model. Specifically, for the multivariate autoregressive forecasting, following Salinas et al. [4], we parameterize the mean vector as $\boldsymbol{\mu}(\mathbf{h}_t) = [\mu_1(\mathbf{h}_{1,t}), \ldots, \mu_N(\mathbf{h}_{N,t})]^\top \in \mathbb{R}^N$ and adopt a low-rank-plus-diagonal parameterization of the covariance matrix $\boldsymbol{\Sigma}(\mathbf{h}_t) = \boldsymbol{L}_t \boldsymbol{L}_t^\top + \operatorname{diag}(\mathbf{d}_t)$, where $\mathbf{d}_t = [d_1(\mathbf{h}_{1,t}), \ldots, d_N(\mathbf{h}_{N,t})]^\top \in \mathbb{R}_+^N$ and $\boldsymbol{L}_t = [\mathbf{l}_1(\mathbf{h}_{1,t}), \ldots, \mathbf{l}_N(\mathbf{h}_{N,t})]^\top \in \mathbb{R}^{N \times R}$, $R \ll N$ is the rank parameter. Here, $\mu_i(\cdot)$, $d_i(\cdot)$, and $\mathbf{l}_i(\cdot)$ are the mapping functions that generate the mean and covariance parameters for each time series i based on the hidden state $\mathbf{h}_{i=1:N,t}$. This parameterization guarantees that $\mathbf{\Sigma}(\mathbf{h}_t)$ is full-rank, ensuring that the eigen-decomposition in Eq. (12) is always well-defined. In practice, we use shared mapping functions across all time series, denoted as $\mu_i = \tilde{\mu}$, $d_i = \tilde{d}$, and $\mathbf{l}_i = \tilde{\mathbf{l}}$. This parameterization ensures that $\mathbf{\Sigma}(\mathbf{h}_t)$ is positive definite and efficiently parameterized. The diagonal component provides stability, while the low-rank component captures the covariance structure. The Gaussian assumption also enables the use of random subsets of time series (i.e., batch size $B \leq N$) for model optimization in each iteration, making it feasible to apply our method to high-dimensional time series datasets. Similarly, in the univariate Seq2Seq forecasting task, the mean $\boldsymbol{\mu}(\mathbf{h}_i)$ and covariance $\mathbf{\Sigma}(\mathbf{h}_i)$ are defined over the forecast horizon for each specific time series, based on the hidden states $\mathbf{h}_{i,t=T+1:T+Q}$. As a result, we can model the joint distribution $p(\mathbf{z}_{i,T+1:T+Q})$ over the forecasted values. We implemented our models using PyTorch Forecasting [43], with input data consisting of lagged time series values and covariates. Extensive experiments were conducted on a variety of real-world time series datasets from GluonTS [44] (see Appendix §B). Full details of the experimental setup are provided in Appendix §C.

4.2 Toy Example

We first perform a toy experiment following Roordink and Hess [45] using a true distribution $P = \mathcal{N}\left(\begin{bmatrix}1\\-1\end{bmatrix}, \begin{bmatrix}1 & 0.8\\0.8 & 4\end{bmatrix}\right)$ and a predictive distribution $Q = \mathcal{N}\left(\begin{bmatrix}\mu\\-1\end{bmatrix}, \begin{bmatrix}\sigma^2 & 2\rho\sigma\\2\rho\sigma & 4\end{bmatrix}\right)$, where we control the deviation of the three parameters μ, ρ, σ to study the various properties of different scores. As shown in Fig. 3, the log-score increases sharply when the standard deviation σ or correlation coefficient ρ deviate from their true values, indicating high sensitivity to covariance misspecification. The ES shows lower sensitivity to the covariance structure but produces non-smooth curves due to its sample-based approximation. In contrast, the MVG-CRPS displays comparable sensitivity to deviations in all three parameters. It also produces smooth curves with a clear minimum at zero deviation, reflecting its closed-form evaluation.

We further examine the robustness of different scoring rules for estimating the parameters of this predictive distribution under data contamination, and analyze the trade-off between computational cost and estimation accuracy for the ES with varying sample sizes (see Appendix §D.1). Overall, MVG-CRPS demonstrates greater robustness than the log-score across all three parameters, particularly for μ and σ, and provides more consistent estimates than the ES due to its sampling-free formulation (Fig. A1). We also observe that the ES produces less accurate estimates than MVG-CRPS for μ and σ. Although we do not claim superiority over the ES beyond efficiency, this discrepancy is likely attributable to the variance introduced by its Monte Carlo approximation. Additionally, we observe that the gains in estimation accuracy diminish rapidly as the sample size increases, and the ES does not significantly outperform MVG-CRPS even with 1,000 samples (Fig. A2). Meanwhile, the computational cost of the ES increases monotonically with sample size.

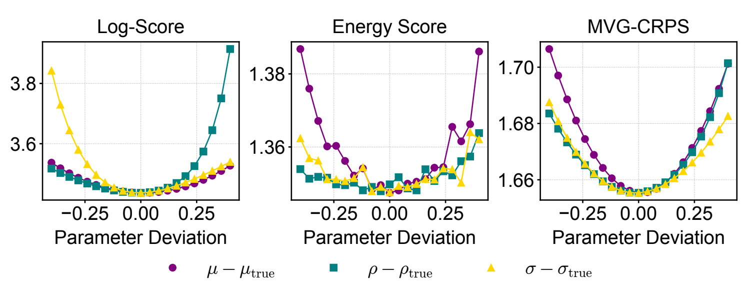

Figure 3: Sensitivity of scoring rules to parameter deviations in the predicted mean, standard deviation, and correlation coefficient from the true data distribution ($\mu_{\text{true}} = 1, \sigma_{\text{true}} = 1, \rho_{\text{true}} = 0.4$). The ES values are computed with a sample size of 500.

4.3 Quantitative Evaluation

We evaluate the MVG-CRPS against models trained with the log-score and the ES using three common metrics for probabilistic forecasts: $\mathrm{CRPS}_{\text{sum}}$, $\mathrm{CRPS}_{\text{mean}}$, and the ES (see Appendix §C.5

Table 1: Comparison of $\mathrm{CRPS}_{\mathrm{sum}}$ across different scoring rules in the multivariate autoregressive forecasting task. The best scores are in boldface. MVG-CRPS scores are underlined when they are not the best overall but exceed the log-score.

	VAR	GPVar			Transformer		
		log-score	energy score	MVG-CRPS	log-score	energy score	MVG-CRPS
`elec_au`	N/A	0.1261±0.0009	**0.0887±0.0004**	0.0967±0.0008	0.1633±0.0005	0.1492±0.0006	**0.0793±0.0004**
`cif_2016`	1.0000±0.0000	0.0122±0.0004	0.0420±0.0006	**0.0111±0.0005**	0.0118±0.0003	0.0240±0.0014	**0.0107±0.0002**
`electricity`	0.1315±0.0006	0.0419±0.0008	0.0616±0.0004	**0.0249±0.0006**	0.0362±0.0002	0.0368±0.0004	**0.0294±0.0004**
`elec_weekly`	0.1126±0.0011	0.1515±0.0028	**0.0417±0.0014**	0.0772±0.0031	0.0937±0.0026	**0.0403±0.0013**	0.0448±0.0014
`exchange_rate`	0.0033±0.0000	0.0207±0.0004	**0.0030±0.0001**	0.0041±0.0001	**0.0047±0.0003**	0.0067±0.0003	0.0091±0.0004
`kdd_cup`	N/A	0.3743±0.0019	0.3210±0.0019	**0.2358±0.0014**	0.2076±0.0013	0.4789±0.0030	**0.1959±0.0017**
`m1_yearly`	N/A	0.4397±0.0041	0.4801±0.0022	**0.3566±0.0029**	0.5344±0.0109	**0.3291±0.0047**	0.4563±0.0111
`m3_yearly`	N/A	0.3607±0.0084	0.2186±0.0042	**0.1423±0.0053**	0.3156±0.0102	0.4050±0.0061	**0.2325±0.0094**
`nn5_daily`	0.2303±0.0005	0.0998±0.0004	0.0958±0.0003	**0.0948±0.0003**	0.0991±0.0003	0.0883±0.0004	**0.0811±0.0002**
`saugeenday`	N/A	0.4040±0.0047	**0.3733±0.0048**	0.3941±0.0055	0.3771±0.0088	**0.3689±0.0053**	0.3705±0.0047
`sunspot`	N/A	18.7115±1.3296	23.3988±0.9662	**17.2438±0.5833**	39.7454±1.4841	**16.6556±0.6167**	22.6495±0.6752
`tourism`	0.1394±0.0012	0.2217±0.0027	0.2112±0.0014	**0.2004±0.0022**	0.2100±0.0017	0.2087±0.0020	**0.2082±0.0015**
`traffic`	3.5241±0.0084	0.0742±0.0004	**0.0505±0.0002**	0.0868±0.0002	**0.0658±0.0002**	0.0667±0.0002	0.0683±0.0000
Avg. Rank		2.62	1.92	**1.46**	2.38	2.00	**1.62**

Table 2: Training time (in minutes) for GPVar using different scoring rules in the multivariate autoregressive forecasting task. Reported times include early stopping and reflect differences in convergence speed across loss functions.

	log-score		energy score		MVG-CRPS	
	per epoch	total	per epoch	total	per epoch	total
`elec_au`	0.86	33.53	16.29	717.00	**0.78**	**29.14**
`cif_2016`	0.13	**1.58**	4.83	401.04	**0.12**	3.85
`electricity`	0.40	67.38	11.17	782.40	**0.38**	**22.70**
`elec_weekly`	0.30	**14.61**	10.95	383.52	**0.26**	18.77
`exchange_rate`	**0.25**	**16.40**	10.20	663.60	0.29	23.63
`kdd_cup`	**0.42**	**11.32**	14.23	2063.52	**0.42**	28.79
`m1_yearly`	0.19	**3.71**	5.66	469.92	**0.18**	8.02
`m3_yearly`	0.43	**7.30**	10.80	291.72	**0.42**	14.49
`nn5_daily`	0.29	**9.21**	11.64	244.50	**0.27**	14.53
`saugeenday`	0.23	**12.65**	10.70	524.46	**0.15**	15.32
`sunspot`	0.44	26.85	10.73	397.26	**0.42**	**16.96**
`tourism`	0.49	23.96	10.56	243.00	**0.46**	**12.51**
`traffic`	0.94	**76.98**	14.92	1044.60	**0.92**	92.46

for definitions). Table 1 presents a comparison of $\mathrm{CRPS}_{\mathrm{sum}}$ for the multivariate autoregressive forecasting task. Overall, the MVG-CRPS achieves the best average rank among the three scoring rules. Notably, it consistently outperforms the log-score across most datasets, indicating that MVG-CRPS leads to models with higher-quality forecasts. As shown in later sections, this improvement is attributed to MVG-CRPS being less sensitive to outliers. Compared to the ES, MVG-CRPS achieves comparable or better performance (Table 1) while being more efficient during training (Table 2). It is important to note that we do not claim MVG-CRPS is more robust than ES; rather, our focus is on its efficiency compared to ES. Results for $\mathrm{CRPS}_{\mathrm{mean}}$ and the ES are provided in Appendix §D.2, and results for the univariate Seq2Seq forecasting task are presented in Appendix §D.3. In both tasks, MVG-CRPS achieves consistent performance across all three evaluation metrics.

4.4 Qualitative Evaluation

To illustrate the robustness of MVG-CRPS, we compare the output covariance matrices from models trained with different loss functions and visualize their probabilistic forecasts. In Fig. 4, the log-score model produces covariance matrices that occasionally exhibit large covariances, despite normalization applied to each time series. This behavior likely reflects the influence of large tail errors during training. In contrast, the MVG-CRPS model captures similar covariance patterns without extreme values, indicating improved robustness to outliers. To highlight the practical impact, we compare GPVar forecasts on the `electricity` dataset (Fig. 5). MVG-CRPS yields sharper and better-calibrated predictions, while the log-score model occasionally produces overly wide intervals,

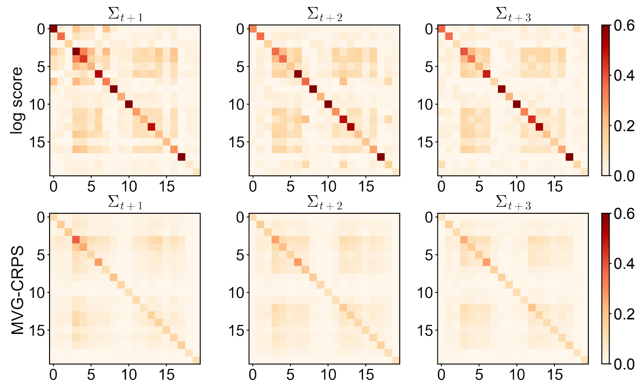

Figure 4: Comparison of output covariance matrices $\mathbf{\Sigma}_t$ from GPVar on the `elec_weekly` dataset. For visual clarity, covariance values are clipped between 0 and 0.6.

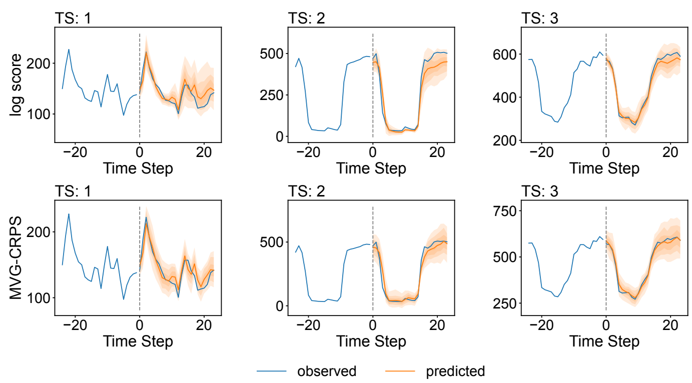

Figure 5: Comparison of probabilistic forecasts from GPVar on the `electricity` dataset.

reflecting greater sensitivity to outliers (e.g., TS 1). Results for the univariate Seq2Seq forecasting task are provided in Appendix §D.3.

5 Conclusion

This paper introduced the MVG-CRPS, a novel strictly proper scoring rule specifically designed for multivariate Gaussian probabilistic forecasting. MVG-CRPS addresses the sensitivity of the log-score to outliers and overcomes the computational inefficiency inherent to the ES. By applying a whitening transformation and leveraging the closed-form expression of the univariate CRPS, our approach achieves robustness to extreme values while remaining computationally efficient and easily integrable into deep learning frameworks. Moreover, the MVG-CRPS exhibits high sensitivity to both the mean and covariance of the predictive distribution—comparable to the log-score—while preserving the robustness properties of the ES. Empirical evaluations on real-world datasets demonstrated significant improvements in both predictive accuracy and robustness compared to existing scoring rules.

Beyond forecasting, the general formulation of MVG-CRPS extends naturally to broader probabilistic regression contexts, such as robust Gaussian process regression, by replacing conventional negative marginal likelihood objectives. Future directions include leveraging copula transformations to extend the MVG-CRPS to non-Gaussian distributions and exploring more efficient covariance parameterizations to enhance scalability. Currently, scalability remains constrained by the computational demands of eigen-decomposition in large-batch scenarios. A possible solution to mitigate this limitation is to adopt an isotropic noise parameterization, i.e., $\mathbf{\Sigma} = \boldsymbol{L}\boldsymbol{L}^\top + \sigma^2\mathbf{I}$, which enables more efficient computation of the SVD.

References

[1] Jan JJ Groen, Richard Paap, and Francesco Ravazzolo. Real-time inflation forecasting in a changing world. *Journal of Business & Economic Statistics*, 31(1):29–44, 2013.

[2] TN Palmer. Towards the probabilistic earth-system simulator: A vision for the future of climate and weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 138(665):841–861, 2012.

[3] Hayley E Jones and David J Spiegelhalter. Improved probabilistic prediction of healthcare performance indicators using bidirectional smoothing models. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 175(3):729–747, 2012.

[4] David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank gaussian copula processes. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] Konstantinos Benidis, Syama Sundar Rangapuram, Valentin Flunkert, Yuyang Wang, Danielle Maddix, Caner Turkmen, Jan Gasthaus, Michael Bohlke-Schneider, David Salinas, Lorenzo Stella, et al. Deep learning for time series forecasting: Tutorial and literature survey. *ACM Computing Surveys*, 55(6):1–36, 2022.

[6] Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. In *International Conference on Learning Representations*, 2021.

[7] Manuel Gebetsberger, Jakob W Messner, Georg J Mayr, and Achim Zeileis. Estimation methods for nonhomogeneous regression models: Minimum continuous ranked probability score versus maximum likelihood. *Monthly Weather Review*, 146(12):4323–4338, 2018.

[8] Mathias Blicher Bjerregård, Jan Kloppenborg Møller, and Henrik Madsen. An introduction to multivariate probabilistic forecast evaluation. *Energy and AI*, 4:100058, 2021.

[9] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

[10] James E Matheson and Robert L Winkler. Scoring rules for continuous probability distributions. *Management Science*, 22(10):1087–1096, 1976.

[11] Tilmann Gneiting and Adrian E Raftery. Weather forecasting with ensemble methods. *Science*, 310(5746): 248–249, 2005.

[12] Lorenzo Pacchiardi, Rilwan A Adewoyin, Peter Dueben, and Ritabrata Dutta. Probabilistic forecasting with generative networks via scoring rule minimization. *Journal of Machine Learning Research*, 25(45): 1–64, 2024.

[13] Jieyu Chen, Tim Janke, Florian Steinke, and Sebastian Lerch. Generative machine learning methods for multivariate ensemble postprocessing. *The Annals of Applied Statistics*, 18(1):159–183, 2024.

[14] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017.

[15] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[16] Vincent Zhihao Zheng, Seongjin Choi, and Lijun Sun. Better batch for deep probabilistic time series forecasting. In *International Conference on Artificial Intelligence and Statistics*, pages 91–99, 2024.

[17] Vincent Zhihao Zheng and Lijun Sun. Multivariate probabilistic time series forecasting with correlated errors. *Advances in Neural Information Processing Systems*, 37, 2024.

[18] Vincent Zhihao Zheng, Seongjin Choi, and Lijun Sun. Probabilistic traffic forecasting with dynamic regression. *Transportation Science*, 2025.

[19] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. *Advances in Neural Information Processing Systems*, 31, 2018.

[20] Emmanuel de Bézenac, Syama Sundar Rangapuram, Konstantinos Benidis, Michael Bohlke-Schneider, Richard Kurle, Lorenzo Stella, Hilaf Hasson, Patrick Gallinari, and Tim Januschowski. Normalizing kalman filters for multivariate time series analysis. *Advances in Neural Information Processing Systems*, 33:2995–3007, 2020.

[21] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pages 8857–8868, 2021.

[22] Alexandre Drouin, Étienne Marcotte, and Nicolas Chapados. Tactis: Transformer-attentional copulas for time series. In *International Conference on Machine Learning*, pages 5447–5493, 2022.

[23] Arjun Ashok, Étienne Marcotte, Valentina Zantedeschi, Nicolas Chapados, and Alexandre Drouin. Tactis-2: Better, faster, simpler attentional copulas for multivariate time series. In *International Conference on Learning Representations*, 2024.

[24] Jochen Bröcker. Reliability, sufficiency, and the decomposition of proper scores. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 135(643):1512–1519, 2009.

[25] Anastasios Panagiotelis, Puwasala Gamakumara, George Athanasopoulos, and Rob J Hyndman. Probabilistic forecast reconciliation: Properties, evaluation and score optimisation. *European Journal of Operational Research*, 306(2):693–706, 2023.

[26] Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 69(2):243–268, 2007.

[27] Stephan Rasp and Sebastian Lerch. Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review*, 146(11):3885–3900, 2018.

[28] Tilmann Gneiting, Adrian E Raftery, Anton H Westveld, and Tom Goldman. Calibrated probabilistic forecasting using ensemble model output statistics and minimum crps estimation. *Monthly Weather Review*, 133(5):1098–1118, 2005.

[29] Kin G Olivares, Geoffrey Négiar, Ruijun Ma, O Nangba Meetei, Mengfei Cao, and Michael W Mahoney. Probabilistic forecasting with coherent aggregation. *arXiv preprint arXiv:2307.09797*, 2023.

[30] Simon Lang, Mihai Alexe, Mariana CA Clare, Christopher Roberts, Rilwan Adewoyin, Zied Ben Bouallègue, Matthew Chantry, Jesper Dramsch, Peter D Dueben, Sara Hahner, et al. Aifs-crps: Ensemble forecasting using a model trained with a loss function based on the continuous ranked probability score. *arXiv preprint arXiv:2412.15832*, 2024.

[31] Pierre Pinson and Julija Tastu. Discrimination ability of the energy score. 2013.

[32] Carol Alexander, Michael Coulon, Yang Han, and Xiaochun Meng. Evaluating the discrimination ability of proper multi-variate scoring rules. *Annals of Operations Research*, 334(1):857–883, 2024.

[33] Diane Bouchacourt, Pawan K Mudigonda, and Sebastian Nowozin. Disco nets: Dissimilarity coefficients networks. *Advances in Neural Information Processing Systems*, 29, 2016.

[34] Michael Scheuerer and Thomas M Hamill. Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities. *Monthly Weather Review*, 143(4):1321–1334, 2015.

[35] Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1(1):125–151, 2014.

[36] Florian Ziel and Kevin Berk. Multivariate forecasting evaluation: On sensitive and strictly proper scoring rules. *arXiv preprint arXiv:1910.07325*, 2019.

[37] Kartik Waghmare and Johanna Ziegel. Proper scoring rules for estimation and forecast evaluation. *arXiv preprint arXiv:2504.01781*, 2025.

[38] Romain Pic, Clément Dombry, Philippe Naveau, and Maxime Taillardat. Proper scoring rules for multivariate probabilistic forecasts based on aggregation and transformation. *Advances in Statistical Climatology, Meteorology and Oceanography*, 11(1):23–58, 2025.

[39] Helga Kristin Olafsdottir, Holger Rootzén, and David Bolin. Fast and robust cross-validation-based scoring rule inference for spatial statistics. *arXiv preprint arXiv:2408.11994*, 2024.

[40] Abdulmajid Murad, Frank Alexander Kraemer, Kerstin Bach, and Gavin Taylor. Probabilistic deep learning to quantify uncertainty in air quality forecasting. *Sensors*, 21(23):8009, 2021.

[41] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[42] Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6989–6997, 2023.

[43] Jan Beitner. Pytorch forecasting. `https://pytorch-forecasting.readthedocs.io`, 2020.

[44] Alexander Alexandrov, Konstantinos Benidis, Michael Bohlke-Schneider, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Danielle C Maddix, Syama Rangapuram, David Salinas, Jasper Schulz, et al. Gluonts: Probabilistic and neural time series modeling in python. *The Journal of Machine Learning Research*, 21 (1):4629–4634, 2020.

[45] Daan Roordink and Sibylle Hess. Scoring rule nets: Beyond mean target prediction in multivariate regression. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 190–205. Springer, 2023.

[46] Alfred Horn. Doubly stochastic matrices and the diagonal of a rotation matrix. *American Journal of Mathematics*, 76(3):620–630, 1954.

[47] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge University Press, 2012.

[48] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2021.

[49] Helmut Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer Science & Business Media, 2005.

Appendix

Table of Contents

A MVG-CRPS is Strictly Proper

Theorem A.1. *Let $\mathbf{z} \sim \mathcal{N}\left(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p\right)$ be a true N-variate Gaussian distribution where the covariance admits eigen-decomposition $\boldsymbol{\Sigma}_p = \boldsymbol{U}_p \boldsymbol{S}_p \boldsymbol{U}_p^\top$, with $\boldsymbol{S}_p = \operatorname{diag}\left(\boldsymbol{\lambda}_p\right)$ containing nonincreasing eigenvalues $\boldsymbol{\lambda}_p = [\lambda_1^p, \ldots, \lambda_N^p]^\top$ and $\boldsymbol{U}_p$ being the corresponding orthonormal matrix. Consider a predictive Gaussian distribution $\mathcal{N}\left(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\right)$, where covariance $\boldsymbol{\Sigma}_q$ admits the eigen-decomposition $\boldsymbol{\Sigma}_q = \boldsymbol{U}_q \boldsymbol{S}_q \boldsymbol{U}_q^\top$ with $\boldsymbol{S}_q = \operatorname{diag}\left(\boldsymbol{\lambda}_q\right)$. Define the transformed variable $\mathbf{v} = \boldsymbol{U}_q^\top\left(\mathbf{z} - \boldsymbol{\mu}_q\right) = [v_1, \ldots, v_N]^\top$. The proposed MVG-CRPS*

$$\operatorname{MCRPS}\left(\Phi_N\left(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\right), \mathbf{z}\right) = \sum_{i=1}^{N} \operatorname{CRPS}\left(\Phi\left(0, \lambda_i^q\right), v_i\right)$$

is proper and strictly proper for multivariate Gaussian distributions.

Proof. Given that $\mathbf{z} \sim \mathcal{N}\left(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p\right)$, we have the transformed variable $\mathbf{v} \sim \mathcal{N}\left(\boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v\right)$ with $\boldsymbol{\mu}_v = \boldsymbol{U}_q^\top(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q) = [\nu_1, \ldots, \nu_N]^\top$ and $\boldsymbol{\Sigma}_v = \boldsymbol{U}_q^\top \boldsymbol{\Sigma}_p \boldsymbol{U}_q = \boldsymbol{U}_q^\top \boldsymbol{U}_p \boldsymbol{S}_p \boldsymbol{U}_p^\top \boldsymbol{U}_q = \boldsymbol{U}_v \boldsymbol{S}_p \boldsymbol{U}_v^\top$, where $\boldsymbol{U}_v = \boldsymbol{U}_q^\top \boldsymbol{U}_p$ is an orthonormal matrix. Thus, each v_i has a marginal distribution $v_i \sim \mathcal{N}\left(\nu_i, \tau_i\right)$ for $i = 1, \ldots, N$, with $\boldsymbol{\tau} = \operatorname{diag}(\boldsymbol{\Sigma}_v) = \operatorname{diag}(\boldsymbol{U}_v \boldsymbol{S}_p \boldsymbol{U}_v^\top) = [\tau_1, \ldots, \tau_N]^\top$. Taking the expectation of $\mathrm{MCRPS}\left(\Phi_N\left(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\right), \mathbf{z}\right)$ under the true distribution, we have

$$\begin{aligned}
\mathbb{E}_{\mathbf{z}\sim\mathcal{N}\left(\boldsymbol{\mu}_p,\boldsymbol{\Sigma}_p\right)}\left[\mathrm{MCRPS}\left(\Phi_N\left(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q\right), \mathbf{z}\right)\right] &= \sum_{i=1}^{N} \mathbb{E}_{v_i\sim\mathcal{N}(\nu_i,\tau_i)}\left[\mathrm{CRPS}\left(\Phi\left(0, \lambda_i^q\right), v_i\right)\right] \\
&\geq \sum_{i=1}^{N} \mathbb{E}_{v_i\sim\mathcal{N}(\nu_i,\tau_i)}\left[\mathrm{CRPS}\left(\Phi\left(\nu_i, \tau_i\right), v_i\right)\right] \\
&= \sum_{i=1}^{N} \mathbb{E}_{\eta_i\sim\mathcal{N}(0,\tau_i)}\left[\mathrm{CRPS}\left(\Phi\left(0, \tau_i\right), \eta_i\right)\right] \\
&= \mathbb{E}_{v\sim\mathcal{N}(0,1)}\left[\mathrm{CRPS}\left(\Phi, v\right)\right] \times \sum_{i=1}^{N} \sqrt{\tau_i} \\
&\geq \mathbb{E}_{v\sim\mathcal{N}(0,1)}\left[\mathrm{CRPS}\left(\Phi, v\right)\right] \times \sum_{i=1}^{N} \sqrt{\lambda_i^p} \\
&= \mathbb{E}_{\mathbf{z}\sim\mathcal{N}\left(\boldsymbol{\mu}_p,\boldsymbol{\Sigma}_p\right)}\left[\mathrm{MCRPS}\left(\Phi_N\left(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p\right), \mathbf{z}\right)\right].
\end{aligned} \tag{17}$$

The first inequality is a direct result of CRPS being a strictly proper scoring rule for univariate Gaussian distributions. We now prove the second inequality.

Recall that $\boldsymbol{\tau} = \mathrm{diag}(\boldsymbol{\Sigma}_v)$ and $\boldsymbol{\Sigma}_v = \boldsymbol{U}_q^\top \boldsymbol{\Sigma}_p \boldsymbol{U}_q$. Let $\boldsymbol{\tau}^*$ be the monotone nonincreasing rearrangement of $\boldsymbol{\tau}$. By the Schur-Horn theorem [46], the diagonal vector $\boldsymbol{\tau}^*$ is majorized by the eigenvalues $\boldsymbol{\lambda}_p$:

$$\sum_{i=1}^{k} \tau_i^* \leq \sum_{i=1}^{k} \lambda_i^p,$$

for $k = 1, 2, \ldots, N-1$, and

$$\sum_{i=1}^{N} \tau_i^* = \sum_{i=1}^{N} \lambda_i^p.$$

Since $f(x) = \sqrt{x}$ is a concave function, Karamata's majorization inequality yields

$$\sum_{i=1}^{N} \sqrt{\lambda_i^p} \leq \sum_{i=1}^{N} \sqrt{\tau_i^*} = \sum_{i=1}^{N} \sqrt{\tau_i}, \tag{18}$$

which proves the second inequality in Eq. (17). Hence, the MVG-CRPS is a proper scoring rule for the multivariate Gaussian distribution.

Equality in Eq. (18) is obtained if, for every i, $\tau_i^* = \lambda_i^p$. By the Schur-Horn theorem, this forces $\boldsymbol{\Sigma}_v$ to be a diagonal matrix (Theorem 4.3.45 in Horn and Johnson [47]). Meanwhile, the CRPS inequality in Eq. (17) is tight exactly when, for every i, $\nu_i = 0$ and $\tau_i = \lambda_i^q$, implying that $\boldsymbol{U}_q^\top(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q) = \mathbf{0}$ and $\mathrm{diag}(\boldsymbol{\Sigma}_v) = \mathrm{diag}(\boldsymbol{S}_{\boldsymbol{q}})$. Since $\boldsymbol{\Sigma}_v$ is diagonal, we have $\boldsymbol{\Sigma}_v = \boldsymbol{U}_q^\top \boldsymbol{\Sigma}_p \boldsymbol{U}_q = \boldsymbol{S}_q$, hence $\boldsymbol{\Sigma}_p = \boldsymbol{\Sigma}_q$. Therefore, all equalities hold if and only if $\boldsymbol{\mu}_p = \boldsymbol{\mu}_q$ and $\boldsymbol{\Sigma}_p = \boldsymbol{\Sigma}_q$. This confirms that the proposed scoring rule is proper and strictly proper for the multivariate Gaussian distribution. □

B Dataset Details

We conducted experiments on a diverse collection of real-world datasets sourced from GluonTS [44]. These datasets are commonly used for benchmarking time series forecasting models, following their default configurations in GluonTS, which include granularity, prediction horizon (Q), and the number of rolling evaluations. For each dataset, we sequentially split the data into training, validation, and testing sets, ensuring that the temporal length of the validation set matched that of the testing set. The temporal length of the testing set was based on the prediction horizon and the required number of rolling evaluations. For example, the testing horizon for the `traffic` dataset is calculated as

$24 + 7 - 1 = 30$ time steps. Consequently, the model generates 24-step predictions (Q) sequentially, with 7 distinct consecutive prediction start points, corresponding to 7 forecast instances. In our experiments, we aligned the conditioning range (P) with the prediction horizon (Q), consistent with the default setting in GluonTS (i.e., $P = Q$). Each time series was individually normalized using a scaler fitted to its own training data [15, 48]. Predictions were then rescaled to their original values for computing evaluation metrics. Table A1 summarizes the statistics of all datasets.

Table A1: Dataset summary.

Dataset	Granularity	# of time series	# of time steps	Q	Rolling evaluation
`elec_au`	30min	5	232,272	60	56
`cif_2016`	monthly	72	120	12	1
`electricity`	hourly	370	5,857	24	7
`elec_weekly`	weekly	321	156	8	3
`exchange_rate`	workday	8	6,101	30	5
`kdd_cup`	hourly	270	10,920	48	7
`m1_yearly`	yearly	181	169	6	1
`m3_yearly`	yearly	645	191	6	1
`nn5_daily`	daily	111	791	56	5
`saugeenday`	daily	1	23,741	30	5
`sunspot`	daily	1	73,924	30	5
`tourism`	quarterly	427	131	8	1
`traffic`	hourly	963	4,025	24	7
`covid`	daily	266	212	30	5
`elec_hourly`	hourly	321	26,304	48	7
`m4_hourly`	hourly	414	1,008	48	7
`pedestrian`	hourly	66	96,432	48	7
`taxi_30min`	30min	1214	1,637	24	56
`uber_hourly`	hourly	262	8,343	24	7
`wiki`	daily	2000	792	30	5

C Experiment Details

C.1 Benchmark Models

The input to benchmark models includes lagged time series values and covariates that encode time and series identification. The number of lagged values is determined by the granularity of each dataset. Specifically, we use lags of $\{1, 24, 168\}$ for hourly data, $\{1, 7, 14\}$ for daily data, and $\{1, 2, 4, 12, 24, 48\}$ for data with sub-hourly granularity. For all other datasets, only lag-1 values are used.

For datasets with hourly or finer granularity, we include the hour of the day and day of the week. For daily datasets, only the day of the week is used. Each time series is uniquely identified by a numeric identifier. All features are encoded as single values; for example, the hour of the day takes values between [0, 23]. These features are concatenated with the model input at each time step to form the model input vector $\mathbf{y}_t$ [4, 17].

Our method requires a state vector $\mathbf{h}_{i,t}$ to generate the parameters for the predictive distribution. To achieve this, we employ different neural architectures: RNNs and Transformer decoders, both of which maintain autoregressive properties for the multivariate autoregressive forecasting task, and MLPs for the univariate Seq2Seq forecasting task. Specifically, we use the GPVar model [4] as our RNN benchmark, the GPT model [41] for the decoder-only Transformer, and the N-HiTS model [42] for the MLPs. All models are trained to output $\mathbf{h}_{i,t}$, which is used to parameterize the predictive distribution.

C.2 Naive Baseline Description

In this paper, we use Vector Autoregression (VAR) [49] as a naive baseline model. The VAR(p) model is formulated as

$$\mathbf{z}_t = \mathbf{c} + \boldsymbol{A}_1 \mathbf{z}_{t-1} + \cdots + \boldsymbol{A}_p \mathbf{z}_{t-p} + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\epsilon), \tag{19}$$

where A_i is an $N \times N$ coefficient matrix, and $\mathbf{c}$ is the intercept term. In our experiments, we employ a VAR model with a lag of 1 (VAR(1)). The parameters in Eq. (19) are estimated using ordinary least squares (OLS), as described in Lütkepohl [49]. VAR models are not applied to datasets with insufficient time series in the testing set and are marked as "N/A" in this paper.

C.3 Hyperparameters

All model parameters are optimized using the Adam optimizer with l_2 regularization set to $1\mathrm{e}^{-8}$, and gradient clipping applied at 10.0. For all methods, we cap the total number of gradient updates at 10,000 and reduce the learning rate by a factor of 2 after 500 consecutive updates without improvement. Table A2 provides the hyperparameter values that remain fixed across all datasets. In the main manuscript, we do **NOT** tune the hyperparameters specifically to favor the proposed loss. Instead, we use the same hyperparameters as those in GPVar [4], which were originally tuned for the log-score. Keeping the hyperparameters consistent across loss functions ensures that any observed improvements are attributable to the loss function itself rather than differences in hyperparameter settings. However, we conduct additional studies using hyperparameters tuned for each loss function in §D.4.

Table A2: Hyperparameters values.

Hyperparameter	Value
learning rate	1e-3
hidden size	40
`n_layers` (RNN/Transformer decoder/MLP)	2
`n_heads` (Transformer)	2
rank (R)	10
sampling dimension (B)	20
dropout	0.01
batch size	16

C.4 Training Procedure

Compute Resources All models were trained in an Anaconda environment using one AMD Ryzen Threadripper PRO 5955WX CPU and four NVIDIA RTX A5000 GPUs, each with 24 GB of memory.

Batch Size Following the method used in GPVar [4], we set the sample slice size to $B = 20$ time series and used a batch size of 16. Since our data sampler processes one slice of time series at a time rather than sampling 16 slices simultaneously, we set `accumulate_grad_batches` to 16, effectively achieving a batch size of 16.

Training Loop During each epoch, the model is trained on up to 400 batches from the training set, followed by the computation of the `valid_loss` on the validation set. Training is halted when one of the following conditions is met:

- A total of 10,000 gradient updates has been reached,
- No improvement in the validation set `valid_loss` is observed for 10 consecutive epochs.

The final model is the one that achieves the lowest `valid_loss` on the validation set.

Covariance Parameterization The covariance matrix $\mathbf{\Sigma}_t$ is parameterized directly by the forecasting model. Specifically, it is constructed as: $\mathbf{\Sigma}_t = \boldsymbol{L}_t \boldsymbol{L}_t^\top + \mathrm{diag}(\mathbf{d}_t)$, where $\boldsymbol{L}_t$ is a low-rank matrix and $\mathbf{d}_t$ is a positive diagonal vector. This parameterization ensures that $\mathbf{\Sigma}_t$ remains positive semi-definite while being computationally efficient to learn. This parameterization is standard in probabilistic forecasting and allows the model to learn both the structure (through $\boldsymbol{L}_t$) and scale (through $\mathbf{d}_t$) of the covariance during training. Without constraints, the MVG-CRPS loss could potentially be minimized by driving all eigenvalues of $\mathbf{\Sigma}_t$ to zero, resulting in a trivial solution. However, this is prevented through the following mechanisms:

- The diagonal entries of the covariance matrix are parameterized as $d_{i,t} = \texttt{softplus}(d_{i,t} + \texttt{diag\_bias}) + \sigma^2_{\text{min}}$, where the `softplus` function ensures that the diagonal entries are strictly positive, regardless of the raw input values, `diag_bias` is initialized to approximately `softplus_inv`(σ^2_{init}), ensuring that the diagonal entries are initially close to σ^2_{init}. For instance, with $\sigma_{\text{init}} = 1.0$, the initial diagonal values start near 1.0. The addition of σ^2_{min} provides a lower bound on the diagonal entries, ensuring that eigenvalues cannot approach zero.

- The low-rank component is parameterized as $\mathbf{L}_{i,t} = \frac{\mathbf{L}_{i,t}}{\sqrt{R}}$, where dividing by rank ensures that the low-rank term is well-scaled relative to the diagonal entries. This normalization prevents the low-rank component from dominating or becoming disproportionately small in the covariance matrix.

Moreover, the MVG-CRPS loss provides a balance between the calibration and sharpness of the forecasts:

$$\mathbf{w}_t = \boldsymbol{S}_t^{-\frac{1}{2}}\mathbf{v}_t = \boldsymbol{S}_t^{-\frac{1}{2}}\boldsymbol{U}_t^{\top}\left(\mathbf{z}_t - \boldsymbol{\mu}_t\right),$$

$$\mathcal{L} = \sum_{t=1}^{T}\sum_{i=1}^{N}\sqrt{\lambda_t^i}\,\mathrm{CRPS}\left(\Phi, w_{i,t}\right).$$

We observe that if the eigenvalues λ_t^i in $\boldsymbol{S}_t$ approach zero, $w_{i,t}$ will be scaled very aggressively. This leads to inflated residuals $w_{i,t}$, which subsequently affect the CRPS computation. Since the CRPS metric integrates over the forecast distribution $F(y)$, penalizing deviations between $F(y)$ and the empirical step function $\mathbf{1}(y \geq w_{i,t})$, artificially large $w_{i,t}$ values (resulting from extreme eigenvalue scaling) will cause the CRPS term to increase significantly. This behavior reflects the importance of ensuring that eigenvalues λ_t^i are well-regularized to prevent distortion in the forecast evaluation. By balancing the eigenvalue contributions, the MVG-CRPS ensures both stable calibration and sharpness in probabilistic forecasting.

SVD and Gradient Calculation We perform SVD on $\boldsymbol{\Sigma}(\mathbf{h}_t)$ to obtain $\mathbf{U}_t$ and $\mathbf{S}_t$ (the eigenvectors and eigenvalues, respectively). These are required to compute the whitening transformation: $\mathbf{w}_t = \mathbf{S}_t^{-\frac{1}{2}}\mathbf{U}_t^{\top}(\mathbf{z}_t - \boldsymbol{\mu}_t)$. During training, gradients of $\mathcal{L}$ need to flow back through the whitened vecotr $\mathbf{w}_t$, the eigenvectors matrix $\mathbf{U}_t$, the eigenvalues matrix $\mathbf{S}_t$, and the covariance matrix $\boldsymbol{\Sigma}_t$. The gradient of $\mathcal{L}$ with respect to $\mathbf{w}_t$ is $\frac{\partial \mathcal{L}}{\partial \mathbf{w}_t}$. Gradients of $\mathbf{w}_t$ are propagated to the whitening transformation: $\mathbf{w}_t = \mathbf{S}_t^{-\frac{1}{2}}\mathbf{U}_t^{\top}(\mathbf{z}_t - \boldsymbol{\mu}_t)$, which involves: (1) gradients with respect to $\mathbf{U}_t$; (2) gradients with respect to $\mathbf{S}_t^{-\frac{1}{2}}$ (i.e., the square root and inverse of singular values); and (3) gradients with respect to $(\mathbf{z}_t - \boldsymbol{\mu}_t)$. Using PyTorch's `torch.linalg.svd`, we calculate the gradients of $\mathbf{U}_t$ and $\mathbf{S}_t$ via automatic differentiation. For the forward pass, the cost of SVD for $\boldsymbol{\Sigma}(\mathbf{h}_t) \in \mathbb{R}^{B\times B}$ is $O(B^3)$, where B is the matrix dimension. For the backward pass, computing the gradients of $\mathbf{U}_t$ and $\mathbf{S}_t$ also incurs $O(B^3)$ computational cost. Memory usage scales as $O(B^2)$ for storing the covariance matrix and the singular value decomposition outputs $(\mathbf{U}_t, \mathbf{S}_t)$. Additional memory is required for autograd intermediate values, scaling as $O(B^3)$. By leveraging PyTorch's autograd system, we integrate the computation of $\mathbf{U}_t$, $\mathbf{S}_t$, and their gradients seamlessly into our end-to-end learning pipeline. This ensures that the whitening transformation and the loss function are fully differentiable, allowing the model parameters to be trained via gradient-based optimizers. The parameter B also plays a crucial role in the scalability of our method. By leveraging the Gaussian assumption, we are able to train the model using a much smaller subset of time series at each step. Consequently, the size of the covariance matrix is reduced to $B \times B$, as opposed to $N \times N$, where N represents the total number of time series in the dataset. This design ensures that the computational complexity of our method does not scale with N. Moreover, B is kept relatively small in our implementation (e.g., $B = 20$), making the approach computationally efficient.

C.5 Evaluation Metrics

In this paper, we repeated the evaluation procedure on the testing set ten times to compute the mean and standard deviation of each metric. For each evaluation, the metrics were calculated by averaging over all forecast instances in the testing set. For example, the reported $\mathrm{CRPS}_{\text{sum}}$ represents the average $\mathrm{CRPS}_{\text{sum}}$ across all forecast instances. Both CRPS and ES were estimated using Monte Carlo approximation based on 100 sampled predictions.

C.5.1 Continuous Ranked Probability Score

The empirical approximation of the Continuous Ranked Probability Score (CRPS) based on a finite sample $\{x_1, \ldots, x_n\}$ drawn from the predictive distribution F is given by:

$$\mathrm{CRPS}(F, z) = \frac{1}{n} \sum_{i=1}^{n} |x_i - z| - \frac{1}{2n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j| \,, \tag{20}$$

where the first term estimates the expected absolute deviation between the predictive samples and the observation z, while the second term estimates the expected absolute deviation between pairs of predictive samples. This Monte Carlo approximation converges to the true CRPS as $n \to \infty$. An efficient empirical approximation of Eq. (20), based on a sorted sample $\{x_{(1)}, \ldots, x_{(n)}\}$ from the predictive distribution F, is given by:

$$\mathrm{CRPS}(F, z) = \frac{1}{n} \sum_{i=1}^{n} \left| x_{(i)} - z \right| - \frac{1}{n^2} \sum_{i=1}^{n-1} i(n-i) \left(x_{(i+1)} - x_{(i)} \right) , \tag{21}$$

where $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$ are the sorted predictive samples. The first term measures the average absolute error between the sorted samples and the observation z, while the second term provides a linear-time estimate of the expected pairwise absolute differences between samples, avoiding the quadratic cost of a double sum. In this paper, we computed the empirical CRPS using Eq. (21).

For a single forecast instance, we compute $\mathrm{CRPS}_{\text{mean}}$ as the average CRPS across all time series and prediction steps:

$$\mathrm{CRPS}_{\text{mean}} = \mathbb{E}_{i,t} \left[\mathrm{CRPS} \left(F_{i,t}, z_{i,t} \right) \right] , \tag{22}$$

where $F_{i,t}$ denotes the predictive distribution for $z_{i,t}$, represented by its empirical CDF. Since CRPS evaluates one marginal distribution at a time, it does not capture joint dependencies across series. To address this, we also compute $\mathrm{CRPS}_{\text{sum}}$ [4, 22, 23], which aggregates both forecasted and observed values across all time series and applies CRPS to the resulting sums:

$$\mathrm{CRPS}_{\text{sum}} = \mathbb{E}_t \left[\mathrm{CRPS} \left(F_t, \sum_i z_{i,t} \right) \right] , \tag{23}$$

where F_t is the empirical distribution formed by summing prediction samples across all time series.

C.5.2 Energy Score

The Energy Score (ES) generalizes the CRPS to evaluate distributional forecasts of vector-valued random variables, making it a suitable multivariate metric for this paper:

$$\mathrm{ES}(F, \mathbf{z}) = \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \mathbf{z}\|^{\beta} - \frac{1}{2n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^{\beta}, \tag{24}$$

where $\|\cdot\|$ denotes the Euclidean norm, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are samples from the predictive distribution, and $\mathbf{z}$ is the observed vector. In this paper, we set $\beta = 1$, following Ashok et al. [23]. To aggregate over the prediction horizon, we compute the Frobenius norm of the forecast matrix $\|\mathbf{z}_{t+1:t+Q}\|_F$ in practice.

D Additional Results

D.1 Synthetic Data Experiment

We design a controlled noise experiment based on the example shown in §4.2 to evaluate the robustness of different proper scoring rules when estimating parameters of a Gaussian distribution in the presence of contaminated data. The experiment focuses on a two-dimensional multivariate Gaussian distribution $P = \mathcal{N}\left(\begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 & 0.8 \\ 0.8 & 4 \end{bmatrix} \right)$. From this distribution, we generate $N = 5000$ samples as our base dataset. To systematically study robustness properties, we introduce contamination at varying levels $\epsilon \in 0\%, 2\%, 4\%$ by randomly selecting ϵ proportion of individual data points and adding a fixed offset of $+3.0$ to introduce outliers.

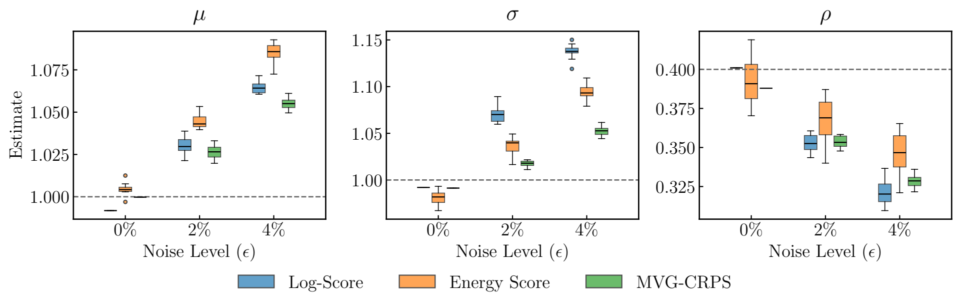

Figure A1: Parameter recovery under data contamination. Boxplots show the estimated parameters (μ, σ, ρ) of a bivariate Gaussian distribution using three proper scoring rules across varying contamination levels. Dashed lines indicate the ground truth values. Each boxplot summarizes estimates from 10 independent runs with different random seeds for contamination.

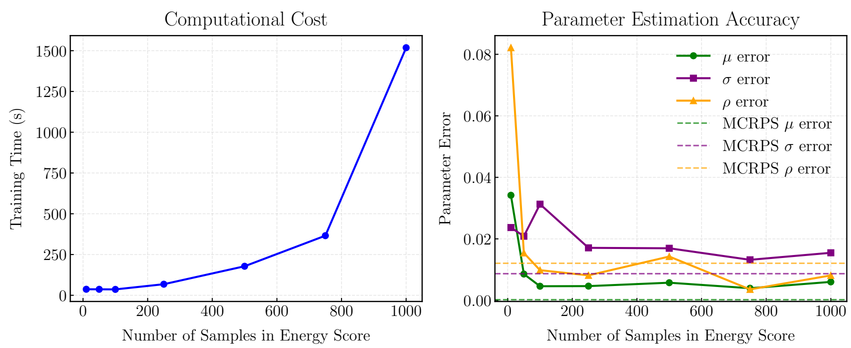

Figure A2: Computational cost versus parameter estimation accuracy for the energy score with varying sample sizes. The left panel shows training time across different numbers of Monte Carlo samples, while the right panel displays absolute errors in parameter estimates (μ, σ, ρ), with dashed lines indicating the corresponding MVG-CRPS reference values.

This experiment compares three proper scoring rules for parameter estimation: the log-score; the energy score, implemented using a Monte Carlo approximation with 500 samples and $\beta = 1.0$; and the proposed MVG-CRPS. For each method and contamination level, we estimate three key parameters of the predictive distribution: μ (location), σ (scale), and ρ (correlation) in

$$Q = \mathcal{N}\left(\begin{bmatrix} \mu \\ -1 \end{bmatrix}, \begin{bmatrix} \sigma^2 & 2\rho\sigma \\ 2\rho\sigma & 4 \end{bmatrix}\right).$$

To ensure that parameter estimates remain within valid ranges, we apply a `softplus` transformation to σ and a `tanh` transformation to ρ, thereby constraining them to appropriate domains.

Optimization is performed using the Adam optimizer with method-specific learning rates: 3×10^{-3} for the log-score and MVG-CRPS, and 1×10^{-2} for the energy score. The number of training iterations also varies: 1000 for the log-score and MVG-CRPS, and 500 for the energy score. These hyperparameters were selected based on preliminary experiments using a validation dataset and a grid search procedure to ensure a fair comparison across methods. To assess statistical significance, we conduct 10 independent runs with different random seeds for each configuration, allowing us to examine the distribution of parameter estimates across trials.

Parameter recovery accuracy is evaluated by comparing the estimated values against the ground truth. We visualize the results using boxplots, which illustrate the distribution of estimates across runs for each method and contamination level (Fig. A1). Across all three parameters, MVG-CRPS consistently yields the most accurate and stable estimates as noise increases. For the location parameter μ and the scale σ, MVG-CRPS maintains estimates closest to the true value with minimal spread, whereas both log-score and energy score drift upward under contamination. For the correlation ρ, noise leads to downward bias for all methods, but MVG-CRPS strikes the best balance between bias and variability. The energy score appears stable under contamination, but this stability follows from its limited sensitivity to changes in correlation, as shown in Fig. 3. Overall, MVG-CRPS shows greater robustness than the log-score and more consistent estimates than the energy score because it does not rely on Monte Carlo sampling.

Using the same example, we conducted a controlled study to examine the trade-off between computational cost and parameter estimation accuracy when using the ES with varying sample sizes. As shown in Fig. A2, training time increases monotonically with sample size due to the pairwise distance computations required by the ES. Estimation errors generally decrease with more samples but exhibit diminishing returns beyond a certain threshold (typically 100–200 samples). For reference, we include MVG-CRPS, which avoids sampling and maintains constant computational cost. Notably, even with large sample sizes (e.g., 1000), the ES does not outperform MVG-CRPS in estimation accuracy.

D.2 Other Metrics for Multivariate Autoregressive Forecasting

The results for $\mathrm{CRPS}_{\text{mean}}$ and ES in the multivariate autoregressive forecasting task are reported in Table A3 and Table A4, respectively. The performance of MVG-CRPS is consistent with the results reported for $\mathrm{CRPS}_{\text{sum}}$ in Table 1.

Table A3: Comparison of $\mathrm{CRPS}_{\text{mean}}$ across different scoring rules in the multivariate autoregressive forecasting task. The best scores are in boldface. MVG-CRPS scores are underlined when they are not the best overall but exceed the log-score.

	VAR	GPVar			Transformer		
		log-score	energy score	MVG-CRPS	log-score	energy score	MVG-CRPS
`elec_au`	N/A	0.1261±0.0009	**0.0887±0.0004**	0.0967±0.0008	0.1633±0.0005	0.1492±0.0006	**0.0793±0.0004**
`cif_2016`	1.0000±0.0000	0.1445±0.0006	0.1690±0.0005	**0.1387±0.0006**	0.1611±0.0010	0.1470±0.0008	**0.1178±0.0003**
`electricity`	0.1598±0.0007	**0.0601±0.0004**	0.0772±0.0003	0.0623±0.0002	**0.0600±0.0002**	0.0705±0.0003	0.0638±0.0002
`elec_weekly`	0.1237±0.0009	0.1427±0.0023	**0.0676±0.0008**	0.0878±0.0026	0.0964±0.0022	0.0726±0.0010	**0.0697±0.0012**
`exchange_rate`	0.0070±0.0000	0.0204±0.0004	0.0094±0.0002	**0.0065±0.0001**	0.0112±0.0002	**0.0102±0.0002**	0.0115±0.0003
`kdd_cup`	N/A	0.3474±0.0008	0.3395±0.0011	**0.2972±0.0010**	0.2959±0.0008	0.4303±0.0022	**0.2282±0.0005**
`m1_yearly`	N/A	0.4397±0.0041	0.4801±0.0022	**0.3566±0.0029**	0.5344±0.0109	**0.3291±0.0047**	0.4563±0.0111
`m3_yearly`	N/A	0.3607±0.0084	0.2186±0.0042	**0.1423±0.0053**	0.3156±0.0102	0.4050±0.0061	**0.2325±0.0094**
`nn5_daily`	0.2446±0.0002	**0.1525±0.0002**	0.1551±0.0002	0.1540±0.0002	0.1500±0.0002	0.1453±0.0001	**0.1410±0.0001**
`saugeenday`	N/A	0.4040±0.0047	**0.3733±0.0048**	0.3941±0.0055	0.3771±0.0088	**0.3689±0.0053**	0.3705±0.0047
`sunspot`	N/A	18.7115±1.3296	23.3988±0.9662	**17.2438±0.5833**	39.7454±1.4841	**16.6556±0.6167**	22.6495±0.6752
`tourism`	0.1444±0.0007	0.2369±0.0027	0.2424±0.0010	**0.2223±0.0017**	0.2290±0.0010	**0.2220±0.0016**	0.2313±0.0017
`traffic`	19.9208±0.0495	**0.1357±0.0002**	0.1367±0.0001	0.1415±0.0001	0.1185±0.0001	0.1327±0.0001	**0.1174±0.0001**
Avg. Rank		2.23	2.23	**1.54**	2.46	1.92	**1.62**

Table A4: Comparison of ES across different scoring rules in the multivariate autoregressive forecasting task. The best scores are in boldface. MVG-CRPS scores are underlined when they are not the best overall but exceed the log-score.

	VAR	GPVar			Transformer		
		log-score	energy score	MVG-CRPS	log-score	energy score	MVG-CRPS
`elec_au` ($\times 10^3$)	N/A	5.4013±0.0372	**3.9136±0.0177**	4.1508±0.0283	7.0039±0.0219	6.3135±0.0243	**3.5217±0.0150**
`cif_2016` ($\times 10^3$)	125.6177±0.0000	4.2733±0.0218	4.9329±0.0161	**4.1677±0.0198**	4.6316±0.0270	4.1063±0.0241	**3.5559±0.0145**
`elec` ($\times 10^4$)	10.4788±0.0757	3.3124±0.0580	4.8317±0.0434	**3.2435±0.0300**	**3.4724±0.0229**	4.3757±0.0414	3.9672±0.0374
`elec_weekly` ($\times 10^7$)	2.2191±0.0308	2.5724±0.0799	**0.8948±0.02344**	1.4040±0.0887	1.5463±0.0582	**0.9338±0.0308**	0.9985±0.0360
`exchange_rate`	0.1301±0.0002	0.3972±0.0074	0.1895±0.0034	**0.1216±0.0013**	0.2136±0.0026	**0.1774±0.0026**	0.2040±0.0045
`kdd_cup` ($\times 10^2$)	N/A	4.7575±0.0186	4.3981±0.0164	**4.0719±0.0180**	4.2809±0.0134	5.9466±0.0427	**3.1788±0.0122**
`m1_yearly` ($\times 10^4$)	N/A	7.3860±0.0789	7.7576±0.0335	**6.1985±0.0505**	8.7079±0.1760	**5.7774±0.0755**	7.5130±0.1784
`m3_yearly` ($\times 10^3$)	N/A	3.6113±0.0703	2.2147±0.0427	**1.4775±0.0495**	3.1996±0.0995	4.0982±0.0621	**2.4253±0.0914**
`nn5_daily` ($\times 10^2$)	4.9419±0.0056	**3.3001±0.0050**	3.3004±0.0052	3.3934±0.0045	3.2546±0.0033	3.1622±0.0045	**3.0996±0.0025**
`saugeenday` ($\times 10^2$)	N/A	1.8098±0.0231	**1.7135±0.0150**	1.9400±0.0208	**1.5780±0.0183**	1.5883±0.0108	1.8043±0.0204
`sunspot` ($\times 10$)	N/A	2.7737±0.1195	3.1658±0.0792	**2.6195±0.1003**	5.4893±0.1132	**2.3153±0.0467**	3.2663±0.0745
`tourism` ($\times 10^5$)	3.5958±0.0354	6.1085±0.1132	5.6774±0.0493	**5.2111±0.0896**	5.0645±0.0526	**4.7502±0.0585**	5.2702±0.0853
`traffic_nips`	3358.5004±10.7535	2.2924±0.0034	**2.1140±0.0023**	2.2916±0.0015	2.2043±0.0012	2.2250±0.0018	**2.2000±0.0018**
Avg. Rank		2.46	2.00	**1.54**	2.38	1.92	**1.69**

D.3 Univariate Seq2Seq Forecasting

The results for the univariate Seq2Seq forecasting task, presented in Table A5, Table A6, and Table A7, are consistent with those from the multivariate autoregressive task. Overall, MVG-CRPS demonstrates improved accuracy compared to both the log-score and the energy score.

Figure A3 visualizes the output covariance matrices from models trained with different loss functions. Similar to the multivariate autoregressive task, the model trained with the log-score exhibits higher variance and covariance values, indicating greater uncertainty that may reduce forecast reliability. The figure illustrates the evolution of daily covariance in the hourly `traffic` dataset, shaped by both the prediction lead time and the time of day. Uncertainty tends to increase during rush hours and at longer forecast horizons. In contrast, the model trained with MVG-CRPS captures these temporal patterns while being less sensitive to extreme values, resulting in more stable estimates.

Figure A4 further compares probabilistic forecasts on the `m4_hourly` dataset. The model trained with MVG-CRPS produces narrower and better-calibrated prediction intervals than the log-score-trained model, particularly for time series with clear cyclical patterns. It also achieves higher accuracy at longer forecast horizons. These results indicate that MVG-CRPS enhances both robustness and calibration, leading to more accurate and reliable forecasts.

Table A5: Comparison of $\mathrm{CRPS}_{\mathrm{sum}}$ across different scoring rules in the univariate Seq2Seq forecasting task. The best scores are in boldface. MVG-CRPS scores are underlined when they are not the best overall but exceed the log-score.

	N-HiTS		
	log-score	energy score	MVG-CRPS
`covid`	0.1297±0.0048	N/A	**0.1011±0.0022**
`elec_hourly`	0.0470±0.0008	N/A	**0.0398±0.0004**
`electricity`	0.0409±0.0003	0.0378±0.0006	**0.0372±0.0003**
`exchange_rate`	0.0089±0.0005	0.0060±0.0002	**0.0053±0.0002**
`m4_hourly`	0.0649±0.0007	0.0595±0.0005	**0.0399±0.0007**
`nn5_daily`	0.0571±0.0003	0.0876±0.0006	**0.0569±0.0004**
`pedestrian`	0.7985±0.0511	0.9110±0.0210	**0.5296±0.0071**
`saugeenday`	0.4804±0.0150	0.4372±0.0100	**0.3864±0.0035**
`taxi_30min`	0.0496±0.0002	0.0603±0.0002	**0.0449±0.0001**
`traffic`	0.2065±0.0007	**0.0815±0.0001**	0.0832±0.0002
`uber_hourly`	0.7027±0.0209	0.6461±0.0052	**0.5380±0.0033**
`wiki`	0.0660±0.0011	**0.0429±0.0003**	0.0465±0.0004
Avg. Rank	2.70	2.10	**1.20**

Table A6: Comparison of $\mathrm{CRPS}_{\mathrm{mean}}$ across different scoring rules in the univariate Seq2Seq forecasting task. The best scores are in boldface. MVG-CRPS scores are underlined when they are not the best overall but exceed the log-score.

	N-HiTS		
	log-score	energy score	MVG-CRPS
`covid`	0.2076±0.0018	0.1440±0.0013	**0.1022±0.0012**
`elec_hourly`	0.0903±0.0005	0.1189±0.0004	**0.0874±0.0003**
`electricity`	0.0671±0.0002	0.0913±0.0002	**0.0635±0.0001**
`exchange_rate`	0.0173±0.0004	0.0077±0.0001	**0.0073±0.0001**
`m4_hourly`	0.1599±0.0003	0.1762±0.0007	**0.1093±0.0005**
`nn5_daily`	0.1964±0.0006	**0.1588±0.0002**	0.1846±0.0008
`pedestrian`	1.0856±0.0262	0.9254±0.0105	**0.7328±0.0076**
`saugeenday`	0.4804±0.0150	0.4372±0.0100	**0.3864±0.0035**
`taxi_30min`	0.3853±0.0001	0.3939±0.0001	**0.3219±0.0000**
`traffic`	0.2514±0.0004	0.1726±0.0001	**0.1583±0.0001**
`uber_hourly`	0.9630±0.0272	0.8229±0.0062	**0.6852±0.0040**
`wiki`	0.4160±0.0006	0.2824±0.0003	**0.2656±0.0002**
Avg. Rank	2.67	2.25	**1.08**

Table A7: Comparison of ES across different scoring rules in the univariate Seq2Seq forecasting task. The best scores are in boldface. MVG-CRPS scores are underlined when they are not the best overall but exceed the log-score.

	N-HiTS		
	log-score	energy score	MVG-CRPS
`covid` ($\times 10^5$)	2.1220±0.0304	N/A	**0.9401±0.0186**
`elec_hourly` ($\times 10^5$)	0.9283±0.0161	N/A	**0.9088±0.0079**
`elec` ($\times 10^5$)	0.2535±0.0018	0.3123±0.0019	**0.2431±0.0020**
`exchange_rate`	0.2876±0.0055	0.1272±0.0022	**0.1240±0.0022**
`m4_hourly` ($\times 10^4$)	0.2852±0.0026	0.2890±0.0029	**0.2423±0.0027**
`nn5_daily` ($\times 10^3$)	0.4170±0.0018	**0.3272±0.0005**	<u>0.3958±0.0021</u>
`pedestrian` ($\times 10^3$)	1.1571±0.0177	0.9746±0.0081	**0.8337±0.0066**
`saugeenday` ($\times 10^2$)	**1.6690±0.0391**	1.7752±0.0216	1.7698±0.0129
`taxi_30min` ($\times 10^2$)	6.9676±0.0045	6.7906±0.0058	**5.6679±0.0004**
`traffic`	3.6810±0.0136	2.2524±0.0018	**2.2200±0.0022**
`uber_hourly`	6.3252±0.1785	5.4214±0.0326	**4.2826±0.0320**
`wiki` ($\times 10^6$)	1.1535±0.0047	0.9352±0.0069	**0.9338±0.0083**
Avg. Rank	2.60	2.20	**1.20**

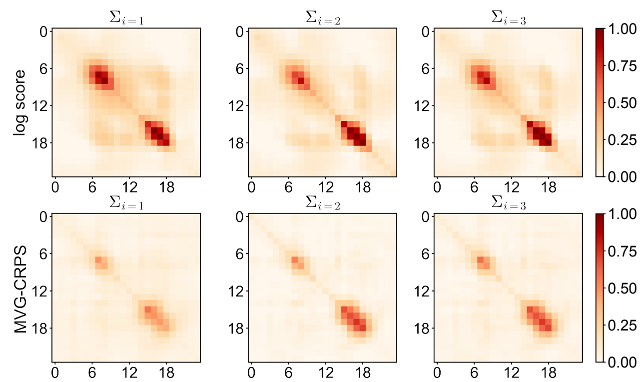

Figure A3: Comparison of output covariance matrices $\mathbf{\Sigma}_i$ from N-HiTS on the `traffic` dataset. For visual clarity, covariance values are clipped between 0 and 1.0.

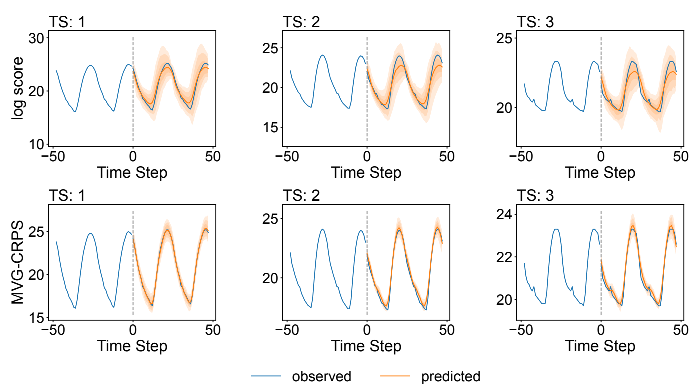

Figure A4: Comparison of probabilistic forecasts from N-HiTS on the `m4_hourly` dataset.

D.4 Hyperparameter Sensitivity

To ensure a fair comparison, our main experiments used fixed hyperparameters across all loss functions. However, since certain hyperparameters such as learning rate and rank do not affect the model architecture, we performed grid searches over learning rates $10^{-2}, 10^{-3}, 10^{-4}$ and rank values $10, 20, 30$ for each dataset. The optimal configuration was selected based on validation performance for each combination of loss function, model group, and dataset. The results are presented in Table A8 and Table A9. With tuned hyperparameters, the MVG-CRPS still achieves the best average rank.

Table A8: Comparison of $\mathrm{CRPS}_{\text{mean}}$ across different scoring rules in the multivariate autoregressive forecasting task. The best scores are in boldface. MVG-CRPS scores are underlined when they are not the best overall but exceed the log-score. The results are obtained using models with the best hyperparameters (learning rate and rank), selected for each loss function, model group, and dataset based on validation performance. For the energy score, hyperparameter tuning was omitted due to extended training time.

	VAR	GPVar			Transformer		
		log-score	energy score	MVG-CRPS	log-score	energy score	MVG-CRPS
`elec_au`	N/A	0.0437±0.0004	0.0887±0.0004	**0.0280±0.0002**	**0.1158±0.0005**	0.1492±0.0006	0.1410±0.0004
`cif_2016`	1.0000±0.0000	0.1444±0.0004	0.1690±0.0005	**0.1275±0.0003**	0.1217±0.0005	0.1470±0.0008	**0.1201±0.0002**
`electricity`	0.1598±0.0007	**0.0601±0.0004**	0.0772±0.0003	0.0665±0.0004	**0.0605±0.0003**	0.0705±0.0003	0.0650±0.0002
`elec_weekly`	0.1237±0.0009	0.1128±0.0014	**0.0676±0.0008**	0.1046±0.0025	0.1000±0.0020	**0.0726±0.0010**	0.1061±0.0013
`exchange_rate`	0.0070±0.0000	**0.0071±0.0001**	0.0094±0.0002	0.0093±0.0002	0.0131±0.0003	**0.0102±0.0002**	0.0161±0.0002
`kdd_cup`	N/A	0.3274±0.0015	0.3395±0.0011	**0.2861±0.0004**	0.2865±0.0012	0.4303±0.0022	**0.2291±0.0010**
`m1_yearly`	N/A	0.4883±0.0088	0.4801±0.0022	**0.3333±0.0015**	0.5394±0.0111	**0.3291±0.0047**	0.4420±0.0070
`m3_yearly`	N/A	0.3606±0.0133	0.2186±0.0042	**0.1423±0.0053**	0.3658±0.0097	0.4050±0.0061	**0.2964±0.0136**
`nn5_daily`	0.2446±0.0002	**0.1474±0.0002**	0.1551±0.0002	0.1510±0.0001	0.1466±0.0001	0.1453±0.0001	**0.1430±0.0001**
`saugeenday`	N/A	0.3715±0.0032	0.3733±0.0048	**0.3600±0.0053**	0.3756±0.0055	**0.3689±0.0053**	0.3831±0.0032
`sunspot`	N/A	**10.7124±0.4618**	23.3988±0.9662	16.1930±0.5734	14.4194±0.5650	16.6556±0.6167	**13.1737±0.6602**
`tourism`	0.1444±0.0007	0.2492±0.0015	0.2424±0.0010	**0.1193±0.0020**	0.2258±0.0020	0.2220±0.0016	**0.2082±0.0014**
`traffic`	19.9208±0.0495	0.1534±0.0002	**0.1367±0.0001**	0.1415±0.0001	0.1422±0.0001	0.1327±0.0001	**0.1152±0.0000**
Avg. Rank		2.08	2.46	**1.46**	2.15	2.08	**1.77**

Table A9: Comparison of ES across different scoring rules in the multivariate autoregressive forecasting task. The best scores are in boldface. MVG-CRPS scores are underlined when they are not the best overall but exceed the log-score. The results are obtained using models with the best hyperparameters (learning rate and rank), selected for each loss function, model group, and dataset based on validation performance. For the energy score, hyperparameter tuning was omitted due to extended training time.

	VAR	GPVar			Transformer		
		log-score	energy score	MVG-CRPS	log-score	energy score	MVG-CRPS
`elec_au` ($\times 10^3$)	N/A	1.9601±0.0200	3.9136±0.0177	**1.2546±0.0066**	**4.9064±0.0215**	6.3135±0.0243	5.9514±0.0150
`cif_2016` ($\times 10^3$)	125.6177±0.0000	4.3478±0.0127	4.9329±0.0161	**3.8815±0.0072**	3.6976±0.0203	4.6316±0.0270	**3.5888±0.0118**
`elec` ($\times 10^4$)	10.4788±0.0757	**3.6854±0.0973**	4.8317±0.0434	3.6913±0.0353	4.9963±0.0371	**3.4724±0.0229**	4.6774±0.0544
`elec_weekly` ($\times 10^7$)	2.2191±0.0308	1.9808±0.0774	**0.8948±0.0234**	1.2270±0.0539	1.5074±0.0600	1.5463±0.0582	**1.0231±0.0402**
`exchange_rate`	0.1301±0.0002	0.2166±0.0061	0.1895±0.0034	**0.1519±0.0018**	0.2317±0.0042	0.2136±0.0026	**0.1569±0.0032**
`kdd_cup` ($\times 10^2$)	N/A	4.7575±0.0186	**4.3981±0.0164**	5.0382±0.0142	5.2922±0.0202	4.2809±0.0134	**3.2651±0.0134**
`m1_yearly` ($\times 10^4$)	N/A	8.1941±0.1388	7.7576±0.0335	**5.9567±0.0210**	8.7995±0.1777	8.7079±0.1777	**7.2322±0.0979**
`m3_yearly` ($\times 10^3$)	N/A	3.6966±0.1408	2.2147±0.0427	**1.4775±0.0495**	3.7233±0.0925	3.1996±0.0995	**2.9483±0.1275**
`nn5_daily` ($\times 10^2$)	4.9419±0.0056	**3.1966±0.0044**	3.3004±0.0052	3.3303±0.0031	**3.1311±0.0038**	3.2546±0.0033	3.1725±0.0033
`saugeenday` ($\times 10^2$)	N/A	**1.6529±0.0150**	1.7135±0.0150	1.7678±0.0188	1.6434±0.0160	**1.5780±0.0183**	1.6426±0.0220
`sunspot` ($\times 10$)	N/A	**1.7726±0.0430**	3.1658±0.0792	2.5742±0.0651	2.1717±0.0468	5.4893±0.1132	**1.9724±0.0363**
`tourism` ($\times 10^5$)	3.5958±0.0354	6.5310±0.0670	5.6774±0.0493	**2.8103±0.1048**	6.0582±0.1162	5.0645±0.0526	**4.5365±0.0662**
`traffic_nips`	3358.5004±10.7535	2.4690±0.0026	**2.1140±0.0023**	2.2967±0.0012	2.2314±0.0016	2.2043±0.0012	**2.1626±0.0020**
Avg. Rank		2.15	2.08	**1.77**	2.46	2.23	**1.31**

D.5 Controlled Outlier Experiment

We conducted an additional experiment by injecting synthetic outliers into the training data. Specifically, a fixed proportion of observations for each sensor was perturbed with large noise ($\pm 5\times$ the sensor's standard deviation). The test data remained clean to isolate the impact of training-time contamination. Results in Fig. A5 indicate that models trained with the log-score degrade rapidly under such noise, whereas the MVG-CRPS demonstrates greater robustness.

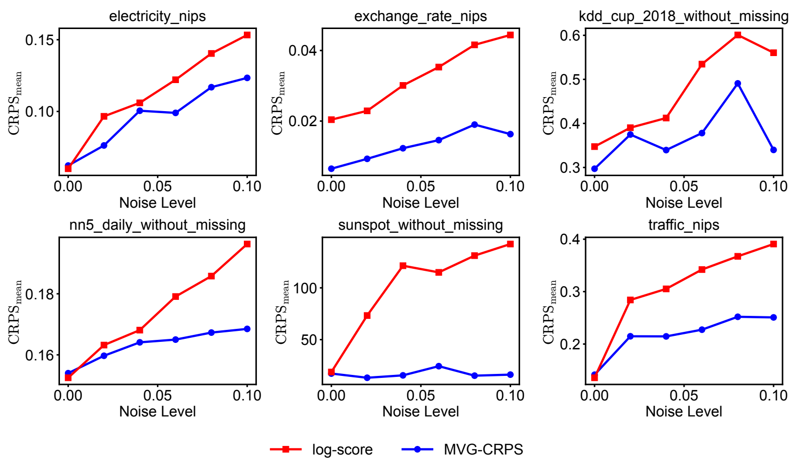

Figure A5: Controlled outlier experiment using GPVar. A fixed proportion of training samples per sensor is perturbed by adding large noise.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and/or introduction have clearly stated the claims made, including the contributions made in the paper and important assumptions and limitations.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations of the work in the "Conclusion" section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We have provided a proof showing that the proposed MVG-CRPS is strictly proper under Gaussian assumption.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have fully disclosed all the information needed to reproduce the main experimental results of the paper in the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The code will be released after the paper is accepted. However, we have provided a sufficient amount of experimental details in the Appendix.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specified all the training and test details necessary to understand the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We ran all of our experiments for 10 times to calculate the standard deviation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper has indicated the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed societal impacts in the last section of this paper.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets (e.g., code, data, models), used in the paper, have been properly credited. The license and terms of use have been explicitly mentioned and properly respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.