Probabilistic Forecasting with Coherent Aggregation

Anonymous authors
Paper under double-blind review

Abstract

Obtaining accurate probabilistic forecasts is an important operational challenge in many applications, like energy management, climate forecast, supply chain planning, and resource allocation. In many of these applications, there is a natural hierarchical structure over the forecasted quantities; and forecasting systems that adhere to this hierarchical structure are said to be coherent. Furthermore, operational planning benefits from accuracy at all levels of the aggregation hierarchy. Building accurate and coherent forecasting systems, however, is challenging: classic multivariate time series tools and neural network methods are still being adapted for this purpose. In this paper, we augment an MQForecaster neural network architecture with a novel deep Gaussian factor forecasting model that achieves coherence by construction, yielding a method we call the *Deep Coherent Factor Model Neural Network* (`DeepCoFactor`) model. `DeepCoFactor` generates samples that can be differentiated with respect to the model parameters, allowing optimization on various sample-based learning objectives that align with the forecasting system's goals, including quantile loss and the scaled Continuous Ranked Probability Score (CRPS). In a comparison to state-of-the-art coherent forecasting methods, `DeepCoFactor` achieves significant improvements in scaled CRPS forecast accuracy, with average gains of 15%, as measured on six publicly-available forecasting datasets.

1 Introduction

Obtaining accurate forecasts is an important step for long-term planning in complex and uncertain environments, with applications ranging from energy to supply chain management, from transportation to climate prediction (Hong et al., 2014; Gneiting & Katzfuss, 2014; Makridakis et al., 2022). Going beyond point forecasts such as means and medians, probabilistic forecasting provides a key tool for forecasting uncertain future events. This involves, e.g., forecasting that there is a 90% chance of rain on a certain day, or that there is a 99% chance that people will want to buy fewer than 100 items at a certain store on a certain week. Providing more detailed predictions of this form permits finer uncertainty quantification. This in turn permits planners to prepare for different scenarios and to allocate resources depending on their anticipated likelihood and cost structure. This can lead to better resource allocation, improved decision making, and less waste.

In many forecasting applications, there exist natural hierarchies over the quantities one wants to predict, such as energy consumption at various temporal granularities (from monthly to weekly), different geographic levels (from building-level to city-level to state-level), or retail demand for specific items (in a hierarchical product taxonomy). Typically, most or all levels of the hierarchy are important: the bottom levels are key for operational short-term planning, while higher levels of aggregation provide insights into longer-term or broader trends. Moreover, it is often desired that probabilistic forecasts are coherent (or consistent) to ensure efficient decision-making at all levels (Hong et al., 2014; Jeon et al., 2019). Coherence is achieved when the forecast distribution assigns zero probability to forecasts that do not satisfy the hierarchy's constraints (Panagiotelis et al., 2023; Ben Taieb et al., 2017; Olivares et al., 2023) (see Definition 2.1). Designing an accurate model, capable of leveraging information from all hierarchical levels, while enforcing coherence is a well-known and challenging task (Hyndman et al., 2011).

The hierarchical forecasting literature has been dominated by two-stage reconciliation approaches, where univariate methods are first fitted and later reconciled towards coherence. For many years, most research

Method	End-to-End	Multivariate Series Inputs	Multivariate Output Distribution	Arbitrary Learning Objective
`PERMBU` (Ben Taieb et al., 2017)	✗	✗	✗	✗
`Bootstrap` (Panagiotelis et al., 2023)	✗	✗	✗	✗
`Normality` (Wickramasuriya, 2023)	✗	✗	✓	✗
`DPMN` (Olivares et al., 2023)	✓	✗	✓	✗
`HierE2E` (Rangapuram et al., 2021)	✓	✓	✗	✓
`DeepCoFactor` (ours)	✓	✓	✓	✓

Table 1: Coherent forecast methods' desirable properties.

focused on mean reconciliation (Hyndman et al., 2011; 2024; Vitullo, 2011; Hyndman et al., 2016; Dangerfield & Morris, 1992; Wickramasuriya et al., 2019; Mishchenko et al., 2019). More recent statistical methods consider coherent probabilistic forecasts through variants of the bootstrap reconciliation technique (Ben Taieb et al., 2017; Panagiotelis et al., 2023) or the clever use of the Gaussian forecast distributions' properties (Wickramasuriya, 2023). Large-scale applications of hierarchical forecasting require one to simplify over the two-stage reconciliation process by favoring *end-to-end* approaches that simultaneously fit all levels of the hierarchy, while still achieving coherence. The end-to-end approach refers to training a model constrained to achieve coherence by optimizing directly for accuracy. End-to-end methods offer advantages such as reduced complexity, improved computational efficiency, and enhanced adaptability by streamlining the entire forecasting pipeline into a single, unified model. More importantly, end-to-end models generally achieve better accuracy compared to two-stage models that are first trained independently for optimized accuracy and then made coherent through various reconciliation approaches (Rangapuram et al., 2021; Olivares et al., 2023).

To the best of our knowledge, only three methods yield coherent probabilistic forecasts and allow models to be trained in an end-to-end manner: Rangapuram et al. (2021), Olivares et al. (2023), and Das et al. (2023). (There is also parallel research on hierarchical forecasts with relaxed constraints (Han et al., 2021; Paria et al., 2021; Kamarthi et al., 2022; 2024; Umagami et al., 2023), but we limit our analysis of this line of work since our focus is on strictly coherent forecasting methods.) In particular, Olivares et al. (2023) considers a finite mixture of Poisson distributions that captures correlations implicitly through latent variables and does not leverage cross-time series information, and coherence is achieved by constructing probabilistic forecasts for aggregated time series through equality completion (Donti et al., 2021). On the other hand, Rangapuram et al. (2021) leverages the multivariate time series information, but it achieves coherence through a differentiable projection layer, which could degrade forecast accuracy: it does not directly model correlations between the multivariate outputs, but rather it couples them through its projection layer. Dedicated effort is still necessary to capture these hierarchical relationships to improve forecast accuracy. Such probabilistic methods can benefit from the ability to optimize for arbitrary loss functions through making samples differentiable, as demonstrated by Rangapuram et al. (2021). The capacity to optimize any loss computed from forecast samples can help align the forecasting system's goals with the neural network's learning objective.

Summarized, an ideal hierarchical forecasting method should satisfy several desiderata: 1) be end-to-end coherent; 2) model a joint multivariate probability distribution, capturing the intricate relationships between series within the hierarchy; 3) leverage cross time series information to accurately reflect these relationships; and 4) generate differentiable samples, to enable the method to optimize for arbitrary learning objectives that align with the forecasting system's goals. In this paper, we present a method which satisfies all of these ideal properties; see Table 1 for a summary.

We introduce the *Deep Coherent Factor Model Neural Network* (`DeepCoFactor`), a method for producing probabilistic coherent forecasting that satisfies all the desired properties stated above. Our **main contributions** are the following.

1. We design a multivariate Gaussian factor model that achieves forecast coherence exactly by construction, representing the joint forecast distribution over the bottom-level series. This factor model is generic and can be applied to most neural forecasting models with minimal modifications.

2. In contrast to other joint distribution models, our factor model samples' differentiability—enabled by the reparameterization trick—offers greater versatility, allowing to optimize a wider range of learning objectives beyond negative log likelihood. We leverage this versatility by training the model using both the Continuous Ranked Probability Score (CRPS) and the Energy Score (Matheson & Winkler, 1976). For hierarchical forecasting tasks, aligning the training objective with the CRPS evaluation metric leads to significant improvements in accuracy.

3. We combine the multivariate factor model with an augmented `MQCNN` architecture (Wen et al., 2017; Olivares et al., 2022c) into the *Deep Coherent Factor Model Neural Network* (`DeepCoFactor`). Our augmented `MQCNN` architecture accepts Vector Autorregresive (VAR) inputs through a cross series multi layer perceptron module.

4. `DeepCoFactor` achieves state-of-the-art performance across six publicly available benchmark datasets, demonstrating a substantial improvement in both probabilistic and mean forecast accuracy. On average, `DeepCoFactor` outperforms the second-best method by 15 percent in probabilistic forecasting accuracy and delivers a 30 percent increase in mean forecast accuracy.

2 Hierarchical Forecast Task.

Notation. We introduce the hierarchical forecast task following Olivares et al. (2023). We denote a hierarchical multivariate time series vector by $\mathbf{y}_{[i],t} = \left[\mathbf{y}_{[a],t}^{\top} \mid \mathbf{y}_{[b],t}^{\top}\right] \in \mathbb{R}^{N_a+N_b}$, where $[i] = [a] \cup [b]$, $[a]$, and $[b]$ denote the set of full, aggregate and bottom indices of the time series, respectively. There are $|[i]| = N_a + N_b$ time series in total, with $|[a]| = N_a$ aggregates from the $|[b]| = N_b$ bottom time series, at the finest level of granularity. We use t as a time index. In our notations, we keep track of shape of tensors using square brackets in subscripts. Since each aggregated time series is a linear transformation of the multivariate bottom series, we write the hierarchical aggregation constraint as

$$\mathbf{y}_{[i],t} = \mathbf{S}_{[i][b]}\mathbf{y}_{[b],t} \iff \begin{bmatrix}\mathbf{y}_{[a],t}\\ \mathbf{y}_{[b],t}\end{bmatrix} = \begin{bmatrix}\mathbf{A}_{[a][b]}\\ \mathbf{I}_{[b][b]}\end{bmatrix}\mathbf{y}_{[b],t}. \tag{1}$$

The aggregation matrix $\mathbf{A}_{[a][b]}$ represents the collection of linear transformations for deriving the aggregates, and sums the bottom series to the aggregate levels. The hierarchical aggregation constraints matrix $\mathbf{S}_{[i][b]}$ obtained by stacking $\mathbf{A}_{[a][b]}$ and the $N_b \times N_b$ identity matrix $\mathbf{I}_{[b][b]}$.

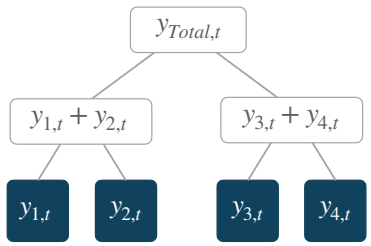
For a simple example, consider $N_b = 4$ bottom-series, so $[b] = \{1, 2, 3, 4\}$ and $y_{Total,t} = \sum_{i=1}^{4} y_{i,t}$. Figure 1 shows an example of such hierarchical structure, where the multivariate hierarchical time series is defined

$$\mathbf{y}_{[a],t} = [y_{Total,t},\ y_{1,t} + y_{2,t}\ y_{3,t} + y_{4,t}]^{\top}, \qquad \mathbf{y}_{[b],t} = [y_{1,t},\ y_{2,t},\ y_{3,t},\ y_{4,t}]^{\top}. \tag{2}$$

Probabilistic Forecast Task. Consider historical temporal features $\mathbf{x}^{(h)}_{[b][:t]}$, known future information $\mathbf{x}^{(f)}_{[b][t+1:t+N_h]}$, and static data $\mathbf{x}^{(s)}_{[b]}$, forecast creation date t and forecast horizons in $[t+1 : t+N_h]$. A classic forecasting task aims to estimate the following conditional probability [1]:

$$\mathbb{P}_{[i]}\left(\mathbf{Y}_{[i],t+\eta} \mid \mathbf{x}^{(h)}_{[b][:t]},\ \mathbf{x}^{(f)}_{[b][t+1:t+N_h]},\ \mathbf{x}^{(s)}_{[b]}\right) \qquad \text{for } \eta = 1, \cdots, N_h. \tag{3}$$

[1] The conditional independence assumption across t in Eqn. 3, maintains the computational tractability of the forecasting task, and is an assumption used by most forecasting methods.



(a) Graph representation

$$\mathbf{S}_{[i][b]} = \begin{bmatrix} \mathbf{A}_{[a][b]} \\ \\ \mathbf{I}_{[b][b]} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(b) Matrix representation

Figure 1: A simple time series hierarchical structure with $N_a = 3$ aggregates over $N_b = 4$ bottom time series. Figure 1a shows the disaggregated bottom variables with blue background. Figure 1b (right) shows the corresponding hierarchical aggregation constraints matrix with horizontal lines to separate levels of the hierarchy. We decompose our evaluation throughout the levels.

The hierarchical forecasting task augments the forecast probability in Eqn. 3 with coherence constraints in Eqn. 2.1 (Ben Taieb et al., 2020; Panagiotelis et al., 2023; Olivares et al., 2022c), by restricting the probabilistic forecast space to assign zero probability to non-coherent forecasts. Definition 2.1 formalizes the intuition, and states that the distribution of a given aggregate random variable is exactly the distribution defined as the aggregates of the bottom-series distributions through the summation matrix $\mathbf{S}_{[i][b]}$.

Definition 2.1. Let $(\Omega_{[b]}, \mathcal{F}_{[b]}, \mathbb{P}_{[b]})$ be a probabilistic forecast space (on the bottom-level series $\mathbf{Y}_{[b]}$), with sample space $(\Omega_{[b]}, \mathcal{F}_{[b]})$, and $\mathbb{P}_{[b]}$ a forecast probability. Let $\mathbf{S}_{[i][b]}(\cdot) : \Omega_{[b]} \mapsto \Omega_{[i]}$ be the linear transformation implied by the aggregation constraints matrix. A **coherent forecast** space $(\Omega_{[i]}, \mathcal{F}_{[i]}, \mathbb{P}_{[i]})$ satisfies

$$\mathbb{P}_{[i]}\left(\mathbf{S}_{[i][b]}(\mathcal{B})\right) = \mathbb{P}_{[b]}(\mathcal{B}), \tag{4}$$

for any set $\mathcal{B} \in \mathcal{F}_{[b]}$ and set's image $\mathbf{S}_{[i][b]}(\mathcal{B}) \in \mathcal{F}_{[i]}$.

Hierarchical Forecast Scoring Rule. In this work and most hierarchical forecasting literature, the performance of probabilistic forecasts is primarily evaluated by the Continuous Ranked Probability Score (CRPS), e.g. Ben Taieb et al. (2017); Rangapuram et al. (2021); Olivares et al. (2023); Panagiotelis et al. (2023); Das et al. (2023); Wickramasuriya (2023). The CRPS between a target y and distributional forecast Y is defined as

$$\mathrm{CRPS}\left(y, Y\right) = \mathbb{E}_{Y}\left[\,|Y - y|\,\right] - \frac{1}{2}\mathbb{E}_{Y,Y'}\left[\,|Y - Y'|\,\right], \tag{5}$$

where Y' is distributed as Y, but is independent of it (Matheson & Winkler, 1976; Gneiting & Raftery, 2007).

CRPS is commonly used because it is a strictly proper scoring rule (Gneiting & Raftery, 2007), and is agnostic of model and distributional assumptions. The metric also gives a summary of peformance on all quantile forecasts (Laio & Tamea, 2007), as

$$\mathrm{CRPS}(y, Y) = 2\int \mathrm{QL}_{q}(y, F_{Y}^{-1}(q))dq, \tag{6}$$

where $F_{Y}^{-1}(q)$ is the q-quantile of variable Y. The q-quantile loss is defined as

$$\mathrm{QL}_{q}(y, F_{Y}^{-1}(q)) = q(y - F_{Y}^{-1}(q))_{+} + (1-q)(F_{Y}^{-1}(q) - y)_{+}. \tag{7}$$

3 Methodology

In this section, we describe our main method, the *Deep Coherent Factor Model Neural Network* (`DeepCoFactor`) model. It consists of a multivariate probabilistic model, an underlying neural network structure, and an end-to-end model estimation procedure.

3.1 Multivariate Probabilistic Model

Our predicted probabilistic forecasts at all hierarchical levels are jointly represented by a Gaussian factor model. Our neural network maps the known information (past, static and known future) to the location, scale and shared factor parameters, and the forecasted factor model parameters are designed to model correlations between the bottom-level series, while conditioning on all known information. Our factor model[2] combined with the coherent aggregation in Eqn. 10 directly estimates the multivariate probability of bottom-level series $\mathbf{y}_{[b][t+1:t+N_h]}$ conditioning on historical, known-future, and static covariates $\mathbf{x}^{(h)}_{[b][:t]}$, $\mathbf{x}^{(f)}_{[b][t+1:t+N_h]}, \mathbf{x}^{(s)}_{[b]}$, i.e.,

$$\mathbb{P}\left(\tilde{\mathbf{Y}}_{[i][t+1:t+N_h]} \mid \mathbf{x}^{(h)}_{[b][:t]},\ \mathbf{x}^{(f)}_{[b][t+1:t+N_h]}, \mathbf{x}^{(s)}_{[b]}\right) = \mathbb{P}\left(\mathbf{S}_{[i][b]}\hat{\mathbf{Y}}_{[b][t+1:t+N_h]} \mid \hat{\boldsymbol{\mu}}_{[b][h],t},\ \hat{\boldsymbol{\sigma}}_{[b][h],t}, \hat{\mathbf{F}}_{[b][k][h],t}\right). \tag{8}$$

At a given forecast creation date t, the model uses the location $\hat{\boldsymbol{\mu}}_{[b][h],t} \in \mathbb{R}^{N_b \times N_h}$, scale $\hat{\boldsymbol{\sigma}}_{[b][h],t} \in \mathbb{R}^{N_b \times N_h}$ and shared factor $\hat{\mathbf{F}}_{[b][k][h],t} \in \mathbb{R}^{N_b \times N_k \times N_h}$ parameters, along with samples from standard normal variables $\mathbf{z}_{[b],\eta} \sim \mathcal{N}(\mathbf{0}_{[b]}, \mathbf{I}_{[b][b]})$, and $\boldsymbol{\epsilon}_{[k],\eta} \sim \mathcal{N}(\mathbf{0}_{[k]}, \mathbf{I}_{[k][k]})$ to compose the following multivariate variables for each horizon:

$$\hat{\mathbf{y}}_{[b],\eta,t} = \hat{\boldsymbol{\mu}}_{[b],\eta,t} + \mathrm{Diag}(\hat{\boldsymbol{\sigma}}_{[b],\eta,t})\mathbf{z}_{[b],\eta,t} + \hat{\mathbf{F}}_{[b][k],\eta,t}\boldsymbol{\epsilon}_{[k],\eta,t}, \qquad \eta = 1, \cdots, N_h. \tag{9}$$

After sampling from the multivariate factor we *coherently aggregate* the clipped outputs of the network,

$$\tilde{\mathbf{y}}_{[i],\eta,\tau} = \mathbf{S}_{[i][b]}\left(\hat{\mathbf{y}}_{[b],\eta,\tau}\right)_{+}. \tag{10}$$

The shared factors enable the factor model to capture the relationships across the disaggregated series, and the covariance structure of the disaggregated series follows:

$$\mathrm{Cov}\left(\hat{\mathbf{y}}_{[b],\eta,t}\right) = \mathrm{Diag}(\hat{\boldsymbol{\sigma}}^2_{[b],\eta,t}) + \hat{\mathbf{F}}_{[b][k],\eta,t}\hat{\mathbf{F}}^{\top}_{[b][k],\eta,t}. \tag{11}$$

Definition 3.1. Any bottom-level multivariate distribution can be transformed into a coherent distribution through **coherent aggregation** [3].Given a sample $\hat{\mathbf{y}}_{[b]} \sim \mathbb{P}_{[b]}$, a coherent $\mathbb{P}_{[i]}$ distribution can be constructed with the following sample transformation

$$\tilde{\mathbf{y}}_{[i]} = \mathbf{S}_{[i][b]}\left(\hat{\mathbf{y}}_{[b]}\right). \tag{12}$$

In other words, it is enough to aggregate the bottom-level forecasts in a bottom-up manner. We include in Appendix A a proof of the approach's coherence property, and the covariance structure.

3.2 Neural Network Architecture

Our framework can adapt to any univariate deep learning forecasting models, so long as they can be made to output factor model parameters. In this paper, we focus on the architecture implementation based on `MQCNN` (Wen et al., 2017; Olivares et al., 2022c), because of its outstanding performance in multi-step forecasting problems. Our `MQCNN`-based architecture has a main encoder which consists of a stack of dilated temporal convolutions, and it is applied to historical information for all series. In addition, it uses a global multi-layer perceptron (MLP) to encode the static and future information. The encoder at time t is described in Eqn. 13 below, and is applied to each disaggregated series:

$$\begin{aligned}
\mathbf{h}^{(h)}_{[i],t} &= \mathrm{TempConvolution}\left([\mathbf{S}_{[i][b]}\mathbf{x}^{(h)}_{[b][:t]}]\right)\ ^{(4)} \\
\mathbf{h}^{(s)}_{[b]} &= \mathrm{MLP}_1\left(\mathbf{x}^{(s)}_{[b]}\right) \\
\mathbf{h}^{(f)}_{[b],t} &= \mathrm{MLP}_2\left(\mathbf{x}^{(f)}_{[b][t+1:t+N_h]}\right).
\end{aligned} \tag{13}$$

[2]Early work on factor forecast models augmenting neural networks done by Wang et al. (2019) does not ensure coherence.

[3]Coherent aggregation can be though of a special case of bootstrap reconciliation (Panagiotelis et al., 2023) that only relies on a bottom-level forecast distribution.

[4]Temporal exogenous data only aggregates the target signal, other features (e.g. calendar) are maintained without aggregation.

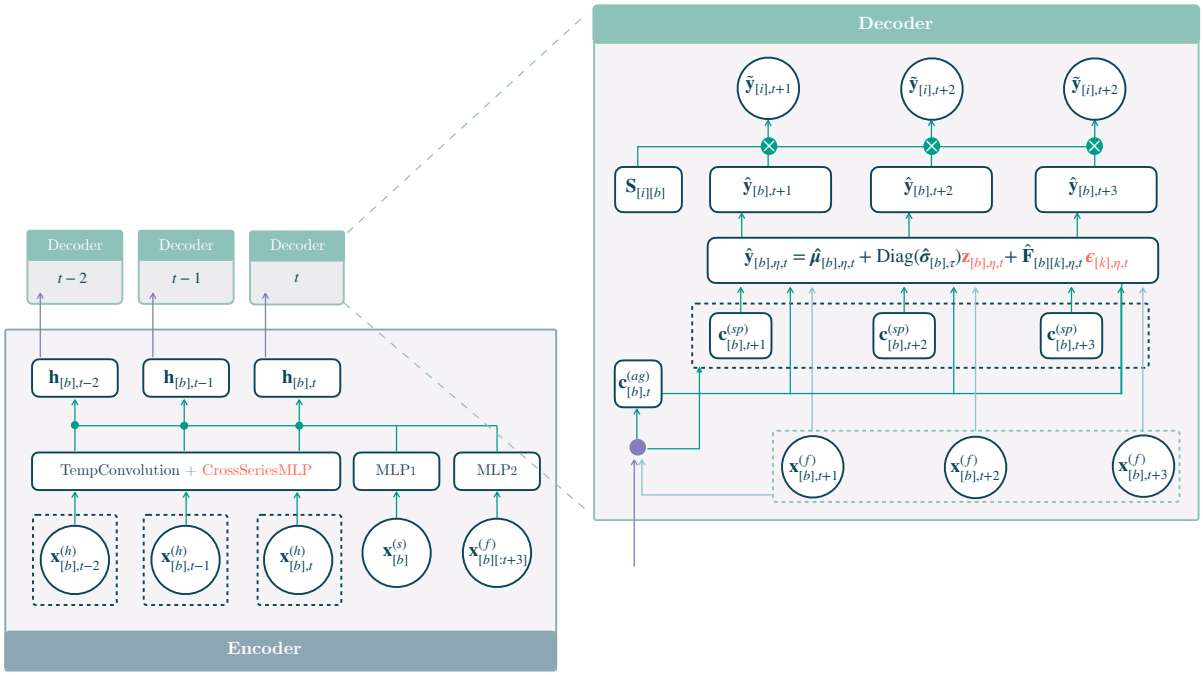


Figure 2: The *Deep Coherent Factor Model Neural Network* is a Sequence-to-Sequence with Context network that uses dilated temporal convolutions as the primary encoder and multilayer perceptron based decoders for the creation of the multi-step forecast. `DeepCoFactor` coherently aggregates the samples of the factor model $\tilde{\mathbf{y}}_{[i],\eta,t} = \mathbf{S}_{[i][b]}\hat{\mathbf{y}}_{[b],\eta,t}$. We mark in red the standard normal samples that are parameter-free, the reparametrization trick allows to apply backpropagation through the factor model outputs. `DeepCoFactor` extends upon the univariate `MQCNN`, through the cross series multi layer perceptron.

We use a residual cross series MLP to capture vector autorregressive relationships in the hierarchy with minimal modifications to the architecture:

$$\mathbf{h}^{(h)}_{[b],t} = \mathrm{CrossSeriesMLP}\left(\mathbf{h}^{(h)}_{[i],t}\right). \tag{14}$$

The `DeepCoFactor` uses a two-stage MLP decoder: the first decoder summarizes information into the horizon agnostic context $\mathbf{c}^{(ag)}_{[b],t}$ and the horizon specific context $\mathbf{c}^{(sp)}_{[b][h],t}$; and the second stage decoder transforms the contexts into the Factor model parameters $\left(\hat{\boldsymbol{\mu}}_{[b][h],t},\ \hat{\boldsymbol{\sigma}}_{[b][h],t}, \hat{\mathbf{F}}_{[b][k][h],t}\right)$. Eqn. 15 describes the operations:

$$\begin{aligned}
\mathbf{h}_{[b],t} &= \left[\mathbf{h}^{(h)}_{[b],t}, \mathbf{h}^{(s)}_{[b]}, \mathbf{h}^{(f)}_{[b],t}\right] \\
\mathbf{c}^{(ag)}_{[b],t} &= \mathrm{MLP}_3\left(\mathbf{h}_{[b],t}\right) \\
\mathbf{c}^{(sp)}_{[b][h],t} &= \mathrm{MLP}_4\left(\mathbf{h}_{[b],t}\right) \\
\left(\hat{\boldsymbol{\mu}}_{[b][h],t},\ \hat{\boldsymbol{\sigma}}_{[b][h],t}, \hat{\mathbf{F}}_{[b][k][h],t}\right) &= \mathrm{MLP}_5\left(\left[\mathbf{c}^{(sp)}_{[b][h],t}, \mathbf{c}^{(ag)}_{[b],t}, \mathbf{x}^{(f)}_{[b][t+1:t+N_h]}\right]\right).
\end{aligned} \tag{15}$$

For the last step, the network composes the factor model samples, using Eqn. 9 and aggregates them (equivalent to bottom-up reconciliation) in Eqn. 10.

We design our method to provide differentiable samples, allowing for sample-based loss differentiation with respect to both distributional parameters and neural network weights. This is achieved using Gaussian linear latent variable models and the reparameterization trick (Kingma & Welling, 2013).

3.3 Learning Objective

Let θ be a model that resides in the class of models Θ defined by the model architecture. Here θ can be thought of a non-linear function mapping from the model feature space to the parameter set for all target horizons, we have

$$(\hat{\boldsymbol{\mu}}_{[b][h],t},\ \hat{\boldsymbol{\sigma}}_{[b][h],t}, \hat{\mathbf{F}}_{[b][k][h],t}) = \theta(\mathbf{x}^{(h)}_{[b][:t]},\ \mathbf{x}^{(f)}_{[b][t+1:t+N_h]}, \mathbf{x}^{(s)}_{[b]}). \tag{16}$$

Let $\hat{Y}_{i,\eta,t}(\theta)$ be the random variable parameterized by θ. In some problems, multi-step coherent forecasts for multiple items are needed (e.g., in retail business, coherent regional demand forecasts are required for each product). Let u be the index of such item within an index set $\{1, \cdots, N_u\}$ of interest, and let $\tilde{Y}_{u,i,\eta,t}(\theta)$ be the coherent forecast random variable for target $y_{u,i,t+\eta}$. Using the reparametrization trick (Kingma & Welling, 2013), within the class of parameters Θ defined by the neural network architecture, we optimize for either CRPS

$$\min_{\theta\in\Theta} \sum_{u,i,\eta,t} \mathrm{CRPS}\left(y_{u,i,t+\eta},\ \tilde{Y}_{u,i,\eta,t}(\theta)\right) = \sum_{u,i,\eta,t} \mathbb{E}\left[|y_{u,i,t+\eta},\ \tilde{Y}_{u,i,\eta,t}|\right] - \frac{1}{2}\mathbb{E}\left[|\tilde{Y}_{u,i,\eta,t} - \tilde{Y}^{'}_{u,i,\eta,t}|\right]. \tag{17}$$

or the Energy Score

$$\min_{\theta\in\Theta} \sum_{u,t} \mathrm{ES}\left(\mathbb{P}_{u,t,[i][h]},\ \tilde{Y}_{u,t,[i][h]}(\theta)\right) = \sum_{u,t} \mathbb{E}\left[||y_{u,t,[i][h]} - \tilde{Y}_{u,t,[i][h]}||_2^{\beta}\right] - \frac{1}{2}\mathbb{E}\left[||\tilde{Y}_{u,t,[i][h]} - \tilde{Y}^{'}_{u,t,[i][h]}||_2^{\beta}\right]. \tag{18}$$

We train the model using stochastic gradient descent (`Adam` (Kingma & Ba, 2014)) with early stopping (Yao et al., 2007). Appendix C contains the network's optimization and hyperparameter selection details.

3.4 Discussion

Here, we discuss differences between `DeepCoFactor` (our method) with two end-to-end coherent probabilistic forecasting baselines in `HierE2E` (Rangapuram et al., 2021) and `DPMN` (Olivares et al., 2023).

The `HierE2E` method of Rangapuram et al. (2021) is "too general." It consists of an augmented `DeepVAR` neural network model (Flunkert et al., 2017) that produces probabilistic forecasts for all time-series in the hierarchy. `HierE2E` claims to be more general than hierarchical forecasting, since it is designed to enforce any convex constraint satisfied by the forecasts; due to the constraining operation in the method, it has to revise the optimized forecasts. It does not leverage specifics of the hierarchical constraints, which are more structured than a general convex constraint. `HierE2E` produces forecast samples from Gaussian distributions for each time-series in the hierarchy, assuming independence; since the samples are not guaranteed to be hierarchically coherent, `HierE2E` couples samples by projecting them on the space of coherent probabilistic forecasts. Both the sampling operation (Kingma & Welling, 2013) and the projection are differentiable, allowing the method to be trained end-to-end. `HierE2E` allows different distribution choices, although they are not explored in the initial paper, since Gaussians can be replaced by any distribution which can be sampled in a differentiable way, i.e., almost any continuous distribution (Ruiz et al., 2016; Figurnov et al., 2018; Jankowiak & Obermeyer, 2018). In Rangapuram et al. (2021), the projection operator ensures coherence, and correlations between bottom-levels are learned only by optimizing the neural network. In contrast, `DeepCoFactor` produces forecasts for bottom-level series only, while relying on common factors to encode correlations. This removes the need to forecast at all levels simultaneously, therefore reducing computational requirements if we are only interested in a subset of the aggregates.

On the other hand, the `DPMN` baseline (Olivares et al., 2023) is "too restrictive," in particular as a Poisson Mixture can be prone to distribution mis-specification problems. It is known that when a probability model is mis-specified, optimizing log likelihood is equivalent to minimizing Kullback–Leibler (KL) divergence with respect to the true probabilistic distribution, KL divergence measures change in probability space, while optimizing CRPS is equivalent to minimizing the Cramer-von Mises criterion (Gneiting & Raftery, 2007), which quantifies the distance with respect to the probability model in the sample space. The `DeepCoFactor` learning objective for the probabilistic model is resilient to distributional mis-specification (Bellemare et al., 2017). Moreover `DeepCoFactor` can be optimized to adapt for other evaluation metrics of interest.

Dataset	# Items (N_u)	Bottom (N_b)	Levels	Aggregated ($N_a + N_b$)	Time range	Frequency	Horizon (N_h)
Labour	1	32	4	57	2/1978-12/2019	Monthly	12
Traffic	1	200	4	207	1/2008-3/2009	Daily	1
Tourism-S	1	56	4	89	1998-2006	Quarterly	4
Tourism-L	1	304	4/5	555	1998-2016	Monthly	12
Wiki	1	150	5	199	1/2016-12/2016	Daily	7
Favorita	4036	54	4	93	1/2013 - 8/2017	Daily	34

Table 2: Summary of publicly-available data used in our empirical evaluation.

Finally DPMN estimates the covariance among time series, but it does not leverage multivariate inputs when encoding the historical time series. Similar to other ARIMA based baselines, on specific hierarchical benchmark datasets such as Traffic, DPMN produces sub-optimal bottom-series forecasts. We improve the encoder for historical time series by adding a CrossSeriesMLP after the Temporal convolution encoder, which bridges the accuracy gap between HierE2E and our MQCNN based approach.

4 Empirical Evaluation

In this section, we present our main empirical results. First, we describe the empirical set up. Second, we evaluate the proposed model, and compare with state-of-the-art hierarchical forecast models. Third, we present ablation study results that further analyze the source of improvements on variants of the DeepCoFactor.

4.1 Setting

Datasets. We analyze six qualitatively different public datasets: Labour, Traffic, Tourism-S, Tourism-L, Wiki, and Favorita, each requiring significant modeling flexibility due to their varied properties. The Favorita dataset is the largest dataset evaluated, which includes count and real-valued regional sales data, with over 340,000 series. The Tourism-S and Tourism-L datasets, report quarterly and monthly visitor numbers to Australian regions, respectively, and they are grouped by region and travel purpose. The Traffic dataset contains daily highway occupancy rates from San Francisco Bay Area, featuring highly correlated series with strong Granger causalities. The smallest Labour dataset tracks monthly Australian employment by status, gender, and geography, the series in this dataset are highly cointegrated. Lastly, the Wiki dataset summarizes daily online article views by country, topic, and access type. We provide more dataset details in Appendix D.

Evaluation metrics. Our main evaluation metric is the mean scaled CRPS from Eqn. 5 defined as the score described in Eqn. 19, divided by the sum of all target values. Let $\boldsymbol{l}^{(g)}$ be a vector of length $N_a + N_b$ consisting of binary indicators for a hierarchical level g, where for each $j \in [i]$, $l_j^{(g)} = 1$ if aggregated series j is included in hierarchical level g, and 0 otherwise. Then sCRPS for hierachical level g is defined as

$$\mathrm{sCRPS}\left(\mathbf{y}_{[i][t+1:t+N_h]},\ \tilde{Y}_{[i][h],t} \mid \boldsymbol{l}^{(g)}\right) = \frac{\sum_{i=1}^{N_a+N_b}\left(\sum_{\eta=1}^{N_h}\mathrm{CRPS}(y_{i,t+\eta}, \tilde{Y}_{i,\eta,t})\right)\cdot l_i^{(g)}}{\sum_{i=1}^{N_a+N_b}||\mathbf{y}_{i,[t+1:t+N_h]}||_1 \cdot l_i^{(g)}}. \tag{19}$$

Baseline Models. We compare our method with the following coherent probabilistic methods: (1) DPMN-GroupBU (Olivares et al., 2023), (2) HierE2E (Rangapuram et al., 2021), (3) ARIMA-PERMBU-MinT (Ben Taieb et al., 2017), (4) ARIMA-Bootstrap-BU (Panagiotelis et al., 2023) and (5) an ARIMA. In addition, in Appendix G we compare our method with the following coherent mean methods: (1) DPMN-GroupBU, (2) ARIMA-ERM (Ben Taieb & Koo, 2019), (3) ARIMA-MinT (Wickramasuriya et al., 2019), (4) ARIMA-BU, (5) an ARIMA and (6) Seasonal Naive. We use the implementation of statistical methods available in StatsForecast and HierarchicalForecast libraries (Olivares et al., 2022c; Garza et al., 2022).

Table 3: Empirical evaluation of probabilistic coherent forecasts. Mean *scaled CRPS* (sCRPS) averaged over 5 runs, at each aggregation level, the best result is highlighted (lower values are preferred). We report 95% confidence intervals, the methods without standard deviation have deterministic solutions.

* The `HierE2E` results differ from Rangapuram et al. (2021), here the sCRPS quantile interval space has a finer granularity of 1 percent instead of 5 percent in Rangapuram et al. (2021). We run `HierE2E`'s best reported hyperparameters on `Tourism-S` with horizon=4, instead of `HierE2E`'s original horizon=8.

** `PERMBU-MinT` on `Tourism-L` is unavailable because its implementation cannot naively be applied to datasets with multiple hierarchies, this is because the bottom up aggregation strategy cannot identify a unique way to obtain the upper level distributions.

Data	Level	DeepCoFactor (crps)	DeepCoFactor (energy)	DPMN-GroupBU	HierE2E *	PERMBU-MinT **	Bootstrap-BU	ARIMA (not coherent)
Labour	Overall	0.0074±0.0005	0.0075±0.0002	0.0102±0.0011	0.0171±.0003	**0.0067±0.0001**	0.0076±0.0001	0.0070
	1 (geo.)	0.0032±0.0010	0.0035±0.0006	0.0033±0.0011	0.0052±.0003	**0.0013±0.0001**	0.0017±0.0001	0.0017
	2 (geo.)	0.0054±0.0005	0.0054±0.0004	0.0088±0.0009	0.0181±.0003	0.0045±0.0001	0.0053±0.0001	**0.0044**
	3 (geo.)	0.0075±0.0003	**0.0074±0.0004**	0.0115±0.0009	0.0188±.0003	0.0076±0.0001	0.0086±0.0001	0.0076
	4 (geo.)	0.0137±0.0004	0.0138±0.0002	0.0173±0.0020	0.0262±.0004	**0.0133±0.0001**	0.0148±0.0001	0.0138
Traffic	Overall	**0.0171±0.0036**	0.0228±0.0075	0.0907±0.0024	0.0375±0.0058	0.0677±0.0061	0.0736±0.0024	0.0751
	1 (geo.)	**0.0026±0.0012**	0.0064±0.0077	0.0397±0.0044	0.0183±0.0091	0.0331±0.0085	0.0468±0.0031	0.0376
	2 (geo.)	**0.0029±0.0014**	0.0064±0.0067	0.0537±0.0024	0.0183±0.0081	0.0341±0.0081	0.0483±0.0030	0.0412
	3 (geo.)	**0.0044±0.0022**	0.0075±0.0059	0.0538±0.0022	0.0209±0.0071	0.0417±0.0061	0.0530±0.0025	0.0549
	4 (geo.)	**0.0587±0.0106**	0.0709±0.0104	0.2155±0.0022	0.0974±0.0021	0.1621±0.0027	0.1463±0.0017	0.1665
Tourism-S	Overall	**0.0631±0.0012**	0.0669±0.0022	0.0740±0.0122	0.0761±0.0007	0.0812±0.0010	0.0703±0.0017	0.0759
	1 (geo.)	**0.0268±0.0031**	0.0304±0.0049	0.0329±0.0074	0.0400±0.0009	0.0375±0.0014	0.0335±0.0026	0.0354
	2 (geo.)	**0.0484±0.0019**	0.0507±0.0036	0.0502±0.0058	0.0609±0.0012	0.0679±0.0016	0.0507±0.0023	0.0709
	3 (geo.)	**0.0784±0.0020**	0.0817±0.0027	0.0914±0.0171	0.0914±0.0008	0.0959±0.0012	0.0845±0.0016	0.0848
	4 (geo.)	**0.0989±0.0017**	0.1050±0.0031	0.1212±0.0238	0.1122±0.0007	0.1234±0.0012	0.1124±0.0013	0.1125
Tourism-L	Overall	**0.1197±0.0037**	0.1268±0.0045	0.1249±0.0020	0.1472±0.0029	-	0.1375±0.0013	0.1416
	1 (geo.)	0.0292±0.0042	0.0294±0.0080	0.0431±0.0040	0.0842±0.0051	-	0.0622±0.0026	**0.0263**
	2 (geo.)	**0.0593±0.0049**	0.0598±0.0059	0.0637±0.0032	0.1012±0.0029	-	0.0820±0.0019	0.0904
	3 (geo.)	**0.1044±0.0030**	0.1077±0.0054	0.1084±0.0033	0.1317±0.0022	-	0.1207±0.0010	0.1389
	4 (geo.)	**0.1540±0.0046**	0.1620±0.0060	0.1554±0.0025	0.1705±0.0023	-	0.1646±0.0007	0.1878
	5 (prp.)	**0.0594±0.0076**	0.0600±0.0040	0.0700±0.0038	0.0995±0.0061	-	0.0788±0.0018	0.0770
	6 (prp.)	0.1100±0.0049	0.1140±0.0034	**0.1070±0.0023**	0.1336±0.0042	-	0.1268±0.0017	0.1270
	7 (prp.)	**0.1824±0.0024**	0.1934±0.0045	0.1887±0.0032	0.1955±0.0025	-	0.1949±0.0010	0.2022
	8 (prp.)	**0.2591±0.0050**	0.2902±0.0054	0.2629±0.0034	0.2615±0.0016	-	0.2698±0.0004	0.2834
Wiki	Overall	**0.2475±0.0076**	0.3184±0.0262	0.3158±0.0240	0.2592±0.0031	0.4008±0.0046	0.2816±0.0036	0.3907
	1 (geo.)	**0.0823±0.0161**	0.1536±0.0666	0.1709±0.0354	0.1007±0.0046	0.1886±0.0129	0.1630±0.0065	0.1981
	2 (geo.)	**0.1765±0.0154**	0.2381±0.0171	0.2299±0.0241	0.1963±0.0037	0.2691±0.0073	0.2192±0.0043	0.2566
	3 (geo.)	**0.2743±0.0108**	0.3213±0.0343	0.3311±0.0230	0.2784±0.0038	0.4049±0.0062	0.2923±0.0032	0.4100
	4 (geo.)	**0.2814±0.0100**	0.3326±0.0332	0.3370±0.0223	0.2900±0.0043	0.4236±0.0062	0.2999±0.0032	0.4182
	5 (prp.)	**0.4231±0.0127**	0.5466±0.1406	0.5098±0.0667	0.4307±0.0039	0.7177±0.0064	0.4339±0.0039	0.6708
Favorita	Overall	**0.2908±0.0025**	0.3279±0.0052	0.4020±0.0182	0.5298±0.0091	0.4670±0.0096	0.4110±0.0085	0.4373
	1 (geo.)	**0.1841±0.0033**	0.2209±0.0045	0.2760±0.0149	0.4714±0.0103	0.2692±0.0076	0.2900±0.0067	0.3112
	2 (geo.)	**0.2754±0.0026**	0.3112±0.0051	0.3865±0.0207	0.5182±0.0107	0.3824±0.0092	0.3877±0.0082	0.4183
	3 (geo.)	**0.2945±0.0025**	0.3313±0.0053	0.4068±0.0206	0.5291±0.0129	0.6838±0.0108	0.4490±0.0098	0.4446
	4 (geo.)	**0.4092±0.0022**	0.4484±0.0068	0.5387±0.0253	0.6012±0.0131	0.5532±0.0116	0.5749±0.0003	0.5749

4.2 Forecasting Results

As mentioned earlier, we compare the proposed model to the DPMN (Olivares et al., 2023), the HierE2E (Rangapuram et al., 2021), and two ARIMA-based reconciliation methods (Wickramasuriya et al., 2019; Panagiotelis et al., 2023). Following previous work, we report the sCRPS at all levels of the defined hierarchies; see Table 3. The ARIMA reconciliation results are generated using Olivares et al. (2022c), with confidence interval computed based on 10 independent runs. Results for HierE2E are generated based on three independent runs using hyperparameters tuned by Olivares et al. (2022c). All metrics for DPMN are quoted from Olivares et al. (2023) with identical experimental setting on all datasets.

As the *Overall* row in Table 3 shows, DeepCoFactor improves sCRPS upon the best alternative in five out of six datasets, with gains of on Favorita of 27.67%, on Tourism-S of 10.24%, on Tourism-L of 4.16%, on Wiki of 4.5% and 54.40% , Tourism-L and Traffic. And a degradation of 8% over the best performing reconciliation on Labour, the smallest dataset. For five out of six datasets our model achieves the best or second best sCRPS accuracy across all the levels of the hierarchy, it is important to consider that aggregate levels are much smaller in sample size for which we prefer the bottom-level measurements as an indicator of the methods' accuracy.

On Traffic, our model achieves remarkably better results than DPMN and HierE2E baselines, which we explain by the ability to model VAR relationships accurately consequence of smoothing the time series features before using them as inputs for other series. For the smallest dataset, Australian Labour, all deep learning models, including DeepCoFactor, showed a decline in accuracy compared to statistical baselines. We hypothesize this performance drop is due to the limited data available, as 57 monthly series may not be sufficient to effectively train complex models like deep learning architectures. Another explanation for the Labour degradation is the presence of strong trends and cycles, which can lead to artificially high correlations between the series, resulting in spurious relationships that may confuse the VAR modules.

We complement the main results with a mean forecast evaluation section in Appendix G. As we see Table 7 the sCRPS accuracy gains are mostly mirrored on the relative squared error metric (relSE) . We qualitatively show DeepCoFactor forecast distributions for a hierarchical structure in the Favorita dataset in Appendix E. We also include in Appendix E a probability-probability comparing the similarity of the empirical and forecast distributions, as it can be seen in Figure 7 the forecast distributions are qualitatively well calibrated.

4.3 Ablation Studies

To analyze DeepCoFactor's source of improvements, we performed two ablation studies on the Traffic dataset, where we investigate first the effects of the learning objective, and second the effects of VAR inputs enabled by the CrossSeriesMLP. Here we report a summary and refer to details in Appendix F.

In the first study, we explore the effects of a cross-series MLP that mimics the vector autoregressive model, we compare the DeepCoFactor architecture with and without the cross series multilayer perceptron (CrossSeriesMLP) introduced in Eqn. 14. Such a module enables the network to share information of the series in the hierarchy with minimal modifications in the architecture. Table 5, and Figure 3 show the CrossSeriesMLP improved Traffic forecasting accuracy by 66% when compared to variants without it. We attribute the effectiveness of the VAR approach to the presence of Granger-causal relationships in the traffic intersections. The VAR augmented DeepCoFactor improves upon well-established univariate architectures, including LSTM (Sak et al., 2014), NBEATS (Oreshkin et al., 2020; Olivares et al., 2022a), NHITS (Challu et al., 2023), TFT (Lim et al., 2021) and FCGAGA (Oreshkin et al., 2021) a spatio-temporal specialized architecture.

In the second ablation study on the learning objectives, we compared the CRPS-based optimization, as described in Eqn. 17, with the negative log-likelihood estimation for the Gaussian factor model introduced in Section 3.1, and other likelihood-estimated distributions. Table 6, and Figure 3 show that the CRPS-optimized factor model, improves forecasting accuracy by nearly 60% when compared to the log likelihood optimized.

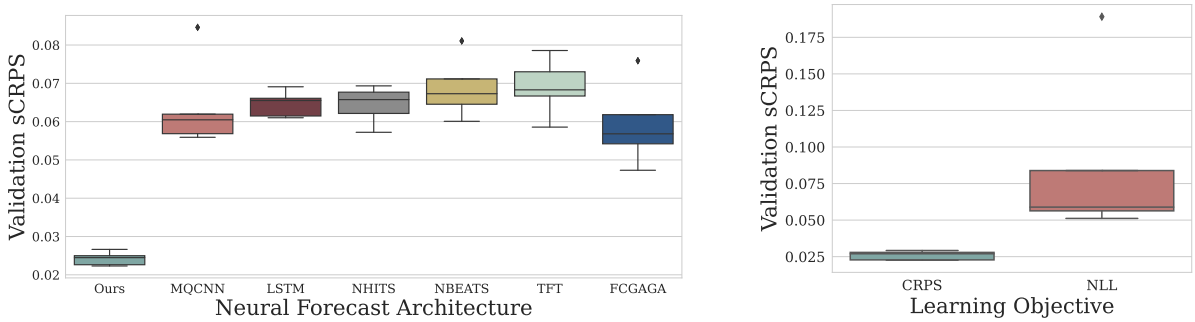


Figure 3: Ablation studies on the Bay Area `Traffic` dataset: a) In highly correlated hierarchies, VAR inputs enabled by the `DeepCoFactor` to significantly improve over the univariate `MQCNN`'s accuracy. b) The factor model CRPS learning objective demonstrates clear advantages over classic negative log-likelihood. Full ablation studies described in Appendix F.

5 Conclusion

In this work, we present a novel multivariate factor forecasting model, integrated with the MQCNN neural network architecture, resulting in the *Deep Coherent Factor Model Neural Network* (`DeepCoFactor`) model. By optimizing CRPS as the learning objective, we achieve significant improvements in forecast accuracy over traditional negative log-likelihood objectives.

Our experiments on six benchmark datasets—Favorita grocery demand, Australian quarterly and monthly tourism, Australian labour, Wikipedia article visits, and Bay Area Traffic—showed consistent improvements in CRPS accuracy, averaging over 15 percent. We observed a 4 percent performance drop on the smaller Australian labour dataset. Ablation studies further confirmed the importance of the CRPS learning objective and the enhancements made to support multivariate time series inputs, which were key to the model's success.

Our model's success on datasets with strong Granger causality is due to its vector autoregressive capabilities. However, in the presence of cointegrated series, vector autorregresive approaches can be ineffective. Developing multivariate time series methods that are robust to cointegration is a promising area for future research.

This study uses the reparametrization trick to enable alternative learning objectives like CRPS and the energy score for neural forecasting tasks and the use of coherent factor models to capture correlations among hierarchical series structures. While we focus on parametrizing the multivariate predictive distribution as a Gaussian multivariate factor model, the frameworks flexibility can also accommodate other distributions that support sample differentiability. This is of special interest for outlier quantiles that cannot be well approximated by Gaussian variables. Exciting future research directions include extending the reparameterization trick to handle discrete distributions, which could further enhance the accuracy of forecast distributions built on this framework. Another promising direction for future research is to extend the usage of the reparameterization trick from the learning objectives, into the hierarchical aggregation structure itself, provided the aggregation structure is done through differentiable transformations.

Finally we observed performance challenges of neural forecasting models when working with limited dataset sizes. Traditional tools like data augmentation and pretraining need to be adapted to handle multivariate time series and associated covariates. Extending these techniques to the hierarchical forecasting domain is a promising research direction.

References

Oren Anava, Vitaly Kuznetsov, and (Google Inc. Sponsorship). Web traffic time series forecasting, forecast future traffic to wikipedia pages. Kaggle Competition, 2018. URL `https://www.kaggle.com/c/web-traffic-time-series-forecasting/`.

Australian Bureau of Statistics. Labour force, australia. Accessed Online, 2019. URL https://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/6202.0Dec%202019?OpenDocument.

Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

Souhaib Ben Taieb and Bonsoo Koo. Regularized regression for hierarchical forecasting without unbiasedness conditions. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 1337–1347, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330976. URL https://doi.org/10.1145/3292500.3330976.

Souhaib Ben Taieb, James W. Taylor, and Rob J. Hyndman. Coherent probabilistic forecasts for hierarchical time series. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3348–3357. PMLR, 06–11 Aug 2017. URL http://proceedings.mlr.press/v70/taieb17a.html.

Souhaib Ben Taieb, James W. Taylor, and Rob J Hyndman. Hierarchical probabilistic forecasting of electricity demand with smart meter data. *Journal of the American Statistical Association*, 116:27 – 43, 2020. URL https://api.semanticscholar.org/CorpusID:43214772.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza, Max Mergenthaler, and Artur Dubrawski. NHITS: Neural Hierarchical Interpolation for Time Series forecasting. In *The Association for the Advancement of Artificial Intelligence Conference 2023 (AAAI 2023)*, 2023. URL https://arxiv.org/abs/2201.12886.

Byron J. Dangerfield and John S. Morris. Top-down or bottom-up: Aggregate versus disaggregate extrapolations. *International Journal of Forecasting*, 8(2):233–241, 1992. URL https://www.sciencedirect.com/science/article/pii/0169207092901210.

A. Das, W. Kong, B. Paria, and R. Sen. Dirichlet proportions model for hierarchically coherent probabilistic forecasting. In Robin J. Evans and Ilya Shpitser (eds.), *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pp. 518–528. PMLR, 31 Jul–04 Aug 2023. URL https://proceedings.mlr.press/v216/das23b.html.

Priya L Donti, David Rolnick, and J Zico Kolter. Dc3: A learning method for optimization with hard constraints. *arXiv preprint arXiv:2104.12225*, 2021.

Corporación Favorita, inversion, Julia Elliott, and Mark McDonald. Corporación favorita grocery sales forecasting, 2017. URL https://kaggle.com/competitions/favorita-grocery-sales-forecasting.

Michael Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. In *Neural Information Processing Systems*, 2018.

Valentin Flunkert, David Salinas, and Jan Gasthaus. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *ArXiv*, abs/1704.04110, 2017.

Federico Garza, Max Mergenthaler Canseco, Cristian Challú, and Kin G. Olivares. StatsForecast: Lightning fast forecasting with statistical and econometric models. PyCon Salt Lake City, Utah, US 2022, 2022. URL https://github.com/Nixtla/statsforecast.

Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1(1):125–151, 2014. URL https://doi.org/10.1146/annurev-statistics-062713-085831.

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

Xing Han, Sambarta Dasgupta, and Joydeep Ghosh. Simultaneously reconciled quantile forecasting of hierarchically related time series. In *International Conference on Artificial Intelligence and Statistics*, 2021.

Tao Hong, Pierre Pinson, and Shu Fan. Global energy forecasting competition 2012, 2014.

Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679 – 688, 2006. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2006.03.001. URL http://www.sciencedirect.com/science/article/pii/S0169207006000239.

Rob J. Hyndman, Roman A. Ahmed, George Athanasopoulos, and Han Lin Shang. Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis*, 55(9):2579–2589, 2011. URL https://www.sciencedirect.com/science/article/pii/S0167947311000971.

Rob J. Hyndman, Alan J. Lee, and Earo Wang. Fast computation of reconciled forecasts for hierarchical and grouped time series. *Comput. Stat. Data Anal.*, 97(C):16–32, may 2016. URL https://doi.org/10.1016/j.csda.2015.11.007.

Rob J Hyndman, George Athanasopoulos, Azul Garza, Cristian Challu, Max Mergenthaler, and Kin G. Olivares. *Forecasting: Principles and Practice*. OTexts, Melbourne, Australia, 2024. available at https://otexts.com/fppy/.

Martin Jankowiak and Fritz Obermeyer. Pathwise derivatives beyond the reparameterization trick. *ArXiv*, abs/1806.01851, 2018.

Jooyoung Jeon, Anastasios Panagiotelis, and Fotios Petropoulos. Probabilistic forecast reconciliation with applications to wind power and electric load. *European Journal of Operational Research*, 279(2):364–379, 2019.

Harshavardhan Kamarthi, Lingkai Kong, Alexander Rodríguez, Chao Zhang, and B Aditya Prakash. Profhit: Probabilistic robust forecasting for hierarchical time-series. *arXiv preprint arXiv:2206.07940*, 2022.

Harshavardhan Kamarthi, Aditya B. Sasanur, Xinjie Tong, Xingyu Zhou, James Peters, Joe Czyzyk, and B. Aditya Prakash. Large scale hierarchical industrial demand time-series forecasting incorporating sparsity. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 5230–5239, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671632. URL https://doi.org/10.1145/3637528.3671632.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

Francesco Laio and Stefania Tamea. Verification tools for probabilistic forecasts of continuous hydrological variables. *Hydrology and Earth System Sciences*, 11(4):1267–1277, 2007.

Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2021.03.012. URL https://www.sciencedirect.com/science/article/pii/S0169207021000637.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4):1346–1364, 2022. URL https://www.sciencedirect.com/science/article/pii/S0169207021001874. Special Issue: M5 competition.

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

Konstantin Mishchenko, Mallory Montgomery, and Federico Vaggi. A self-supervised approach to hierarchical forecasting with applications to groupwise synthetic controls. *ArXiv*, abs/1906.10586, 2019.

Kin G. Olivares, Cristian Challu, Grzegorz Marcjasz, Rafał Weron, and Artur Dubrawski. Neural basis expansion analysis with exogenous variables: Forecasting electricity prices with NBEATSx. *International Journal of Forecasting*, 2022a. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2022.03.001. URL https://www.sciencedirect.com/science/article/pii/S0169207022000413.

Kin G. Olivares, Cristian Challú, Federico Garza, Max Mergenthaler Canseco, and Artur Dubrawski. NeuralForecast: User friendly state-of-the-art neural forecasting models. PyCon Salt Lake City, Utah, US 2022, 2022b. URL https://github.com/Nixtla/neuralforecast.

Kin G. Olivares, Federico Garza, David Luo, Cristian Challú, Max Mergenthaler, Souhaib Ben Taieb, Shanika L. Wickramasuriya, and Artur Dubrawski. HierarchicalForecast: A reference framework for hierarchical forecasting in python. *Work in progress paper, submitted to Journal of Machine Learning Research.*, abs/2207.03517, 2022c. URL https://arxiv.org/abs/2207.03517.

Kin G. Olivares, O. Nganba Meetei, Ruijun Ma, Rohan Reddy, Mengfei Cao, and Lee Dicker. Probabilistic hierarchical forecasting with deep poisson mixtures. *International Journal of Forecasting*, 2023. URL https://www.sciencedirect.com/science/article/pii/S0169207023000432.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: neural basis expansion analysis for interpretable time series forecasting. In *8th International Conference on Learning Representations, ICLR 2020*, 2020. URL https://openreview.net/forum?id=r1ecqn4YwB.

Boris N. Oreshkin, Arezou Amini, Lucy Coyle, and Mark J. Coates. FC-GAGA: Fully connected gated graph architecture for spatio-temporal forecasting. In *The Association for the Advancement of Artificial Intelligence Conference 2021 (AAAI 2021)*, 2021. URL https://arxiv.org/abs/2007.15531.

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.

Biswajit Paria, Rajat Sen, Amr Ahmed, and Abhimanyu Das. Hierarchically Regularized Deep Forecasting. In *Submitted to Proceedings of the 39th International Conference on Machine Learning*. PMLR. Working Paper version available at arXiv:2106.07630, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Syama Sundar Rangapuram, Lucien D Werner, Konstantinos Benidis, Pedro Mercado, Jan Gasthaus, and Tim Januschowski. End-to-end learning of coherent probabilistic forecasts for hierarchical time series. In *International Conference on Machine Learning*, pp. 8832–8843. PMLR, 2021.

Francisco J. R. Ruiz, Michalis K. Titsias, and David M. Blei. The generalized reparameterization gradient. In *NIPS*, 2016.

Hasim Sak, Andrew W. Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *Computing Research Repository*, abs/1402.1128, 2014. URL http://arxiv.org/abs/1402.1128.

Thordis L. Thorarinsdottir and Tilmann Gneiting. Probabilistic Forecasts of Wind Speed: Ensemble Model Output Statistics by using Heteroscedastic Censored Regression. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 173(2):371–388, 11 2009. ISSN 0964-1998. doi: 10.1111/j.1467-985X.2009.00616.x. URL https://doi.org/10.1111/j.1467-985X.2009.00616.x.

Tourism Australia, Canberra. Tourism Research Australia (2005), Travel by Australians. https://www.kaggle.com/luisblanche/quarterly-tourism-in-australia/, 2005.

Ryo Umagami, Yu Ono, Yusuke Mukuta, and Tatsuya Harada. Hiperformer: Hierarchically permutation-equivariant transformer for time series forecasting, 2023. URL https://arxiv.org/abs/2305.08073.

Steven R. Vitullo. Disaggregating time series data for energy consumption by aggregate and individual customer. *Department of Electrical and Computer Engineering, Ph. D. Dissertation.*, 2011.

Yuyang Wang, Alex Smola, Danielle Maddix, Jan Gasthaus, Dean Foster, and Tim Januschowski. Deep factors for forecasting. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6607–6617. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/wang19k.html.

Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. A Multi-horizon Quantile Recurrent Forecaster. In *31st Conference on Neural Information Processing Systems NIPS 2017, Time Series Workshop*, 2017. URL https://arxiv.org/abs/1711.11053.

Shanika L. Wickramasuriya. Probabilistic forecast reconciliation under the Gaussian framework. *Accepted at Journal of Business and Economic Statistics*, 2023.

Shanika L. Wickramasuriya, George Athanasopoulos, and Rob J. Hyndman. Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *Journal of the American Statistical Association*, 114(526):804–819, 2019.

Yuan Yao, Lorenzo Rosasco, and Caponneto Andrea. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007. URL https://doi.org/10.1007/s00365-006-0663-2.

Michaël Zamo and Philippe Naveau. Estimation of the Continuous Ranked Probability Score with Limited Information and Applications to Ensemble Weather Forecasts. *Mathematical Geosciences*, 50(2):209–234, February 2018. doi: 10.1007/s11004-017-9709-7. URL https://hal.science/hal-02976423.

A Multivariate Factor Model Coherence and Covariance

A.1 Coherent Aggregation Properties

The coherence of `DeepCoFactor` is a special case of the bootstrap sample reconciliation technique (Panagiotelis et al., 2023), as explored by Olivares et al. (2023).

Lemma A.1. *Let $(\Omega_{[b]}, \mathcal{F}_{[b]}, \mathbb{P}_{[b]})$ be a probabilistic forecast space, with $\mathcal{F}_{[b]}$ a σ-algebra on $\Omega_{[b]}$. If a forecast distribution $\mathbb{P}_{[i]}$ assigns a zero probability to sets that don't contain coherent forecasts, it defines a coherent probabilistic forecast space $(\Omega_{[i]}, \mathcal{F}_{[i]}, \mathbb{P}_{[i]})$ with $\Omega_{[i]} = \mathbf{S}_{[i][b]}(\Omega_{[b]})$.*

$$\mathbb{P}_{[a]}\left(\mathbf{y}_{[a]} \notin \mathbf{A}_{[a][b]}(\mathcal{B}) \mid \mathcal{B}\right) = 0 \implies \mathbb{P}_{[i]}\left(\mathbf{S}_{[i][b]}(\mathcal{B})\right) = \mathbb{P}_{[b]}\left(\mathcal{B}\right) \quad \forall \mathcal{B} \in \mathcal{F}_{[b]}. \tag{20}$$

Proof. We note the following:

$$\begin{aligned}\mathbb{P}_{[i]}\left(\mathbf{S}_{[i][b]}(\mathcal{B})\right) &= \mathbb{P}_{[i]}\left(\begin{bmatrix}\mathbf{A}_{[a][b]}\\ \mathbf{I}_{[b][b]}\end{bmatrix}(\mathcal{B})\right) = \mathbb{P}_{[i]}\left(\{\begin{bmatrix}\mathbf{A}_{[a][b]}(\mathcal{B})\\ \mathbb{R}^{N_b}\end{bmatrix}\} \cap \{\begin{bmatrix}\mathbb{R}^{N_a}\\ \mathcal{B}\end{bmatrix}\}\right)\\ &= \mathbb{P}_{[a]}\left(\mathbf{A}_{[a][b]}(\mathcal{B}) \mid \mathcal{B}\right)\mathbb{P}_{[b]}\left(\mathcal{B}\right) = (1 - \mathbb{P}_{[a]}\left(\mathbf{y}_{[a]} \notin \mathbf{A}_{[a][b]}(\mathcal{B}) \mid \mathcal{B}\right)) \times \mathbb{P}_{[b]}\left(\mathcal{B}\right) = \mathbb{P}_{[b]}\left(\mathcal{B}\right).\end{aligned}$$

The first equality is the image of a set $\mathcal{B} \in \Omega_{[b]}$ corresponding to the constraints matrix transformation, the second equality defines the spanned space as a subspace intersection of the aggregate series and the bottom series, the third equality uses the conditional probability multiplication rule, the final equality uses the zero probability assumption.

By construction of the samples of our model $\tilde{\mathbf{y}}_{[i]} = \mathbf{S}_{[i][b]}\left(\hat{\mathbf{y}}_{[b]}\right)_{+}$ and $\tilde{\mathbf{y}}_{[a]} = \mathbf{A}_{[a][b]}\left(\hat{\mathbf{y}}_{[b]}\right)_{+}$, satisfying the assumptions of the lemma and proving the coherence of our approach.

□

A.2 Covariance Structure

Here we prove the covariance structure of our factor model introduced in Section 3.1.

Lemma A.2. *Let our factor model be defined by*

$$\hat{\mathbf{y}}_{[b],\eta,t} = \hat{\boldsymbol{\mu}}_{[b],\eta,t} + \mathrm{Diag}(\hat{\boldsymbol{\sigma}}_{[b],\eta,t})\mathbf{z}_{[b],\eta,t} + \hat{\mathbf{F}}_{[b][k],\eta,t}\boldsymbol{\epsilon}_{[k],\eta,t}, \qquad \eta = 1, \cdots, N_h, \tag{21}$$

with independent factors $\mathbf{z}_{[b],\eta} \sim \mathcal{N}(\mathbf{0}_{[b]}, \mathbf{I}_{[b][b]})$, and $\boldsymbol{\epsilon}_{[k],\eta} \sim \mathcal{N}(\mathbf{0}_{[k]}, \mathbf{I}_{[k][k]})$, its covariance satisfies

$$\mathrm{Cov}\left(\hat{\mathbf{y}}_{[b],\eta,t}\right) = \mathrm{Diag}(\hat{\boldsymbol{\sigma}}^2_{[b],\eta,t}) + \hat{\mathbf{F}}_{[b][k],\eta,t}\hat{\mathbf{F}}^{\top}_{[b][k],\eta,t}. \tag{22}$$

Proof. First, we observe that

$$\begin{aligned}\mathrm{Cov}\left(\hat{\mathbf{y}}_{[b],\eta,t}, \hat{\mathbf{y}}_{[b],\eta,t}\right) &= \mathrm{Cov}\left(\mathrm{Diag}(\hat{\boldsymbol{\sigma}}_{[b],\eta,t})\mathbf{z}_{[b],\eta,t}, \mathrm{Diag}(\hat{\boldsymbol{\sigma}}_{[b],\eta,t})\mathbf{z}_{[b],\eta,t}\right)\\ &+ \quad 2\mathrm{Cov}\left(\mathrm{Diag}(\hat{\boldsymbol{\sigma}}_{[b],\eta,t})\mathbf{z}_{[b],\eta,t}, \hat{\mathbf{F}}_{[b][k],\eta,t}\boldsymbol{\epsilon}_{[k],\eta,t}\right)\\ &+ \quad \mathrm{Cov}\left(\hat{\mathbf{F}}_{[b][k],\eta,t}\boldsymbol{\epsilon}_{[k],\eta,t}, \hat{\mathbf{F}}_{[b][k],\eta,t}\boldsymbol{\epsilon}_{[k],\eta,t}\right).\end{aligned} \tag{23}$$

By bilinearity of covariance and independence of the sampled factors, it follows that

$$\mathrm{Cov}\left(\hat{\mathbf{y}}_{[b],\eta,t}, \hat{\mathbf{y}}_{[b],\eta,t}\right) = \mathrm{Diag}(\hat{\boldsymbol{\sigma}}_{[b],\eta,t})\mathrm{Cov}\left(\mathbf{z}_{[b],\eta,t}, \mathbf{z}_{[b],\eta,t}\right)\mathrm{Diag}(\hat{\boldsymbol{\sigma}}_{[b],\eta,t})^{\intercal} + \hat{\mathbf{F}}_{[b][k],\eta,t}\mathrm{Cov}\left(\boldsymbol{\epsilon}_{[k],\eta,t}, \boldsymbol{\epsilon}_{[k],\eta,t}\right)\hat{\mathbf{F}}^{\intercal}_{[b][k],\eta,t}.$$

We conclude that

$$\mathrm{Cov}\left(\hat{\mathbf{y}}_{[b],\eta,t}, \hat{\mathbf{y}}_{[b],\eta,t}\right) = \mathrm{Diag}(\hat{\boldsymbol{\sigma}}^2_{[b],\eta,t}) + \hat{\mathbf{F}}_{[b][k],\eta,t}\hat{\mathbf{F}}^{\intercal}_{[b][k],\eta,t}. \tag{24}$$

□

B Code Script for Sampling

```
def sample(self, distr_args, window_size, num_samples=None):
    """
    **Parameters**
    `distr_args`: Forecast Distribution arguments.
    `window_size`: int=1, for reconciliation reshapes in sample method.
    `num_samples`: int=500, number of samples for the empirical quantiles.

    **Returns**
    `samples`: tensor, shape [B,H,`num_samples`].
    `quantiles`: tensor, empirical quantiles defined by `levels`.
    """
    means, factor_loading, stds = distr_args
    collapsed_batch, H, _ = means.size()

    # [collapsed_batch,H]:=[B*N*Ws,H,F] -> [B,N,Ws,H,F]
    factor_loading = factor_loading.reshape(
        (-1, self.n_series, window_size, H, self.n_factors)
    ).contiguous()
    factor_loading = torch.einsum("iv,bvwhf->biwhf", self.SP, factor_loading)

    means = means.reshape(-1, self.n_series, window_size, H, 1).contiguous()
    stds = stds.reshape(-1, self.n_series, window_size, H, 1).contiguous()

    # Loading factors for covariance Diag(stds) + F F^t -> (SPF)(SPF^t)
    hidden_factor = Normal(
        loc=torch.zeros(
            (factor_loading.shape[0], window_size, H, self.n_factors),
        scale=1.0)
    sample_factors = hidden_factor.rsample(sample_shape=(self.num_samples,))
    sample_factors = sample_factors.permute(
        (1, 2, 3, 4, 0)
    ).contiguous()  # [n_items, window_size, H, F,num_samples]

    sample_loaded_factors = torch.einsum("bvwhf,bwhfn->bvwhn", factor_loading,
                                         sample_factors)
    sample_loaded_means = means + sample_loaded_factors

    # Sample Normal
    normal = Normal(loc=torch.zeros_like(sample_loaded_means), scale=1.0)
    samples = normal.rsample()
    samples = F.relu(sample_loaded_means + stds * samples)

    samples = torch.einsum("iv,bvwhn->biwhn", self.SP, samples)
    samples = samples.reshape(collapsed_batch, H, self.num_samples).contiguous()

    # Compute quantiles and mean
    quantiles_device = self.quantiles.to(means.device)
    quants = torch.quantile(input=samples, q=quantiles_device, dim=-1)
    quants = quants.permute((1, 2, 0))  # [Q,B,H] -> [B,H,Q]
    sample_mean = torch.mean(samples, dim=-1, keepdim=True)
    return samples, sample_mean, quants
```

Figure 4: PyTorch function for sampling from our Gaussian Factor model. Note that the factor samples are shared across all bottom-level distributions. The samples are differentiable with regard to the function inputs. We can easily adapt this function to sample from other distributions.

Table 4: *Deep Coherent Factor Model Neural Network* (`DeepCoFactor`) architecture hyperparameters. We use a small or large model configuration depending on the datasets' size.

Parameter	Considered Values					
	Traffic	Tourism-S	Wiki	Labour	Tourism-L	Favorita
Activation Function.	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU
Static Encoder Dimension.	5	10	5	10	20	20
Temporal Convolution Channel Size.	10	10	10	20	30	30
Future Encoder Dimension.	20	4	20	20	50	50
Horizon Specific Decoder Dimensions.	5	5	5	5	5	5
Horizon Agnostic Decoder Dimensions.	20	10	20	10	20	20
Factor Model Components.	10	10	20	20	10	5
Cross Series MLP Hidden Size.	200	100	200	0	50	5
SGD Batch Size.	1	1	1	1	1	4
SGD Effective Batch Size.	207	89	199	57	555	744
SGD Max steps.	2e3	1e3	2e3	2e3	2e3	80e3
Early Stop Patience steps.	5	5	5	4	5	-1
Learning Rate.	5e-3	5e-4	5e-4	1e-3	5e-4	5e-4

C Training Methodology and Hyperparameters

Here we complement and extend the description of our method in Section 3.

To avoid information leakage we perform ablation studies in the validation set preceding the test set, where we explored variants of the probabilistic method, as well as its optimization. We report these ablation studies in Appendix F. For each dataset, given the prediction horizon h, the test set is composed of the last h time-steps. The validation set is composed of the h time-steps preceding the test set time range. The training set is composed of all dates previous to the validation time-range. When reporting final accuracy results of our model on test set, we used the settings that perform the best in validation set.

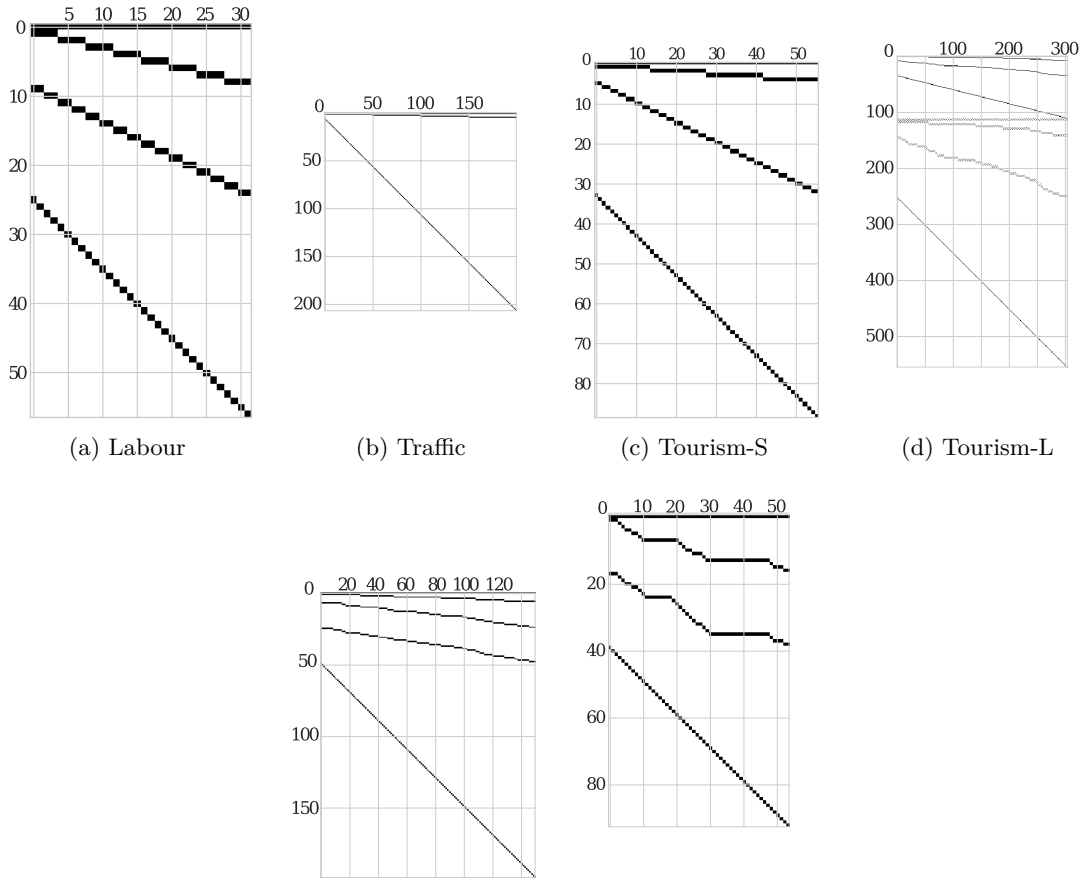
We tune minimally the architecture and its parameters varying only its size and the convolution kernel filters to match the seasonalities present in each dataset. For the `Favorita` dataset we use dilations of $[1, 2, 4, 8, 16, 32]$ to match weekly and monthly seasonalities, for the `Tourism-L` and `Labour` datasets we use dilations of $[1, 2, 3, 6, 12]$ to match the monthly and yearly seasonalities, for the `Traffic` dataset we use dilations of $[1, 7, 14, 28]$ as multiples of 7 to match the weekly seasonalities, for the `Tourism-S` dataset we use dilations of $[1, 2, 4]$ as multiples of 4 to match the quarterly seasonalities. For the `Wiki` dataset we use minimal dilations $[1, 2]$.

The selection of the number of factors follows mostly the memory constraints of the GPU, as the effective batch size implied by our probabilistic model grows rapidly as a function of the multivariate series. In the `Favorita` dataset more factors are likely to continue to improve accuracy but with the tradeoff of the computational speed. Similarly the Cross series MLP hidden size is selected following the GPU memory constraints.

We share a learning rate of 5e-4 constant across the three datasets, which shows that the method is reasonably robust across different forecasting tasks. During the optimization of the networks we use adaptive moments stochastic gradient descent (Kingma & Ba, 2014) with early stopping (Yao et al., 2007) guided by the sCRPS signal measured in the validation set. We use a learning rate scheduler that decimates the learning rate four times during the optimization (SGD Maxteps/4), to ensure the convergence of the optimization.

The `DeepCoFactor` model is implemented using Pytorch (Paszke et al., 2019), with the NeuralForecast library framework (Olivares et al., 2022b). We run all experiments using a single NVIDIA V100 GPU.

As mentioned earlier in Section 4.1 the statistical methods available in StatsForecast and HierarchicalForecast libraries (Olivares et al., 2022c; Garza et al., 2022). In particular we use available code on the hierarchical baselines repository, and hierarchical datasets repository. As described in Table 2, we deviate slightly from the experimental setting in Rangapuram et al. (2021), to ensure replicability of the results of this paper, for Table 3 we rerun the `HierE2E` (Rangapuram et al., 2021) baseline on `Labour`, `Traffic`, `Wiki` and `Tourism-S` using their best reported hyperparameters.



(a) Labour (b) Traffic (c) Tourism-S (d) Tourism-L

(e) Wiki (f) Favorita

Figure 5: Hierarchical constraints of the empirical evaluation datasets. (a) `Labour` reports 57 series number of employees by full-time status, gender and geographic levels. (b) `Traffic` organizes the occupancy series of 200 highways into quarters, halves, and totals. (c) `Tourism-S` categorizes its 89 regional visit series based on travel purpose, zones, states, and country-level aggregations and urbanization within regions. (d) `Tourism-L` categorizes its 555 regional visit series based on travel purpose, zones, states, and country-level geographical aggregations. (e) `Wiki` groups 150 daily visits to Wikipedia articles by language and article categorical taxonomy. (f) `Favorita` classifies its grocery sales by store, city, state, and country levels.

D Dataset Details

`Labour`: The Labour dataset (Australian Bureau of Statistics, 2019) tracks monthly Australian employment from February 1978 to December 2019, reporting total employees by part-time/full-time status, gender, and geography. It includes $N = 57$ series in total, with $N_a = 25$ aggregate series and $N_b = 32$ bottom-level series. We use 8 months from May 2019 to December 2019 as test, and the rest of the data as training and validation.

`Traffic`: The Traffic dataset (Ben Taieb & Koo, 2019) contains daily (aggregated from hourly rates) freeway occupancy rates for 200 car lanes in the San Francisco Bay Area, aggregated from January 2008 to March 2009. The data is grouped into three levels: four groups of 50 lanes, two groups of 100 lanes, and one overall group of 200 lanes, following the random grouping in Rangapuram et al. (2021); Olivares et al. (2023). Consistent with prior work (Ben Taieb & Koo, 2019; Rangapuram et al., 2021; Olivares et al., 2023), we split the dataset into 120 training, 120 validation, and 126 test samples, reporting accuracy for the last test date. We use geographic node dummies, weekend indicators, and proximity to Saturday for exogenous variables.

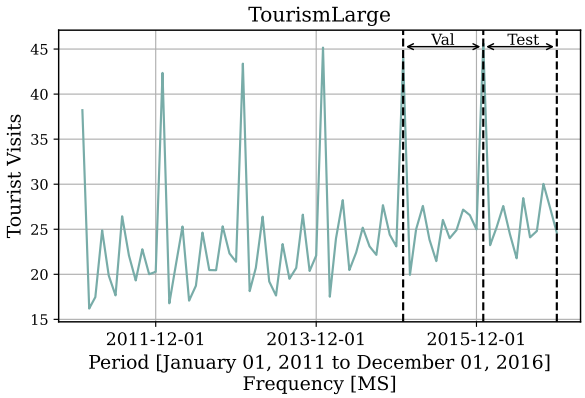


Figure 6: `Tourism-L` dataset partition into train, validation, and test sets used in our experiments. All datasets use the last horizon window as defined in Table 2 (marked by the second dotted line), and the previous window preceding the test set as validation (between the first and second dotted lines). Validation provides the signal for hyperparameter selection and the ablation studies.

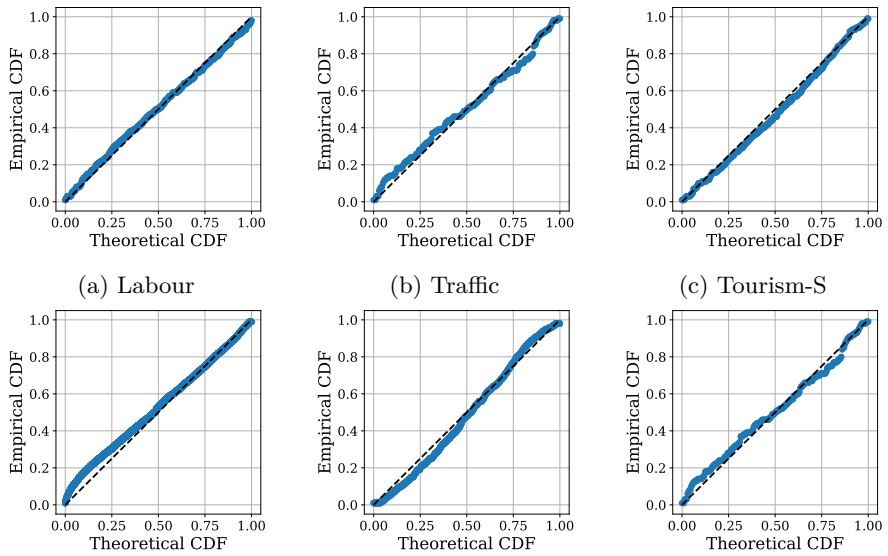
`Tourism-S`: The `Tourism-S` dataset (Tourism Australia, Canberra, 2005) records quarterly visits to Australia from 1996 to 2006. We use data from 2005 for validation, 2006 for testing, and the remaining quarters for training. Each series contains 28 quarterly observations and is structured with aggregated visit data by country, purpose of travel, state, regions and urbanization level within regions. The most disaggregated level includes 56 regions by urbanization, while the aggregated data consists of 33 series. We use state dumies, and for the future exogenous quarterly dummies and use a seasonal naive 4 and 8 anchors.

`Tourism-L`: The `Tourism-L` dataset (Wickramasuriya et al., 2019) represents visits to Australia, at a monthly frequency, between January 1998 and December 2016. We use 2015 for validation, and 2016 for testing, and all previous years for training. The dataset contains 228 monthly observations. For each month, we have the number of visits to each of Australia's 78 regions, which are aggregated to the zone, state and national level, and for each of four purposes of travel. These two dimensions of aggregation total $N = 304$ leaf entities (a region-purpose pair), with a total of $M = 555$ series in the hierarchy. We pre-process the data to include static features we use purpose of travel as well as state dummies, for the historical information we use month dummies, and for the future exogenous we use month and a seasonal naive anchor forecast that helps greatly to account for the series seasonality.

`Favorita`: The Favorita dataset (Favorita et al., 2017) contains grocery sales of the Ecuatorian Corporación Favorita in $N = 54$ stores. We perform geographical aggregation of the sales at the store, city, state and national levels, following (Olivares et al., 2023). This yields a total of $M = 94$ aggregates. Concerning features, we use past unit sales and number of transactions as historical data. In the `Favorita` dataset we include item perishability static information, geographic state dummy variables, and for the historic exogenous features and future exogenous features we use promotions and day of the week. During the model's optimization we consider a balanced dataset of items and stores, for 217,944 bottom level series (4,036 items * 54 stores), along with aggregate levels for a total of 371,312 time series. The dataset is at the daily level and starts from 2013-01-01 and ends by 2017-08-15 that comprehend 1688 days, we keep 34 days (1654 to 1988 days) as hold-out test and 34 days (1620 to 1654 days) as validation.

`Wiki`: The Wikipedia dataset (Anava et al., 2018) contains daily views of 145,000 online articles from July 2015 to December 2016. The dataset is processed into $N_b = 150$ bottom series and $N_b = 49$ aggregate series based on country, access, agent and article categories, following Ben Taieb & Koo (2019); Rangapuram et al. (2021) processing. The last week of December 2016 is used as test, and the remaining data as training and validation. We use day of the week dummies to capture seasonalities, country, access and agent dummy static features, and a seasonal naive 7 anchor.

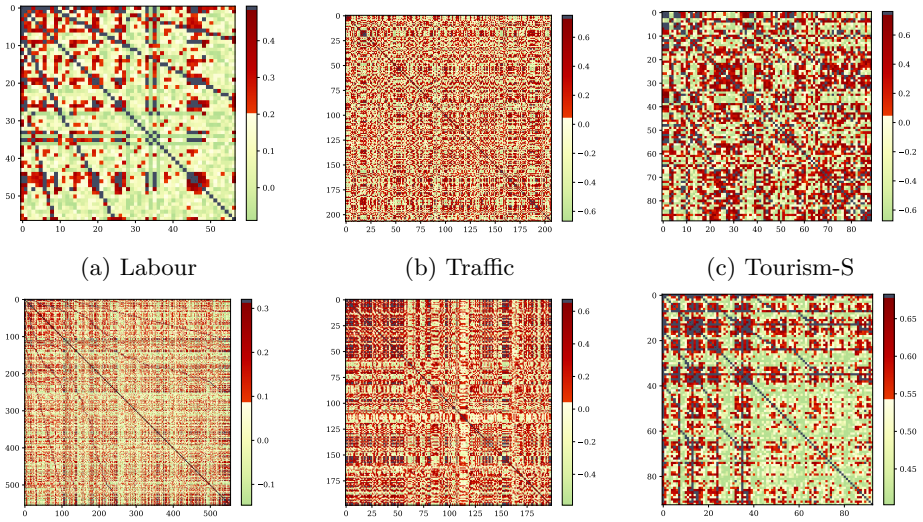
E Forecast Distributions Visualization



(a) Labour (b) Traffic (c) Tourism-S

(d) Tourism-L (e) Wiki (f) Favorita

Figure 7: Probability-Probability plot comparing the cumulative probabilities of the empirical distribution and the `DeepCoFactor`' forecast distribution. Points near the 45-degree line indicate similarity between the distributions. The plot uses a single run from the models reported in the main results Table 3.



(a) Labour (b) Traffic (c) Tourism-S

(d) Tourism-L (e) Wiki (f) Favorita

Figure 8: Estimated correlation matrices produced by `DeepCoFactor`. The color scale indicates the strength of correlation between pairs of series in the hierarchy. Green represents negative correlations, red indicates positive correlations, and black highlights the most highly correlated series. The estimated covariances are sparse, with many correlations close to zero. Aggregated levels, seen in the top-left corner, show stronger positive correlations. The plot uses a single run from the models in Table 3.

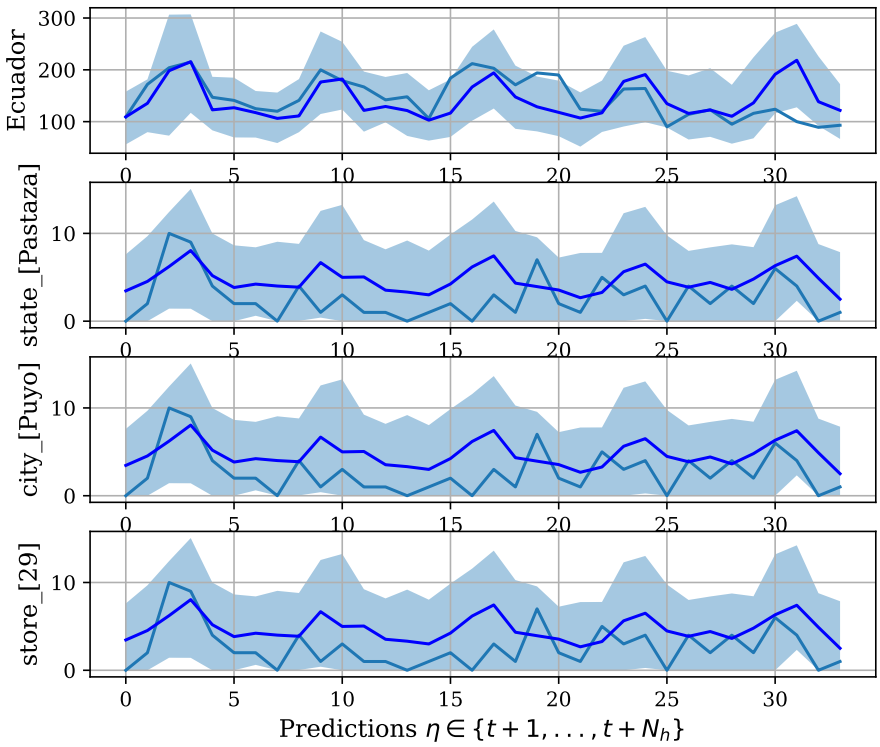


Figure 9: `DeepCoFactor` forecast distributions on the `Favorita` dataset. We show the forecasted demand for a grocery item on a store of the Puyo City, in the State of Pastaza and the whole country demand in the top row. Forecast distributions show the 90% forecast intervals in light blue, and the forecasted median in dark blue. The clipped Normal distribution achieves non-negative predictions and a point mass at zero.

Table 5: Ablation study on the `Traffic` dataset, empirical evaluation of probabilistic coherent forecasts. Mean *scaled continuous ranked probability score* (sCRPS) averaged over 5 runs, at each aggregation level, the best result is highlighted (lower measurements are preferred). We report 95% confidence intervals. Comparison of different neural forecasting architectures trained with the Factor Model and CRPS.

Level	DeepCoFactor	MQCNN	TFT	NHITS	NBEATS	LSTM	FCGAGA
Overall	0.0242±0.0035	0.0613±0.0257	0.0698±0.0106	0.0644±0.0095	0.0688±0.0156	0.0646±0.0066	0.0592±0.0235
Total	0.0035±0.0015	0.0432±0.0296	0.0311±0.0185	0.0339±0.0131	0.0423±0.0175	0.0377±0.0130	0.0241±0.0103
Halves	0.0048±0.0030	0.0437±0.0289	0.0402±0.0104	0.0376±0.0122	0.0443±0.0159	0.0402±0.0089	0.0259±0.0099
Quarters	0.0041±0.0022	0.0440±0.0281	0.0441±0.0111	0.0420±0.0084	0.0478±0.0132	0.0402±0.0084	0.0285±0.0083
Lanes	0.0905±0.0084	0.1145±0.0174	0.1686±0.0159	0.1442±0.0213	0.1409±0.0182	0.1404±0.0071	0.1583±0.0923

Table 6: Ablation study on the `Traffic` dataset. empirical evaluation of probabilistic coherent forecasts. Mean *scaled continuous ranked probability score* (sCRPS) averaged over 5 runs, at each aggregation level, the best result is highlighted (lower measurements are preferred). We report 95% confidence intervals. Comparison of `MQCNN`-based architecture trained with different learning objectives.

* The Normal and StudentT are non coherent forecast distributions, in contrast to the Factor Model and the Poisson Mixture.

Level	DeepCoFactor			Other Distributions		
	CRPS	Energy Score	NLL	PoissonMixture	StudentT*	Normal*
Overall	0.0259±0.0060	0.0269±0.0044	0.0879±0.1136	0.0827±0.1408	0.0600±0.0367	0.0734±0.0253
Total	0.0023±0.0022	0.0031±0.0030	0.0623±0.1335	0.0667±0.1817	0.0280±0.0283	0.0556±0.0327
Halves	0.0028±0.0018	0.0035±0.0027	0.0640±0.1318	0.0765±0.2030	0.0296±0.0300	0.0510±0.0316
Quarters	0.0043±0.0015	0.0074±0.0019	0.0644±0.1313	0.0742±0.1782	0.0301±0.0294	0.0450±0.0334
Lanes	0.0942±0.0195	0.0882±0.0159	0.1608±0.0615	0.1136±0.0056	0.1523±0.0752	0.1422±0.0554

F Ablation Studies Details

To analyze the sources of improvements in our model, we conducted ablation studies on variants of the `DeepCoFactor`/`MQCNN`/`DPMN` (Wen et al., 2017; Olivares et al., 2022c). We utilized a simplified setup on the `Traffic` dataset, focusing on the same forecasting task as the main experiment. We evaluated the sCRPS from Eqn. 19 on the validation set across 5 randomly initialized neural networks. The experiments use the same hyperparameters as reported in Table 4, and vary a single characteristic of interest of the network and measuring its effects on the validation dataset.

In our first ablation study we explore the effects of the impact of including vector autorregresive relationships of the hierarchy through the CrossSeriesMLP module described in Eqn. 14. In the experiment we train a `DeepCoFactor` with and without the module on the `Traffic` dataset. Additionally we compare different well-performing neural forecasting architectures augmented with the CrossSeriesMLP including (1) `LSTM` (Sak et al., 2014), (2) `NBEATS` (Oreshkin et al., 2020; Olivares et al., 2022a), (3) `NHITS` (Challu et al., 2023), (4) `TFT` (Lim et al., 2021), (5) `FCGAGA` (Oreshkin et al., 2021) a network specialized on spatio-temporal forecasting. We use the default implementations available in the NeuralForecast library (Olivares et al., 2022b) [5]. Table 6 shows that convolution-based architectures continue to deliver state-of-the-art results., more importantly using the CrossSeriesMLP improves sCRPS upon the alternative (without) by 66 percent. The technique breaches the gap to the `HierE2E` (Rangapuram et al., 2021), that previously outperformed all alternative methods by over 50 percent. We attribute the improvements to the heavy presence of Granger causal relationships between the traffic lanes, as they carry lag historical information that influence each other.

In our second ablation study we explore learning objective alternatives to Eqn. 17, for this purpose we replace the `DeepCoFactor`'s last layer with different distribution outputs including Normal, Student-T and Poisson Mixture distributions (Olivares et al., 2023). In addition we also compare with our own Factor model approach, as we can see in Table 6 and Figure 3, the CRPS optimization of the Factor Model improves upon the negative log likelihood by 60 percent the mean sCRPS in the validation set. The difference is highly driven by outlier runs, but it is expected as the CRPS objective has much convenient numerical properties, starting by its bounded gradients. Another important note is that in the literature factor model estimation is usually done using evidence lower bound optimization, as the latent factors can quickly land in subpar local minima. In this ablation study we show the CRPS offers a reliable alternative to both.

[5] Optimization of the neural forecasting networks follows details from Appendix C, we increased four times the early stopping patience on `NHITS`/`NBEATS`/`FCGAGA` to account for the gradient variance compared to `MQCNN`'s gradient based on forking sequences.

Table 7: Empirical evaluation of mean hierarchical forecasts. *Relative squared error* (relSE) averaged over 5 runs, at each aggregation level, the best result is highlighted (lower values are preferred). We report 95% confidence intervals, the methods without standard deviation have deterministic solutions.

* The ARIMA-ERM results for Tourism-L differ from Rangapuram et al. (2021), as we improved the numerical stability of their implementation.

Data	Level	DeepCoFactor (crps)	DeepCoFactor (energy)	DPMN-GroupBU	ARIMA-ERM *	ARIMA-MinT-ols	ARIMA-BU	ARIMA (not hier.)	SNaive (not hier.)
Labour	Overall	0.3882±0.0507	0.3816±0.0813	0.7896±0.2825	0.5671	**0.3136**	0.4527	0.3289	5.0683
	1 (geo.)	0.1483±0.0614	0.1421±0.0902	0.3307±0.3012	0.0395	**0.0320**	0.1455	0.0369	5.9572
	2 (geo.)	0.4965±0.0898	0.4889±0.1322	1.3206±0.4084	0.4274	**0.3590**	0.5927	0.3609	5.8649
	3 (geo.)	0.6645±0.0528	**0.6580±0.0765**	1.2675±0.2874	0.9561	0.6807	0.8742	0.6723	4.0696
	4 (geo.)	0.7301±0.0236	**0.7234±0.0285**	1.1693±0.3414	1.8568	0.7605	0.8364	0.8364	2.6208
Traffic	Overall	**0.0008±0.0004**	0.0021±0.0031	0.1750±0.0099	0.0199	0.0425	0.0217	0.0433	0.0709
	1 (geo.)	**0.0001±0.0002**	0.0013±0.0027	0.1619±0.0099	0.0133	0.0344	0.0168	0.0302	0.0547
	2 (geo.)	**0.0001±0.0003**	0.0014±0.0031	0.1835±0.0101	0.0135	0.0380	0.0180	0.0392	0.0676
	3 (geo.)	**0.0005±0.0007**	0.0017±0.0035	0.1819±0.0100	0.0373	0.0647	0.0295	0.0850	0.0989
	4 (geo.)	**0.1354±0.0325**	0.1454±0.0604	0.9964±0.0430	0.6355	0.5876	0.5669	0.5669	1.3118
Tourism-S	Overall	0.1078±0.0129	0.1104±0.0131	0.1141±0.0241	0.3781	0.1772	**0.0994**	0.1886	0.2198
	1 (geo.)	0.1044±0.0189	0.0874±0.0202	0.1362±0.0359	0.4174	0.1722	**0.0696**	0.1649	0.2596
	2 (geo.)	0.0969±0.0060	0.1004±0.0175	0.0797±0.0103	0.367	0.1605	**0.0819**	0.2066	0.1741
	3 (geo.)	**0.1381±0.0098**	0.1642±0.0097	0.1241±0.0299	0.3033	0.2015	0.1684	0.1819	0.2163
	4 (geo.)	**0.1576±0.0079**	0.1872±0.0206	0.1597±0.0371	0.3369	0.2444	0.2218	0.2218	0.2557
Tourism-L	Overall	**0.0951±0.0145**	0.0952±0.0188	0.1113±0.0158	0.1178	0.1251	0.2979	0.1414	0.1306
	1 (geo.)	0.0447±0.0171	0.0405±0.0327	0.0597±0.0212	0.0596	0.0472	0.4002	**0.0343**	0.0582
	2 (geo.)	**0.1014±0.0180**	0.1016±0.0304	0.1121±0.0152	0.1293	0.1476	0.3340	0.2530	0.1628
	3 (geo.)	0.2309±0.0124	0.2299±0.0253	**0.2250±0.0196**	0.2529	0.3556	0.4238	0.4429	0.3695
	4 (geo.)	0.3075±0.0134	0.3263±0.0296	**0.2980±0.0197**	0.3236	0.4288	0.4012	0.4835	0.4766
	5 (prp.)	0.0596±0.0195	**0.0560±0.0078**	0.0798±0.0195	0.0895	0.0856	0.1703	0.0973	0.0615
	6 (prp.)	**0.1199±0.0115**	0.1232±0.0085	0.1403±0.0150	0.1466	0.1537	0.1986	0.1663	0.1577
	7 (prp.)	**0.2484±0.0119**	0.2644±0.0195	0.2654±0.0212	0.2705	0.3017	0.3151	0.2914	0.3699
	8 (prp.)	0.3432±0.0157	0.3830±0.0260	**0.3302±0.0235**	0.3543	0.3970	0.3769	0.3769	0.4969
Wiki	Overall	**0.5301±0.0502**	0.7639±0.1517	0.9270±0.1237	0.9768	1.0253	0.9621	1.0419	0.9288
	1 (geo.)	**0.1980±0.0626**	0.4673±0.2430	0.6294±0.1769	0.8347	1.0528	0.8271	1.1357	0.6555
	2 (geo.)	**0.4963±0.0548**	0.7627±0.1917	0.9266±0.1056	1.0342	0.9913	0.9615	0.8300	1.0672
	3 (geo.)	**0.8272±0.0777**	0.9492±0.3670	1.2059±0.0997	1.0844	1.0097	1.1105	1.0398	1.1441
	4 (geo.)	**0.8152±0.0732**	0.9339±0.3571	1.1789±0.0961	1.0744	1.0030	1.0858	1.0259	1.1095
	5 (prp.)	**0.8820±0.0698**	1.2234±0.5854	1.1958±0.0732	1.0881	1.0294	1.0504	1.0504	1.1080
Favorita	Overall	**0.5885±0.0291**	0.7517±0.0255	0.7563±0.0713	0.8163	0.9465	0.8276	0.9665	1.1420
	1 (geo.)	**0.6109±0.0400**	0.7864±0.0291	0.7944±0.0568	0.8362	0.8999	0.8415	0.9217	1.1269
	2 (geo.)	**0.5618±0.0265**	0.7183±0.0216	0.7355±0.1057	0.7830	1.0057	0.8050	1.0451	1.1078
	3 (geo.)	**0.5619±0.0256**	0.7256±0.0259	0.7303±0.1035	0.7986	1.0418	0.8192	1.0881	1.1315
	4 (geo.)	**0.5854±0.0130**	0.7083±0.0274	0.6770±0.0351	0.8199	0.8808	0.8228	0.8228	1.2815

G Mean Forecast Accuracy Evaluation

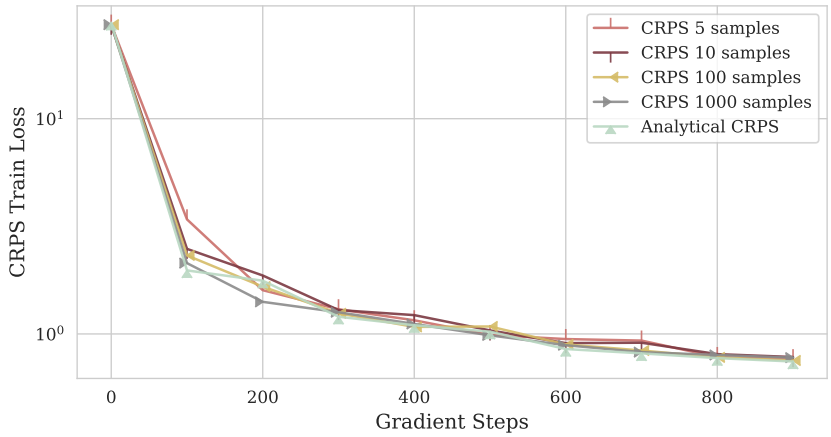
To complement the probabilistic and L1-based results in Section 4.1. We also evaluate mean forecasts denoted by $\bar{\mathbf{y}}_{[i][h],t} := (\bar{\mathbf{y}}_{[i],1,t}, \cdots, \bar{\mathbf{y}}_{[i],N_h,t})$ through the *relative squared error* relSE (Hyndman & Koehler, 2006), that considers the ratio between squared error across forecasts in all levels over squared error of the `Naive` forecast (i.e., a point forecast using the last observation $\mathbf{y}_{[i],t}$) as described by

$$\mathrm{relSE}\left(\mathbf{y}_{[i][h],t},\ \bar{\mathbf{y}}_{[i][t+1:t+N_h]} \mid \boldsymbol{l}^{(g)}\right) = \frac{\sum_{i=1}^{N_a+N_b} \|\mathbf{y}_{i,[t+1:t+N_h]} - \bar{\mathbf{y}}_{i,[h],t}\|_2^2 \cdot l_i^{(g)}}{\sum_{i=1}^{N_a+N_b} \|\mathbf{y}_{i,[t+1:t+N_h]} - y_{i,t} \cdot \mathbf{1}_{[h]}\|_2^2 \cdot l_i^{(g)}}. \tag{25}$$

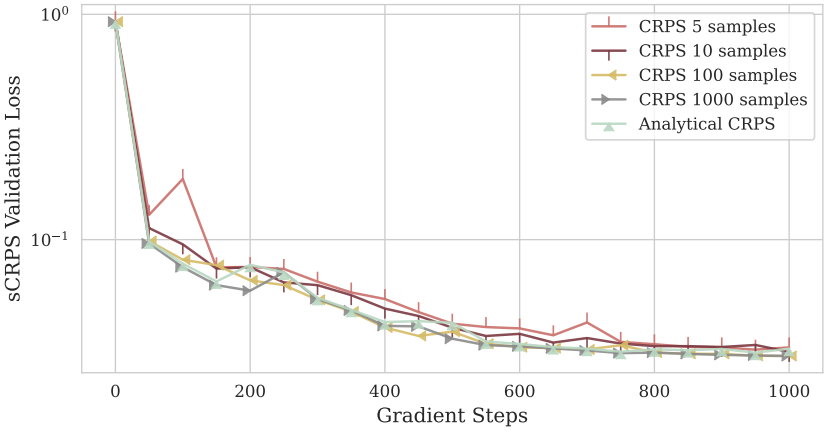
As discussed in Section 3.1, our factor model naturally defines a probabilistic coherent system, where hierarchical coherence emerges as a consequence. In this section, we compare our method with the following coherent mean approaches: (1) `DPMN-GroupBU`, (2) `ARIMA-ERM` (Ben Taieb & Koo, 2019), (3) `ARIMA-MinT` (Wickramasuriya et al., 2019), (4) `ARIMA-BU`, (5) `ARIMA`, and (6) Seasonal Naive. The statistical methods are implemented using the StatsForecast and HierarchicalForecast libraries (Olivares et al., 2022c; Garza et al., 2022). In particular we use available code on the hierarchical baselines repository, and hierarchical datasets repository.

The `DeepCoFactor` model improves relSE accuracy in five out of six datasets by an average of 32%, while degrading accuracy on the smallest dataset, Labour, by 32%. These changes can be attributed to squared error metrics' sensitivity to outliers, though the results are generally consistent with sCRPS outcomes.

H Efficient Estimation of Scoring Rules



(a) Training Trajectory



(b) Validation Trajectory

Figure 10: Training and validation *Continuous Ranked Probability Score* curves on the SF Bay Area `Traffic` dataset. We show curves for `DeepCoFactor` varying as a function of the gradient steps. Each curve uses a different number of Monte Carlo samples to estimate CRPS.

For the network's optimization we implement the following CRPS Monte Carlo estimator for Eqn. 5

$$\hat{\mathrm{CRPS}}\left(y, Y\right) = \frac{1}{N} \sum_{i=1}^{N} |y_i - y| - \frac{1}{2N(N-1)} \sum_{i,j=1}^{N} |y_i - y_j| \tag{26}$$

Where $y_i, y_j \quad i, j = 1, ..., N$ are independent samples of the random variable Y, and y a possible observation.

To assess the effect of sample size on the CRPS Monte Carlo estimator, we ran diagnostics throughout the model training process. Figure 10 shows the training and validation loss trajectories for varying Monte Carlo sample sizes, using model configurations from Table 4. For reference, we include the analytical CRPS of truncated normals (Thorarinsdottir & Gneiting, 2009), despite minor distributional differences compared to Eqn. 10. For a comprehensive review of alternative CRPS estimators we refer to Zamo & Naveau (2018).

We found that reducing the number of Monte Carlo samples had minimal impact on both training and validation scores. Given GPU memory constraints, decreasing the sample size effectively reduces computational overhead without significantly affecting model performance. We observed similar phenomenons with the energy score learning objective, and its relationship to Monte Carlo samples.